Mobility Data Management

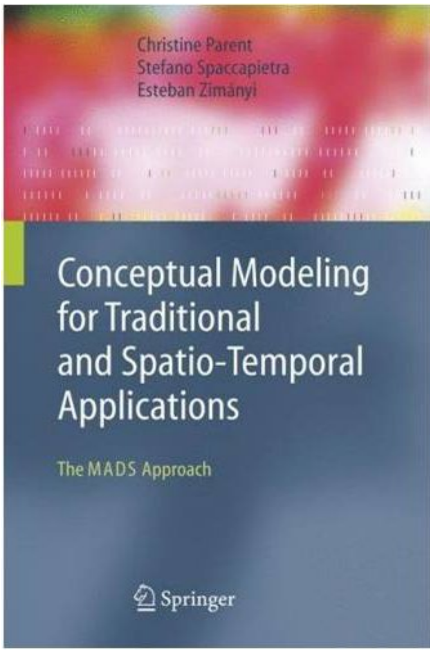
(Spatio-Temporal Data Warehouses)

Esteban ZIMÁNYI
Department of Computer & Decision Engineering (CoDe)
Université Libre de Bruxelles
ezimanyi@ulb.ac.be

CIKM Tutorial on
Mobility Data: Modeling, Management, and Understanding
Toronto, Canada
October 26, 2010

Contents

- Motivation
- Conventional Data Warehouses
- Spatial Data Warehouses
- Temporal Data Warehouses
- Spatio-Temporal Data Warehouses
- Support of Continuous Fields
- Conclusions & Future Works

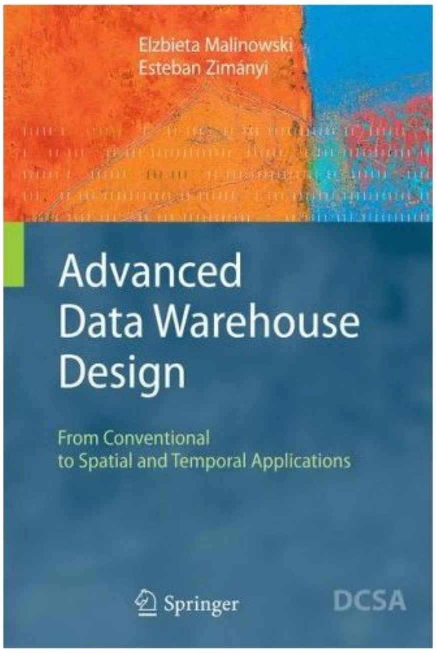


Christine Parent
Stefano Spaccapietra
Esteban Zimányi

Conceptual Modeling for Traditional and Spatio-Temporal Applications

The MADS Approach

Springer



Elzbieta Malinowski
Esteban Zimányi

Advanced Data Warehouse Design

From Conventional to Spatial and Temporal Applications

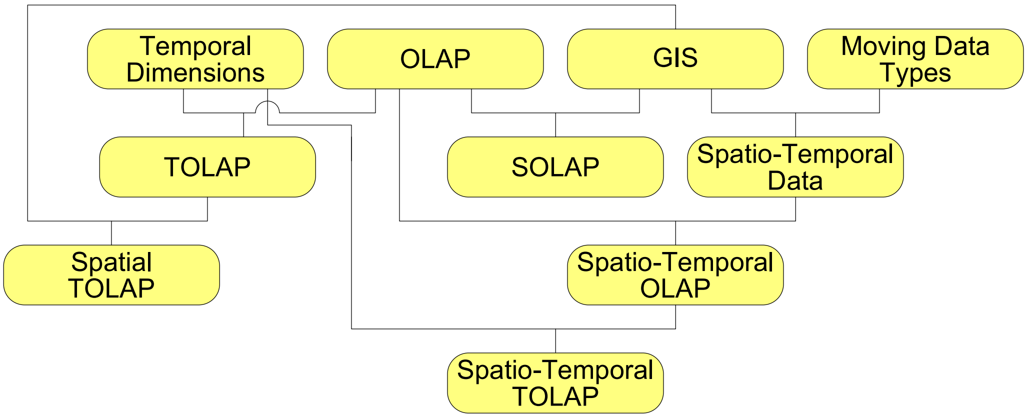
Springer

DCSA

Motivation

- **Data Warehouses** (**DW**) and **OLAP** systems are currently used for storing **huge amounts of data** and performing **analysis** and **business intelligence**
  - This is what we need for **mobility data**!
- Several proposals aim at extending DW and OLAP with **spatial/temporal features**
- **No commonly agreed definition** of what is a spatio-temporal DW and what functionality it should support
- Proposed solutions **vary considerably** in the kind of **data that can be represented** and the kind of **queries that can be expressed**
- We defined
  - **Conceptual framework for spatio-temporal DWs** using an extensible type system
  - **Taxonomy of several classes of queries** of increasing expressive power using an extension of the tuple relational calculus with aggregated functions
- This provides the **underlying basis** for **implementing** spatio-temporal DWs

A Taxonomy for Spatio-Temporal Data Warehouses



Temporal Dimensions

OLAP

GIS

Moving Data Types

TOLAP

SOLAP

Spatio-Temporal Data

Spatial TOLAP

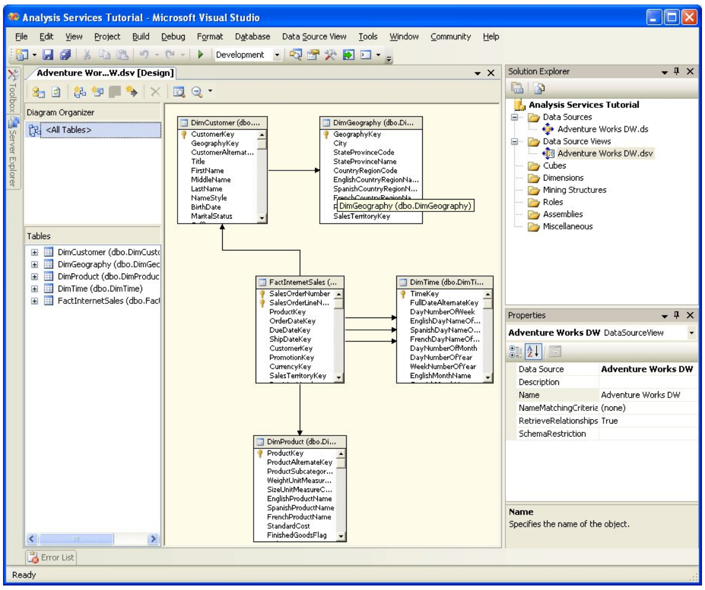
Spatio-Temporal OLAP

Spatio-Temporal TOLAP

Conventional Data Warehouses

- **Operational databases** (**OLTP**) not suitable for data analysis
  - Contain detailed data
  - Do not include historical data
  - Perform poorly for complex queries due to normalization
- **Data warehouses** address requirements of decision-making users
- A data warehouse is a collection of **subject-oriented**, **integrated**, **nonvolatile**, and **time-varying** data to support data management decisions
- **Online analytical processing** (**OLAP**): Allows decision-making users to perform interactive analysis of data

