Spatio-Temporal Data Management

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

Invited Lectures
Poznań University of Technology
Poznań, Poland, 1-3 October 2012

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Topics

- **Introduction**
- Time Ontology
- Temporal Conceptual Modeling
- Manipulating Temporal Databases with SQL-92
- Temporal Support in SQL 2011
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Introduction

- Applications with temporal aspects abound
  - Academic
  - Accounting
  - Data warehousing
  - Financial
  - Geographical Information Systems
  - Insurance
  - Inventory
  - Law
  - Medical records
  - Reservation systems
  - Scientific databases
  - ...

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Need for a Temporal DBMS

- It is difficult to identify applications not needing management of temporal data
- These applications would benefit from **built-in temporal support** in the DBMS
  - More efficient application development
  - Potential increase of performance
- **Temporal DBMS**: Provide mechanisms to store and manipulate time-varying information

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Case Study

- Personnel management in a database

```
Employee(Name, Salary, Title, BirthDate DATE)
```

- It is easy to know the salary of an employee

```
SELECT Salary
FROM Employee
WHERE Name = 'John'
```

- It is also easy to know the date of birth of an employee

```
SELECT BirthDate
FROM Employee
WHERE Name = 'John'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Converting to a Temporal Database

- We want to keep the employment history

`Employee(Name, Salary, Title, BirthDate, FromDate DATE, ToDate DATE)`

Name	Salary	Title	BirthDate	FromDate	ToDate
John	60.000	Assistant	9/9/60	1/1/95	1/6/95
John	70.000	Assistant	9/9/60	1/6/95	1/10/95
John	70.000	Lecturer	9/9/60	1/10/95	1/2/96
John	70.000	Professor	9/9/60	1/2/96	1/1/97

- For the data model, new columns are identical to attribute `BirthDate`

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Determine the Salary

- To know the employee's current salary, things are more difficult

```
SELECT Salary
FROM Employee
WHERE Name = 'John' AND FromDate <= CURRENT_TIMESTAMP
      AND CURRENT_TIMESTAMP <= ToDate
```

- Determine the salary history
  - Result: for each person, the maximal intervals of each salary

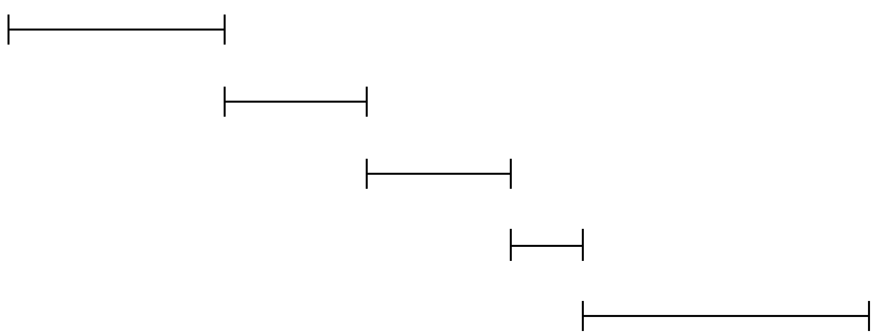
Name	Salary	FromDate	ToDate
John	60.000	1/1/95	1/6/95
John	70.000	1/6/95	1/1/97

  - An employee could have arbitrarily many title changes between salary changes

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Determine the Salary, cont.

- Alternative 1
  - Give the user a printout of `Salary` and `Title` information, and have the user determine when his/her salary changed
- Alternative 2
  - Use SQL as much as possible
  - Find those intervals that overlap or are adjacent and that should be merged



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SQL Code

```
CREATE TABLE Temp(Salary, FromDate, ToDate) AS
  SELECT Salary, FromDate, ToDate
  FROM Employee
  WHERE Name = 'John'