Current DW Tools: SQL Server Analysis Services

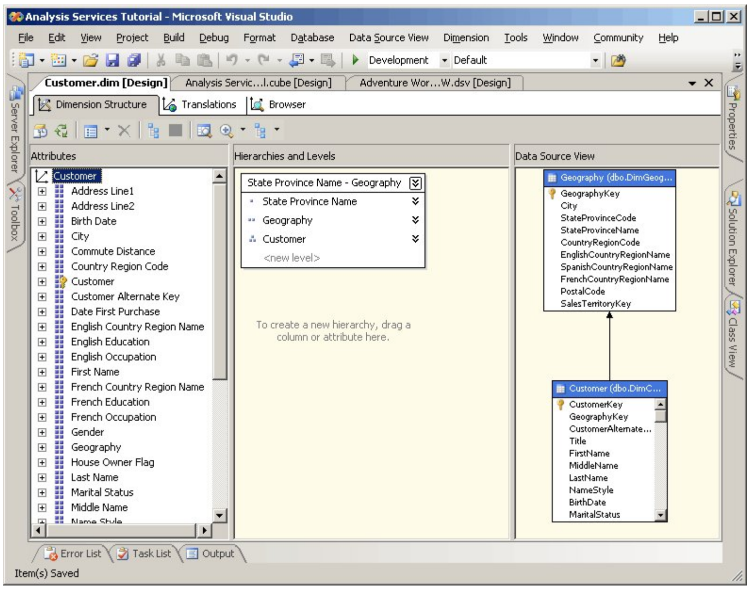


Adventure Works DW DataSourceView

Data Source	**Adventure Works DW**
Description	
Name	Adventure Works DW
NameMatchingCriteria	(none)
RetrieveRelationships	True
SchemaRestriction	

Name
Specifies the name of the object.

Current DW Tools: Hierarchies (1)


Analysis Services Tutorial - Microsoft Visual Studio
File Edit View Project Build Debug Format Database Data Source View Dimension Tools Window Community Help
Development
Default
Customer.dim [Design]
Analysis Servic...l.cube [Design]
Adventure Wor...W.dsv [Design]
Dimension Structure
Translations
Browser
Server Explorer
Toolbox
Properties
Solution Explorer
Class View
Error List
Task List
Output

Attributes

- Customer
- Address Line1
- Address Line2
- Birth Date
- City
- Commute Distance
- Country Region Code
- Customer
- Customer Alternate Key
- Date First Purchase
- English Country Region Name
- English Education
- English Occupation
- First Name
- French Country Region Name
- French Education
- French Occupation
- Gender
- Geography
- House Owner Flag
- Last Name
- Marital Status
- Middle Name
- Name Style

Hierarchies and Levels

State Province Name - Geography

- State Province Name
- Geography
- Customer
- <new level>

To create a new hierarchy, drag a column or attribute here.

Data Source View

Geography (dbo.DimGeog...

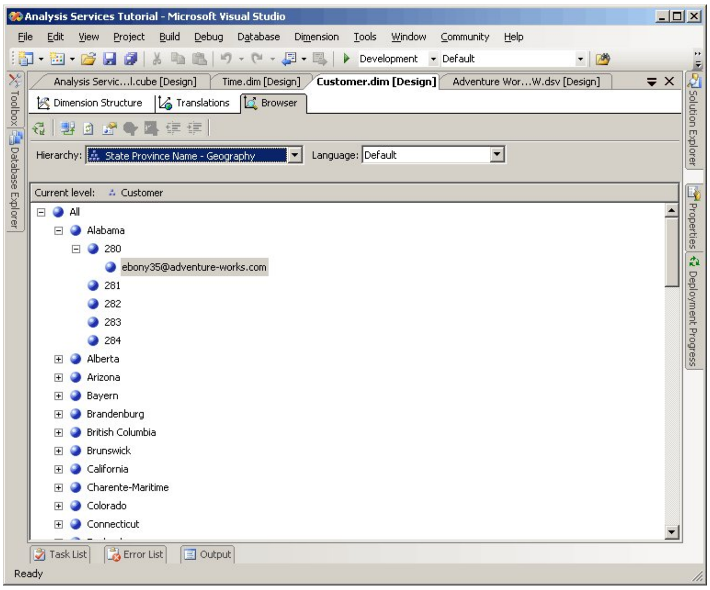
- GeographyKey
- City
- StateProvinceCode
- StateProvinceName
- CountryRegionCode
- EnglishCountryRegionName
- SpanishCountryRegionName
- FrenchCountryRegionName
- PostalCode
- SalesTerritoryKey

Customer (dbo.DimC...

- CustomerKey
- GeographyKey
- CustomerAlternate...
- Title
- FirstName
- MiddleName
- LastName
- NameStyle
- BirthDate
- MaritalStatus

Item(s) Saved

Current DW Tools: Hierarchies (2)


Analysis Services Tutorial - Microsoft Visual Studio
File Edit View Project Build Debug Database Dimension Tools Window Community Help
Development
Default
Analysis Servic...l.cube [Design]
Time.dim [Design]
Customer.dim [Design]
Adventure Wor...W.dsv [Design]
Dimension Structure
Translations
Browser
Toolbox
Database Explorer
Solution Explorer
Properties
Deployment Progress
Task List
Error List
Output

Hierarchy: State Province Name - Geography

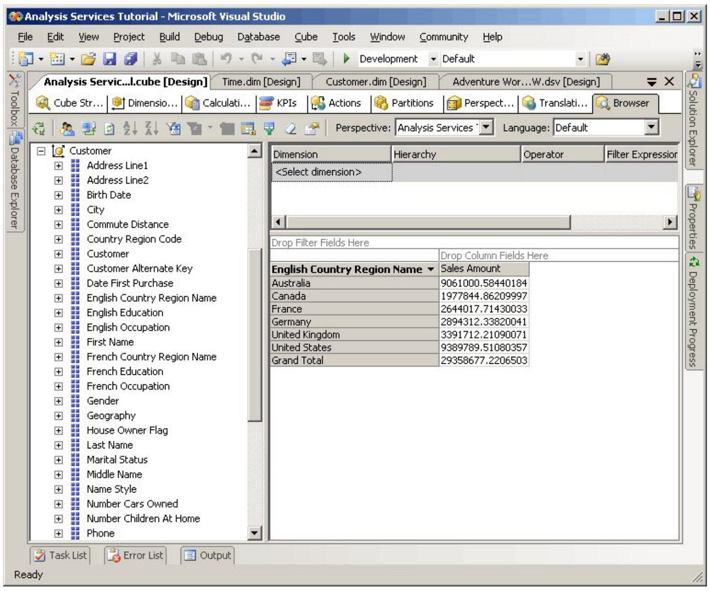
Language: Default

Current level: Customer

- All
  - Alabama
    - 280
      - ebony35@adventure-works.com
    - 281
    - 282
    - 283
    - 284
  - Alberta
  - Arizona
  - Bayern
  - Brandenburg
  - British Columbia
  - Brunswick
  - California
  - Charente-Maritime
  - Colorado
  - Connecticut

Ready

Current DW Tools: Pivot Tables (1)



Customer
- Address Line1
- Address Line2
- Birth Date
- City
- Commute Distance
- Country Region Code
- Customer
- Customer Alternate Key
- Date First Purchase
- English Country Region Name
- English Education
- English Occupation
- First Name
- French Country Region Name
- French Education
- French Occupation
- Gender
- Geography
- House Owner Flag
- Last Name
- Marital Status
- Middle Name
- Name Style
- Number Cars Owned
- Number Children At Home
- Phone

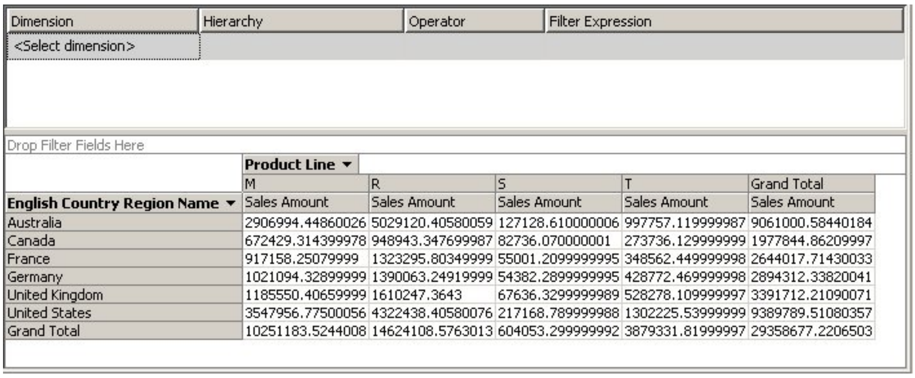
Dimension	Hierarchy	Operator	Filter Expression
<Select dimension>			

Drop Filter Fields Here

Drop Column Fields Here

English Country Region Name	Sales Amount
Australia	9061000.58440184
Canada	1977844.86209997
France	2644017.71430033
Germany	2894312.33820041
United Kingdom	3391712.21090071
United States	9389789.51080357
Grand Total	29358677.2206503

Current DW Tools: Pivot Tables (2)

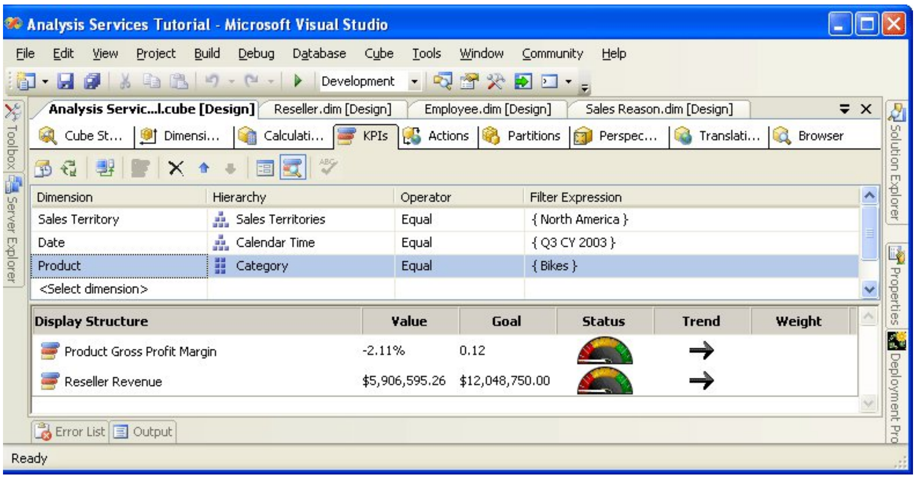


Dimension	Hierarchy	Operator	Filter Expression
<Select dimension>			

Drop Filter Fields Here

	Product Line				
	M	R	S	T	Grand Total
English Country Region Name	Sales Amount	Sales Amount	Sales Amount	Sales Amount	Sales Amount
Australia	2906994.44860026	5029120.40580059	127128.610000006	997757.119999987	9061000.58440184
Canada	672429.314399978	948943.347699987	82736.070000001	273736.129999999	1977844.86209997
France	917158.25079999	1323295.80349999	55001.2099999995	348562.449999998	2644017.71430033
Germany	1021094.32899999	1390063.24919999	54382.2899999995	428772.469999998	2894312.33820041
United Kingdom	1185550.40659999	1610247.3643	67636.3299999989	528278.109999997	3391712.21090071
United States	3547956.77500056	4322438.40580076	217168.789999988	1302225.53999999	9389789.51080357
Grand Total	10251183.5244008	14624108.5763013	604053.299999992	3879331.81999997	29358677.2206503

Current DW Tools: Key Performance Indications

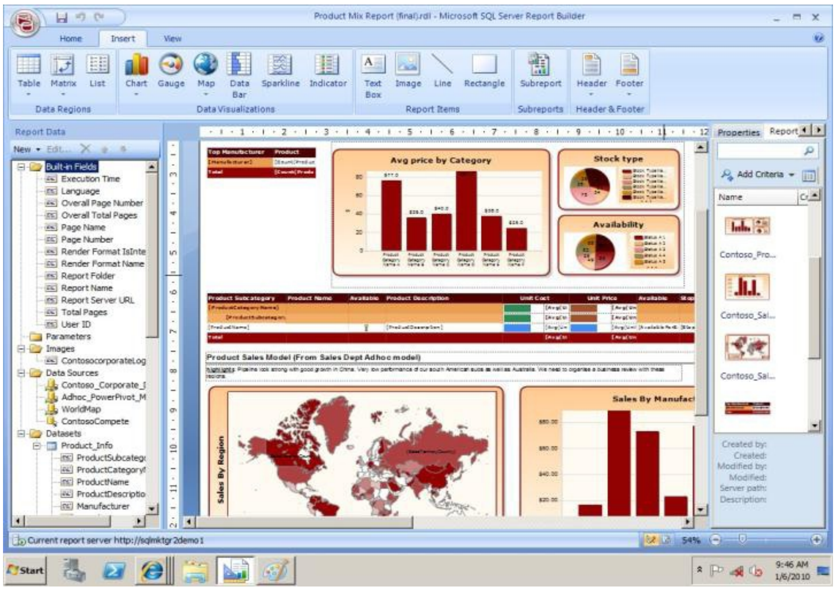


Analysis Services Tutorial - Microsoft Visual Studio

Dimension	Hierarchy	Operator	Filter Expression
Sales Territory	Sales Territories	Equal	{ North America }
Date	Calendar Time	Equal	{ Q3 CY 2003 }
Product	Category	Equal	{ Bikes }
<Select dimension>			