repeat
      UPDATE Temp T1
      SET (T1.ToDate) = (SELECT MAX(T2.ToDate)
                         FROM Temp AS T2
                         WHERE T1.Salary = T2.Salary
                           AND T1.FromDate < T2.FromDate
                           AND T1.ToDate >= T2.FromDate
                           AND T1.ToDate < T2.ToDate)
      WHERE EXISTS ( SELECT *
                     FROM Temp as T2
                     WHERE T1.Salary = T2.Salary
                       AND T1.FromDate < T2.FromDate
                       AND T1.ToDate >= T2.FromDate
                       AND T1.ToDate < T2.ToDate)
until no tuples updated
```

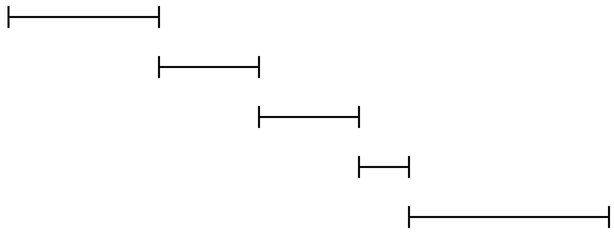
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

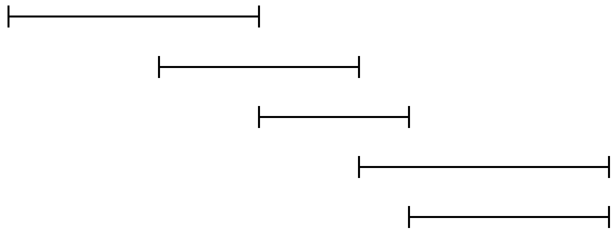
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SQL Code, cont.

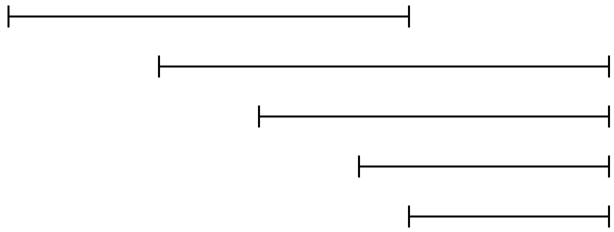
- Initial table



- After one pass



- After two passes



SQL Code, cont.

- Loop is executed logN times in the worst case, where N is the number of tuples in a chain of overlapping or adjacent value-equivalent tuples
- Then delete extraneous, non-maximal intervals

```
DELETE FROM Temp T1
WHERE EXISTS (
    SELECT *
    FROM Temp AS T2
    WHERE T1.Salary = T2.Salary
    AND ( (T1.FromDate > T2.FromDate AND T1.ToDate <= T2.ToDate)
     OR   (T1.FromDate >= T2.FromDate AND T1.ToDate < T2.ToDate) )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Same Functionality Entirely in SQL

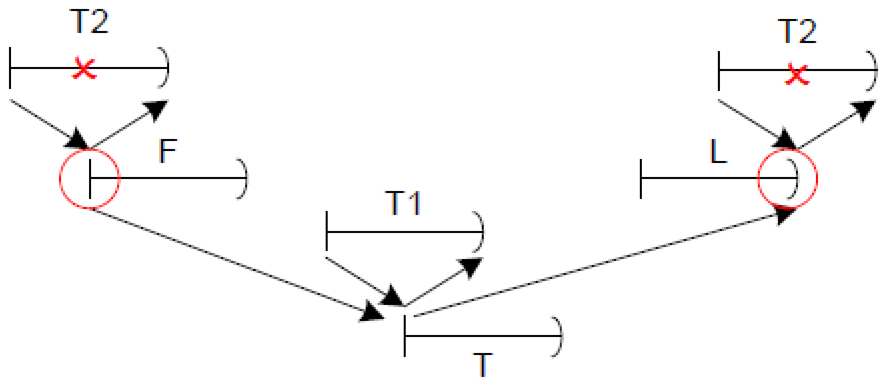
```
CREATE VIEW Temp(Salary, FromDate, ToDate) AS
  SELECT Salary, FromDate, ToDate
  FROM Employee
  WHERE Name = 'John'

SELECT DISTINCT F.Salary, F.FromDate, L.ToDate
FROM Temp AS F, Temp AS L
WHERE F.FromDate < L.ToDate AND F.Salary = L.Salary
AND NOT EXISTS (
    SELECT *
    FROM Temp AS T
    WHERE T.Salary =  F.Salary
    AND F.FromDate < T.FromDate AND T.FromDate < L.ToDate
    AND NOT EXISTS (
        SELECT *
        FROM Temp AS T1
        WHERE T1.Salary =  F.Salary
        AND T1.FromDate < T.FromDate AND T.FromDate <= T1.ToDate) )
AND NOT EXISTS (
    SELECT *
    FROM Temp AS T2
    WHERE T2.Salary = F.Salary
    AND ( (T2.FromDate < F.FromDate AND F.FromDate <= T2.ToDate)
    OR    (T2.FromDate <= L.ToDate AND L.ToDate < T2.ToDate)))
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Same Query in Tuple Relational Calculus



$$
\begin{aligned}
\{ & f.\text{FromDate}, l.\text{ToDate} \mid \\
& \text{Temp}(f) \land \text{Temp}(l) \land f.\text{FromDate} < l.\text{ToDate} \land f.\text{Salary} = l.\text{Salary} \land \\
& (\forall t)(\text{Temp}(t) \land t.\text{Salary} = f.\text{Salary} \land f.\text{FromDate} < t.\text{FromDate} \land \\
& \qquad t.\text{FromDate} < l.\text{ToDate} \rightarrow \\
& \qquad (\exists t_1)(\text{Temp}(t_1) \land t_1.\text{Salary} = f.\text{Salary} \land t_1.\text{FromDate} < t.\text{FromDate} \land \\
& \qquad\qquad t.\text{FromDate} <= t_1.\text{ToDate})) \land \\
& \neg(\exists t_2)(\text{Temp}(t_2) \land t_2.\text{Salary} = f.\text{Salary} \land \\
& \qquad ((t_2.\text{FromDate} < f.\text{FromDate} \land f.\text{FromDate} <= t_2.\text{ToDate}) \lor \\
& \qquad\; (t_2.\text{FromDate} <= l.\text{ToDate} \land l.\text{ToDate} < t_2.\text{ToDate}))) \}
\end{aligned}
$$

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Other Possibilities

- Use the transitive closure or triggers in SQL3
- TSQL2