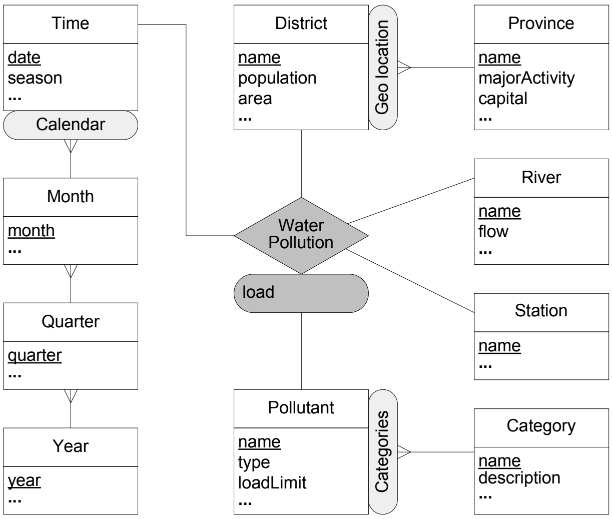
Display Structure	Value	Goal	Status	Trend	Weight
Product Gross Profit Margin	-2.11%	0.12		→	
Reseller Revenue	$5,906,595.26	$12,048,750.00		→	

Current DW Tools: Reports



Product Mix Report (final).rdl - Microsoft SQL Server Report Builder

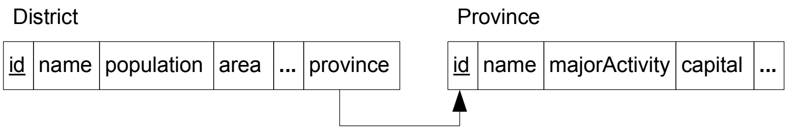
Conventional Data Warehouses: Example



Data Types [Güting et al.]

- **Base types**: int, real, bool, string, id (identifier type)
- **Time types**: instant, periods
- **Range constructor**: range(α) allows to represent finite sets of disjoint, nonadjacent intervals over type α (e.g., int, real, . . .)
- Types have usual interpretation, except that domain is extended by $\perp$ (undefined)
- Operations on these types are formally defined (see later)

Query Language



District

id	name	population	area	...	province

Province

id	name	majorActivity	capital	...

- Functional SQL-like query language with aggregate functions, variables, nested subqueries
- **Simple query**: Name and population of districts of the Antwerp province

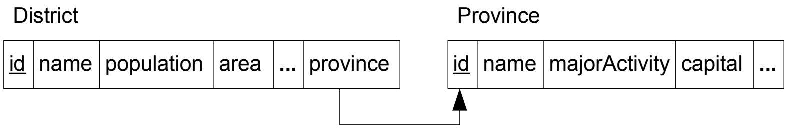
```
SELECT d.name, d.population
FROM District d, Province p
WHERE d.province=p.id AND p.name='Antwerp'
```

- **Aggregation query**: Total population of districts of the Antwerp province

```
sum( SELECT d.population FROM District d, Province p
WHERE d.province=p.id AND p.name='Antwerp' )
```

- As in SQL, the inner query defines a bag (not a set)
- If two districts have the same population they appear twice in the result to be aggregated

Query Language (2)



District

id	name	population	area	...	province

Province

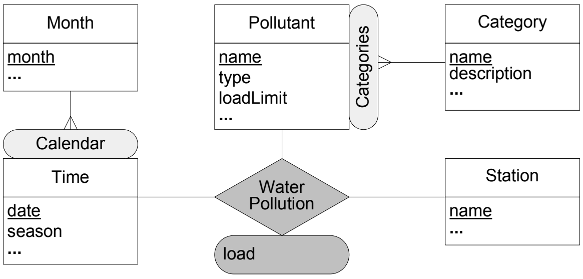
id	name	majorActivity	capital	...

- **Aggregation query with group filtering**: Total population by province provided that it is greater than 100,000

```
SELECT p.name, totalPop
FROM Province p
WHERE totalPop = sum(
    SELECT d.id, d.population FROM District d
    WHERE d.province=p.id )
AND totalPop > 100000
```

- Corresponds to an SQL query with the GROUP BY and HAVING clauses

OLAP Queries



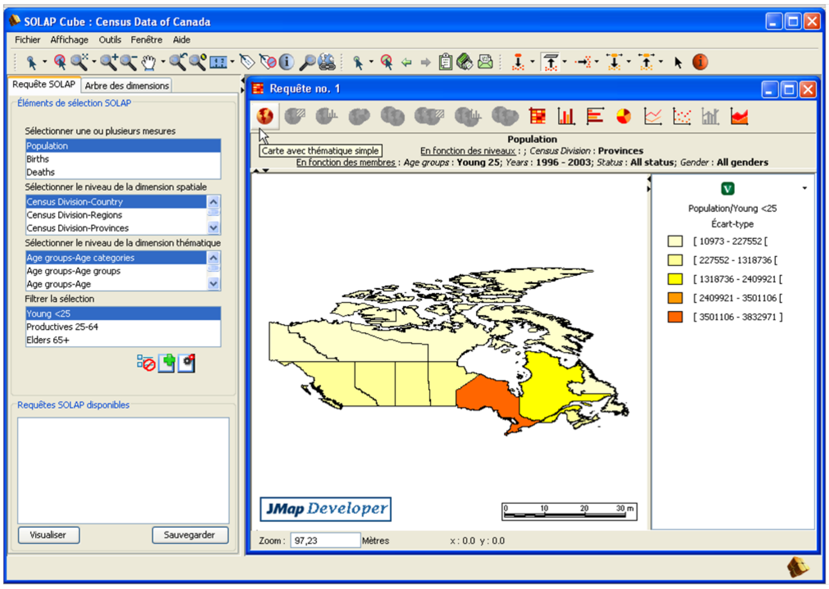
- For water stations and pollutants of organic category give the maximum load by month

```
SELECT s.name, p.name, m.month, maxLoad
FROM Station s, Pollutant p, Month m, Category c
WHERE p.category=c.id AND c.name=‘Organic’
AND maxLoad= max(
      SELECT w.load FROM WaterPollution w, Time t
      WHERE w.station=s.id AND w.pollutant=p.id AND w.time=t.id
      AND t.month=m.id )
```

Spatial Data Warehouses: Current State

- Relatively **new field** of research
- No **formal definition** of SDWs: many different meanings
- Much research related to **implementation issues**
  - Integration of DWs and GISs
  - Improvement of performance by using indexes
  - Materializations for spatial data cubes
  - . . .
- Some proposals and implementations of **spatial OLAP**
- Little research on **conceptual modeling for SDWs**

Current SDW Tools: JMap



SOLAP Cube : Census Data of Canada

Éléments de sélection SOLAP

Sélectionner une ou plusieurs mesures: **Population**, Births, Deaths

Sélectionner le niveau de la dimension spatiale: **Census Division-Country**, Census Division-Regions, Census Division-Provinces

Sélectionner le niveau de la dimension thématique: **Age groups-Age categories**, Age groups-Age groups, Age groups-Age

Filtrer la sélection: **Young <25**, Productives 25-64, Elders 65+

Requêtes SOLAP disponibles

Requête no. 1

Population

En fonction des niveaux : ; *Census Division* : **Provinces**

En fonction des membres : *Age groups* : **Young 25**; *Years* : **1996 - 2003**; *Status* : **All status**; *Gender* : **All genders**

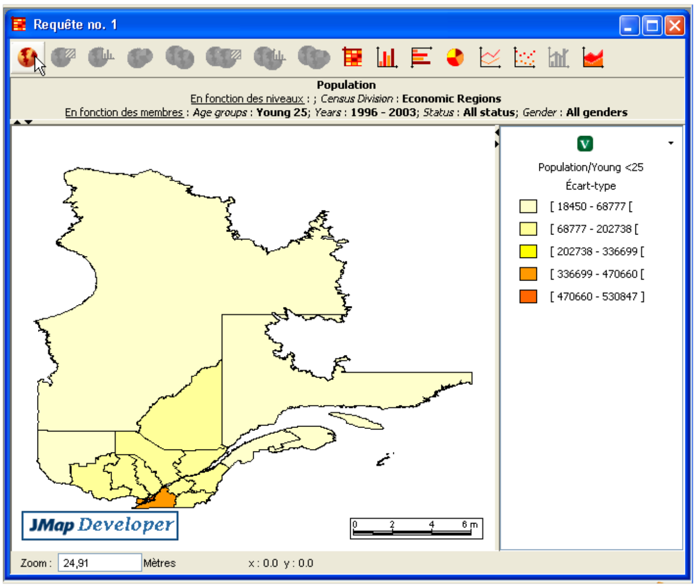
Population/Young <25

Écart-type

- [10973 - 227552 [
- [227552 - 1318736 [
- [1318736 - 2409921 [
- [2409921 - 3501106 [
- [3501106 - 3832971]

Zoom : 97,23 Mètres x : 0.0 y : 0.0

Current SDW Tools: Reports



Requête no. 1

Population

En fonction des niveaux : ; *Census Division* : **Economic Regions**

En fonction des membres : *Age groups* : **Young 25**; *Years* : **1996 - 2003**; *Status* : **All status**; *Gender* : **All genders**

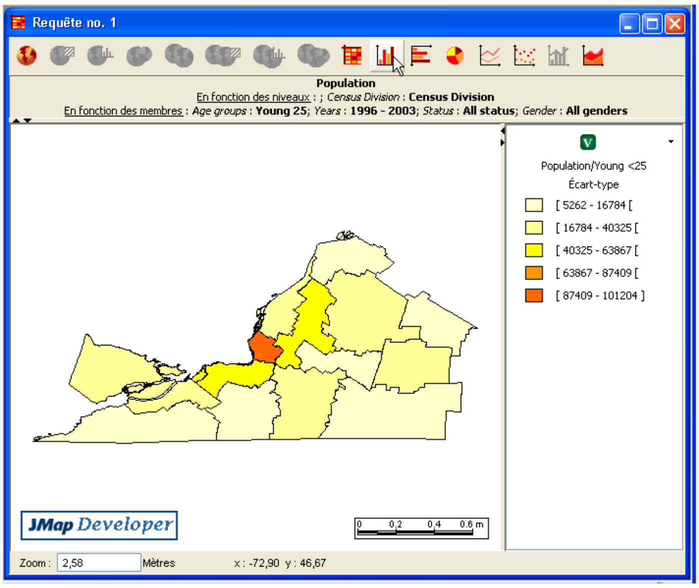
Population/Young <25

Écart-type

- [18450 - 68777 [
- [68777 - 202738 [
- [202738 - 336699 [
- [336699 - 470660 [
- [470660 - 530847]

Zoom : 24,91 Mètres x : 0.0 y : 0.0

Current SDW Tools: Reports



Requête no. 1

Population

En fonction des niveaux : ; *Census Division* : **Census Division**

En fonction des membres : *Age groups* : **Young 25**; *Years* : **1996 - 2003**; *Status* : **All status**; *Gender* : **All genders**

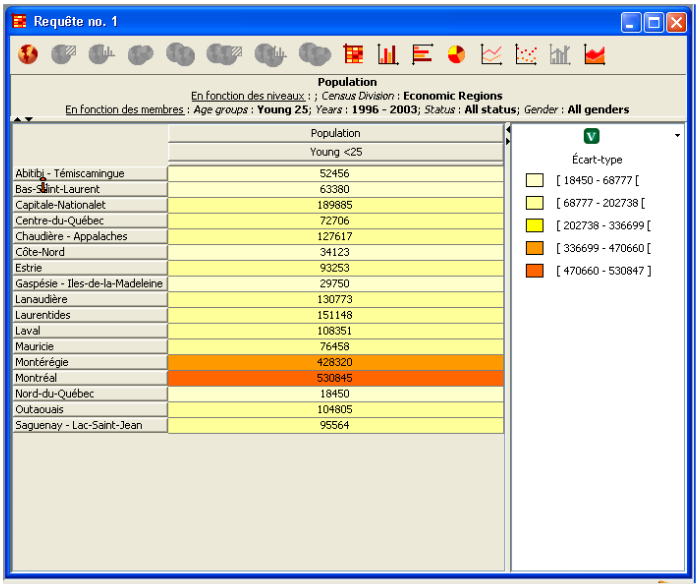
Population/Young <25

Écart-type

- □ [5262 - 16784 [
- □ [16784 - 40325 [
- 🟨 [40325 - 63867 [
- 🟧 [63867 - 87409 [
- 🟧 [87409 - 101204]

Zoom : 2,58 Mètres x : -72,90 y : 46,67

Current SDW Tools: Reports



Requête no. 1

Population

En fonction des niveaux : ; *Census Division* : **Economic Regions**

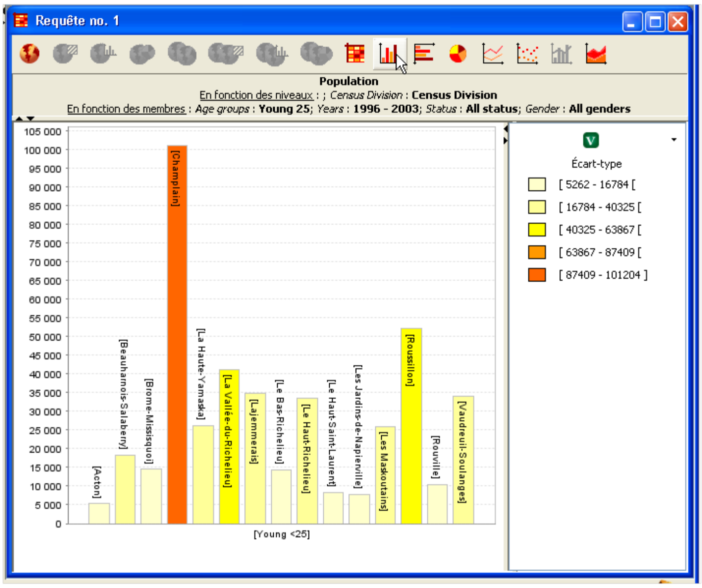
En fonction des membres : *Age groups* : **Young 25**; *Years* : **1996 - 2003**; *Status* : **All status**; *Gender* : **All genders**