```
SELECT Salary
FROM Employee
WHERE Name = 'Bob'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Alternative: Reorganize the Schema

- Split the information on Salary, Title, and BirthDate

```
Employee(Name, BirthDate DATE)
EmployeeSal(Name, Salary, FromDate DATE, ToDate DATE)
EmployeeTitle(Name, Title, FromDate DATE, ToDate DATE)
```

- Determine the information about the salary is easy now

```
SELECT Salary, FromDate, ToDate
FROM EmployeeSal
WHERE Name = 'John'
```

- However, how to obtain a table of salary, title intervals?

Example of Temporal Join

EmployeeSal

Name	Salary	FromDate	ToDate
John	60.000	1/1/95	1/6/95
John	70.000	1/6/95	1/1/97

EmployeeTitle

Name	Title	FromDate	ToDate
John	Assistant	1/1/95	1/10/95
John	Lecturer	1/10/95	1/2/96
John	Professor	1/2/96	1/1/97

EmployeeSal $\bowtie$ EmployeeTitle

Name	Salary	Title	FromDate	ToDate
John	60.000	Assistant	1/1/95	1/6/95
John	70.000	Assistant	1/6/95	1/10/95
John	70.000	Lecturer	1/10/95	1/2/96
John	70.000	Professor	1/2/96	1/1/97

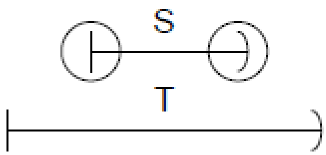
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Evaluation of Temporal Join

- Alternative 1: Print the two tables and leave the user make the combinations
- Alternative 2: Use SQL entirely

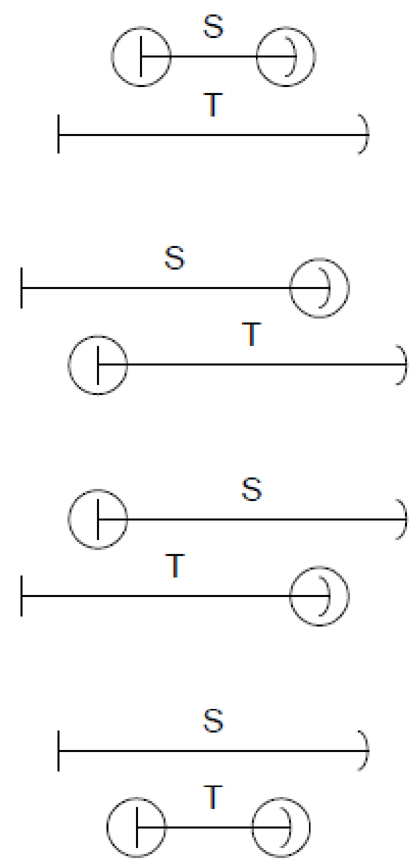



```
SELECT S.Name, Salary, Title, S.FromDate, S.ToDate
FROM EmployeeSal S, EmployeeTitle T
WHERE S.Name = T.Name
  AND T.FromDate <= S.FromDate
  AND S.ToDate < T.ToDate
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Join in SQL