	Population
	Young <25
Abitibi - Témiscamingue	52456
Bas-Saint-Laurent	63380
Capitale-Nationalet	189885
Centre-du-Québec	72706
Chaudière - Appalaches	127617
Côte-Nord	34123
Estrie	93253
Gaspésie - Iles-de-la-Madeleine	29750
Lanaudière	130773
Laurentides	151148
Laval	108351
Mauricie	76458
Montérégie	428320
Montréal	530845
Nord-du-Québec	18450
Outaouais	104805
Saguenay - Lac-Saint-Jean	95564

Écart-type

- □ [18450 - 68777 [
- □ [68777 - 202738 [
- 🟨 [202738 - 336699 [
- 🟧 [336699 - 470660 [
- 🟧 [470660 - 530847]

Current SDW Tools: Reports



Requête no. 1

Population

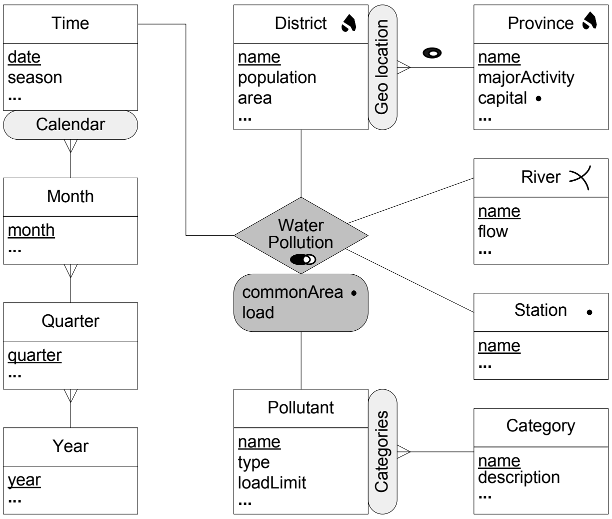
<u>En fonction des niveaux</u> : ; *Census Division* : **Census Division**

<u>En fonction des membres</u> : *Age groups* : **Young 25**; *Years* : **1996 - 2003**; *Status* : **All status**; *Gender* : **All genders**

Écart-type

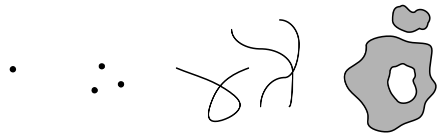
🟨	[5262 - 16784 [
🟨	[16784 - 40325 [
🟨	[40325 - 63867 [
🟧	[63867 - 87409 [
🟧	[87409 - 101204]

Spatial Data Warehouses: Example



Spatial Types [Güting et al.]

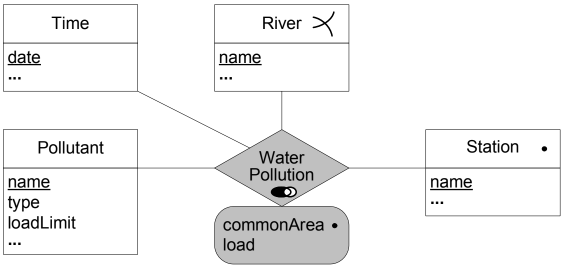
- Three types: point, points, line, region



- Base and spatial types have an associated set of operations

Class	Operations
Predicates	isempty, $=$, $\neq$, intersects, inside, $<$, $\leq$, $\geq$, $>$, before, touches, attached, overlaps, on_border, in_interior
Set Operations	intersection, union, minus, crossings, touch_points, common_border
Aggregation	min, max, avg, center, single
Numeric	no_components, size, duration, length, area, perimeter
Distance and Direction	distance, direction
Base Type Specific	and, or, not

Spatial OLAP (SOLAP) Queries



- For stations located in the Schelde river, give the average content of lead in the last quarter of 2008

```
SELECT s.name, avgLead
FROM Station s, River r, Pollutant p
WHERE r.name=‘Schelde’ AND p.name=‘Lead’
AND intersects(r.geometry,s.geometry) AND avgLead = avg(
    SELECT w.load FROM WaterPollution w, Time t
    WHERE w.station=s.id AND w.pollutant=p.id AND w.time= t.id
    AND t.date >= 1/10/2008 AND t.date <= 31/12/2008 )
```

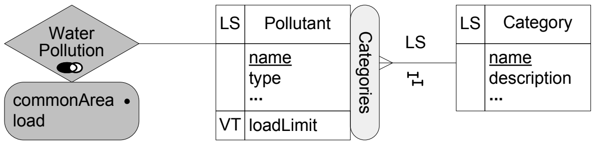
Temporal Information

- Current DWs and OLAP tools allow to model temporal evolution of measures through a Time dimension
  - Do not allow to keep track of changes in other dimensions
  - Existing solutions (slowly changing dimensions) are unsatisfactory and ad hoc
- **Temporal databases** have been studied for several decades for managing time-varying information
- Combining this research with data warehouses leads to **temporal data warehouses**

Temporal Dimensions

- **Valid time** (**VT**): when data is true in modelled reality
  - when specific salary was paid for employee
- **Transaction time** (**TT**): when data is true in database
  - when specific salary was recorded in database
- **Bitemporal** (**BT**): combine valid and transaction time
- **Lifespan** (**LS**): time during which entity exists
  - when employee has been hired and fired
- **DWLT** (**DW loading time**): generated in DWs
  - when data about employee recorded in the DW
- Temporal support for levels, attributes, child-parent relationships, and measures

Temporal Data Warehouses



- Arise when **evolution of dimension instances** is supported
  - Also referred to as **slowly changing dimensions** [Kimball 96]
- Represented using **moving types** moving(α) where α is a **base type**
  - Lifespan of Pollutant is of type moving(bool)
  - loadLimit of Pollutant is of type moving(real)
  - Relationship between Pollutant and Category is of type moving(id)

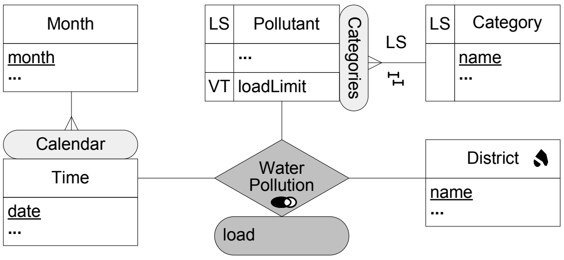
Moving Types [Güting et al.]

- Capture the evolution over time of base types and spatial types
- Obtained by applying a constructor moving(α)
  - A value of type moving(point) is a continuous function f : instant → point
- Operations on moving types

Class	Operations
Projection to Domain/Range	deftime, rangevalues, locations, trajectory, routes, traversed, inst, val
Interaction with Domain/Range	atinstant, atperiods, initial, final, present, at, atmin, atmax, passes
Rate of change	derivative, speed, turn, velocity
Lifting	(all new operations inferred)

- **Lifting**: Operations of moving types generalize those of the nontemporal types
  - A distance function with signature moving(point) × moving(point) → moving(real) calculates the distance between two moving points
- **Semantics**: result is **computed at each time instant** using the non-lifted operation

Temporal OLAP (TOLAP) Queries



- For each district and pollutant category, give the average load of water pollution by month

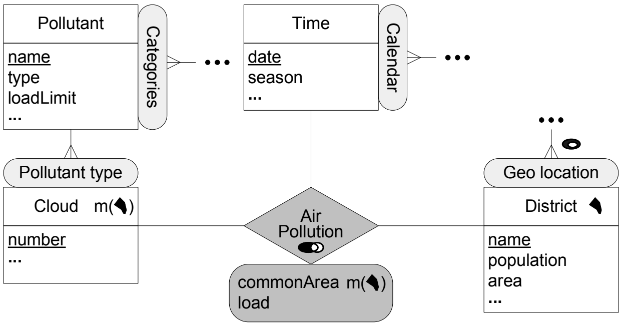
```
SELECT d.name, c.name, m.month, avgLoad
FROM District d, Category c, Month m
WHERE avgLoad = avg(
    SELECT w.load FROM WaterPollution w, Time t, Pollutant p
    WHERE w.district=d.id AND w.time=t.id AND t.month=m.id
    AND w.pollutant=p.id AND val(initial(atperiods(p.category,t)))=c.id )
```

Spatio-Temporal Processes

- Evolution of **spatial attributes** of objects
  - Movements, shape modifications, . . .
- Evolution of **properties of space**
  - Temperature, land occupation, . . .
- Evolution of **spatial relationships** linking objects
  - Plane – Storm: far ⇒ near ⇒ inside . . .
  - Country – Country: adjacent ⇒ disjoint ⇒ adjacent . . .
- **Generation / transition** of (spatial) objects
  - Splitting or merging parcels, countries, . . .
- **Transmission of properties**
  - Disease, innovation , fashion, . . .

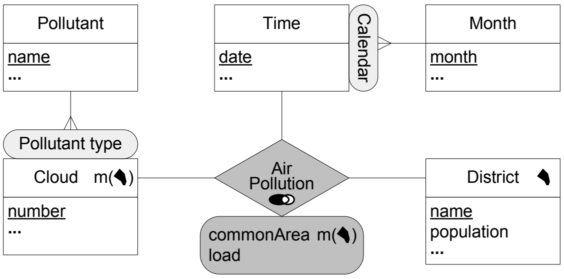
Trajectory data is just a special case of spatio-temporal data

Spatio-Temporal OLAP (ST-OLAP)



- Arise when **spatial objects evolve over time**
- Evolution captured by **moving types** moving(α) where α is a **spatial type**

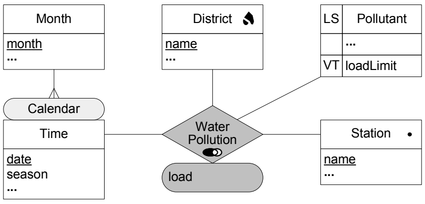
Spatio-Temporal OLAP (ST-OLAP) Queries



- For each district, give by month the total number of persons affected by polluting clouds

```
SELECT d.name,m.month,totalNo
FROM District d, Month m
WHERE totalNo = area( union(
        SELECT traversed(p.commonArea) FROM AirPollution p, Time t
        WHERE p.district=d.id AND p.time=t.id AND t.month=m.id ) ) /
    area(d.geometry) * d.population
```

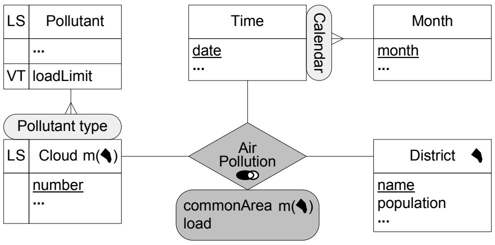
Spatial TOLAP (S-TOLAP)



- Arise when there are **spatial objects/attributes and temporal dimensions**
- For each month and water station in the district of Namur, give the average load if it is larger than the load limit during the reported month

```
SELECT m.month, s.name, avgLoad
FROM Month m, Station s, District d, Pollutant p
WHERE d.name=‘Namur’ AND inside(s.geometry,d.geometry) AND avgLoad = avg(
    SELECT w.load FROM WaterPollution w, Time t
    WHERE w.station=s.id AND w.time=t.id AND t.month=m.id
    AND w.pollutant=p.id )
AND avgLoad > val(initial(atperiods(p.loadLimit,m.month)))
```

Spatio-Temporal TOLAP (ST-TOLAP)



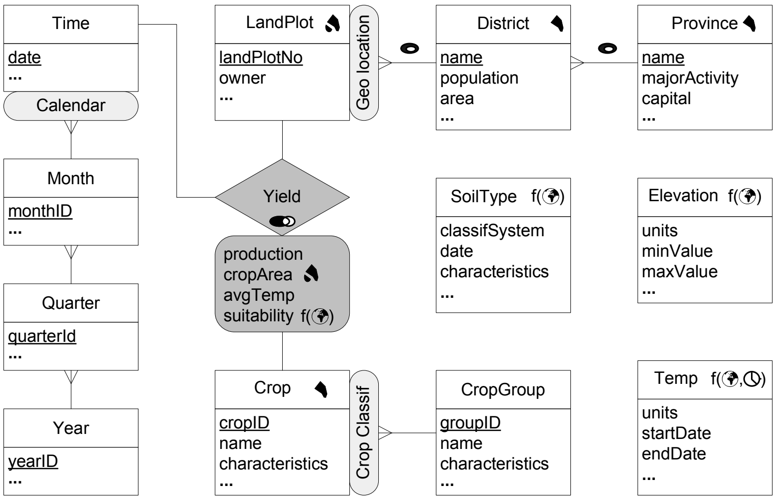
- **Most general case**: there are **moving geometries and temporal dimensions**
- Number of days when the Gent district was under at least one cloud of carbon monoxide (CO) whose load is larger than the load limit at the time when the cloud appeared

```
duration(union(
  SELECT t.date
  FROM Time t, AirPollution p, District d, Cloud c, Pollutant l
  WHERE p.time=t.id AND p.district=d.id AND d.name=‘Ghent’
  AND p.cloud=c.id AND c.pollutant=l.id AND l.name=‘CO’ AND p.load >
      val(atinstant(l.loadLimit,inst(initial(at(c.lifespan,true)))))))
```

Supporting Continuous Fields

- Describe physical phenomena that change continuously in time and/or space
  - temperature, pressure, land elevation, . . .
- Formally, a field is composed of:
  (1) a domain $\mathcal{D}$, which is a continuous set
  (2) a range of values $\mathcal{R}$
  (3) a mapping function f from $\mathcal{D}$ to $\mathcal{R}$
- Multidimensional analysis of continuous data still open
- We defined the field data type, extending the type system of Güting et al.

Spatial Data Warehouses with Continuous Fields



Time: date, …; Calendar

Month: monthID, …

Quarter: quarterId, …

Year: yearID, …

LandPlot: landPlotNo, owner, …

Geo location

District: name, population, area, …

Province: name, majorActivity, capital, …

Yield: production, cropArea, avgTemp, suitability f(🌍)

SoilType f(🌍): classifSystem, date, characteristics, …

Elevation f(🌍): units, minValue, maxValue, …

Crop: cropID, name, characteristics, …

Crop Classif

CropGroup: groupID, name, characteristics, …

Temp f(🌍,🕓): units, startDate, endDate, …

Modeling Continuous Fields

- *Non-temporal* field levels and measures are identified by f(🌍)
- *Temporal* field levels are identified by f(🌍,🕓)
- Field levels have a geometry attribute of type field(α) or moving(field(α))
- Field dimensions are **not connected to a fact relationship** (unlike other DW models)
- **Field measures** are represented by a field data type
  - suitability measure could be precomputed as a function of many factors: e.g., soil type, soil pH level, and temperature
- Traditional numerical measures can be calculated from field data
  - avgTemp keeps the average temperature (a real value) of each instance of the fact relationship, is computed from dimensions Temperature, LandPlot, and Time

Field Types [Vaisman & Zimányi]

- Capture the variation in space of base types
- Obtained applying a constructor field(α)
  - A value of type field(real) (e.g., altitude) is a continuous function f : point → real
- Operations on field types

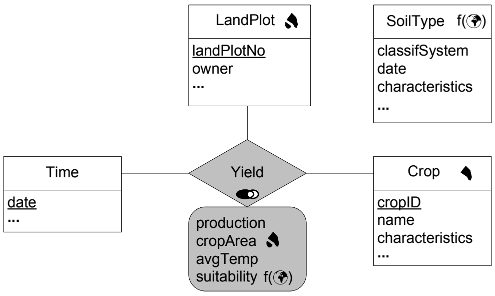
Class	Operations
Projection to Domain/Range	defspace, rangevalues, point, val
Interaction with Domain/Range	atpoint, atpoints, atline, atregion, at, atmin, atmax, defined, takes,concave, convex, flex
Lifting	(all new operations inferred)
Rate of change	partialder_x, partialder_y
Aggregation operators	integral, area, surface, favg, fvariance, fstdev

- **Lifting** applies to fields
  - The + operator with signature $\alpha \times \alpha \rightarrow \alpha$ generalized by allowing any argument to be a field as in field(α) × field(α) → field(α)
- **Semantics**: result is **computed at each point in space** using the non-lifted operation

Operations on Field Types

- partialder_x and partialder_y: partial derivatives of the function defining the field with respect to x and y
  - $\frac{\partial f}{\partial x}(x, y) = \lim_{h \to 0} \frac{f(x+h,y)-f(x,y)}{h}$
- Basic field aggregation
  - integral: $\iint_S f(x, y)\, dxdy$
  - area: $\iint_S dxdy$
  - surface: $\iint_S \sqrt{1 + \frac{df^2}{dx} + \frac{df^2}{dy}}\, dxdy$
- Derived operators for fields
  - favg: integral/area
  - fvariance: $\iint_S \frac{(f(x,y)-\text{favg})^2}{\text{area}}\, dxdy$
  - fstdev: $\sqrt{\text{fvariance}}$

SOLAP-CF Queries



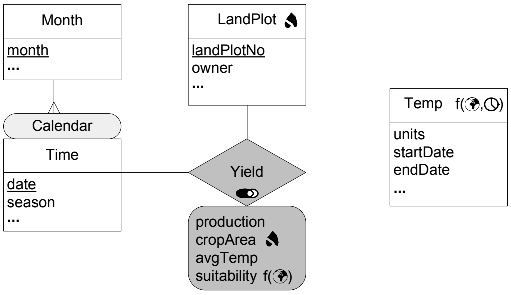
- For land plots having at least 30% of their surface of clay soil, give the average suitability value for wheat on February 1st, 2009

```
SELECT l.number, favg(y.suitability)
FROM LandPlot l, SoilType s, Yield y, Time t, Crop c
WHERE y.landplot=l.id AND y.time=t.id AND t.date=1/2/2009
AND y.crop=c.id AND c.name=‘Wheat’
AND (area(defspace(atregion(at(s.geometry,‘Clay’),l.geometry)))/
     area(l.geometry)) >= 0.3
```

Temporal Field Types

- Model phenomena whose value change along time and space:
  - temperature, precipitation, . . .
- Represented by composing the moving type and the field type constructors
  - Capture the evolution over time of field types
- A temporal field over α is defined by moving(field(α))
  - moving(field(real)) defines a continuous function f : instant → (point → real)
- Operators for moving fields as before

Spatiotemporal OLAP with Continuous Fields



- For each land plot and each month, give a field computing the average temperature of the month at each point in the land plot

```
SELECT l.number, m.month, temp FROM LandPlot l, Month m
WHERE first=min( SELECT t.date FROM Time t WHERE t.month=m.id )
AND last=max( SELECT t.date FROM Time t WHERE t.month=m.id )
AND temp= avg(
    SELECT atperiods(atregion(t.geometry,l.geometry),range(first,last))
    FROM Temp t )
```

Conclusions

- **Spatio-temporal DWs** result from **combining GIS, OLAP, and temporal data types**
  - Temporal data types model geometries that evolve over time (moving objects) and evolving (slowly changing) dimensions
  - Field data types model continuous fields that change in space
  - Temporal fields obtained by composing field and temporal data types
- We defined a **new field data type** and associated operators
- We extended the **MultiDim conceptual model for data warehouses**
- We defined a **taxonomy for spatio-temporal OLAP queries** that
  - characterizes features required by spatio-temporal DWs
  - allows to classify different works addressing this issue in the literature

Future Work

- Our framework is at a **conceptual level**: implementation considerations were omitted
- **From abstract to concrete models**
  - Grid and TIN models for continuous fields
  - Performing efficient OLAP-style navigation of field hierarchies
  - Interpolation of field values across time and space, relating this issue to OLAP
  - . . .
- We are currently **implementing a prototype** based on these ideas
- Spatio-temporal DWs contain huge amounts of data ⇒ **optimization issues are essential**
  - appropriate index structures, pre-aggregation, efficient query optimization, . . .
- Queries can be expressed in several equivalent ways, but evaluation time of these queries may vary significatively

References

- Anthony C. Klug. Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions. *J. ACM* 29(3): 699-717 (1982)
- Ralf Hartmut Güting et al. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.* 25(1): 1-42 (2000)
- Ralf Hartmut Güting, Markus Schneider. *Moving Objects Databases*, Morgan Kaufmann 2005
- Elzbieta Malinowski, Esteban Zimányi. *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer, 2008
- Alejandro A. Vaisman, Esteban Zimányi. A multidimensional model representing continuous fields in spatial data warehouses. *GIS 2009*: 168-177
- Alejandro A. Vaisman, Esteban Zimányi. What Is Spatio-Temporal Data Warehousing? *DaWaK 2009*: 9-23
- Leticia I. Gómez, Alejandro A. Vaisman, Esteban Zimányi. Physical Design and Implementation of Spatial Data Warehouses Supporting Continuous Fields. *DaWak 2010*: 25-39

Mobility Data Management

(Spatio-Temporal Data Warehouses)

Esteban ZIMÁNYI

ULB

Questions ?