```
SELECT S.Name, Salary, Title, S.FromDate, S.ToDate
FROM EmployeeSal S, EmployeeTitle T
WHERE S.Name = T.Name
  AND T.FromDate <= S.FromDate
  AND S.ToDate <= T.ToDate
UNION ALL
SELECT S.Name, Salary, Title, S.FromDate, T.ToDate
FROM EmployeeSal S, EmployeeTitle T
WHERE S.Name = T.Name
  AND S.FromDate > T.FromDate
  AND T.ToDate < S.ToDate
  AND S.FromDate < T.ToDate
UNION ALL
SELECT S.Name, Salary, Title, T.FromDate, S.ToDate
FROM EmployeeSal S, EmployeeTitle T
WHERE S.Name = T.Name
  AND T.FromDate > S.FromDate
  AND S.ToDate < T.ToDate
  AND T.FromDate < S.ToDate
UNION ALL
SELECT S.Name, Salary, Title, T.FromDate, T.ToDate
FROM EmployeeSal S, EmployeeTitle T
WHERE S.Name = T.Name
  AND T.FromDate >= S.FromDate
  AND T.ToDate <= S.ToDate
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Join, cont.

- Alternative 3: Use embedded SQL
- TSQL2: Give the salary and title history of employees

```
SELECT EmployeeSal.Name, Salary, Title
FROM EmployeeSal, EmployeeTitle
WHERE EmployeeSal.Name = EmployeeTitle.Name
```


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Introduction: Summary

- Applications managing temporal data abound
- Classical DBMS are not adequate
- If a temporal DBMS is used
  - Schemas are simpler
  - SQL queries are much simpler
  - Much less procedural code is necessary
- Benefits
  - Application code is less complex
    - Easier to understand, to produce, to ensure correctness, to maintain
  - Performance may be increased by relegating functionality to DBMS

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Topics

- Introduction
- **Time Ontology**
- Temporal Conceptual Modeling
- Manipulating Temporal Databases with SQL-92
- Temporal Support in SQL 2011
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Time Ontology

- Notions of time
  - Structure
  - Density
  - Boundedness
- TSQL2 time ontology
- Time data types
- Times and facts

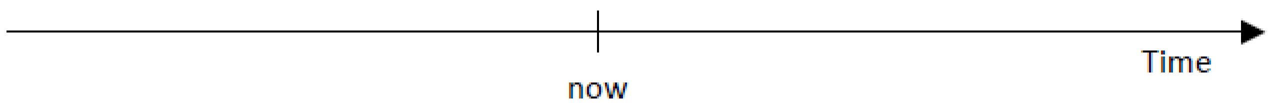
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

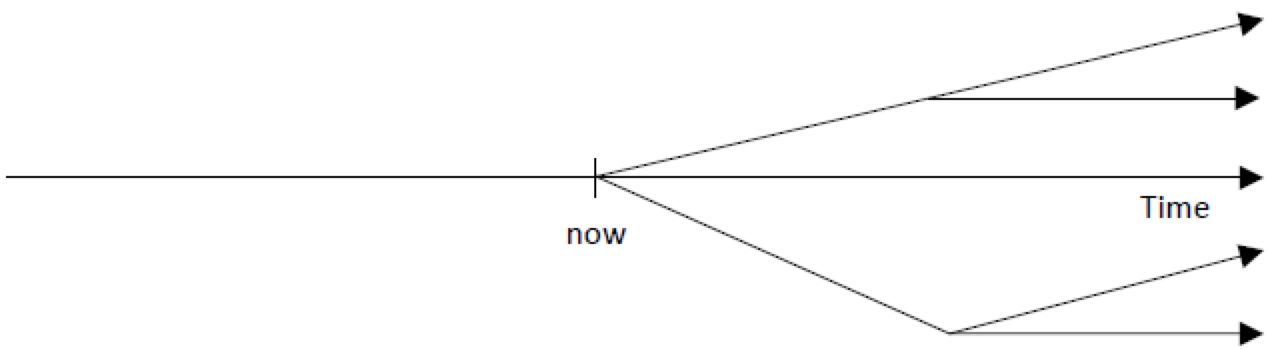
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Time Structure

- Linear: total order on instants



- Hypothetical (possible futures): tree rooted on now



- Directed Acyclic Graph (DAG): possible futures may merge
- Periodic/cyclic time: weeks, months, . . ., for recurrent processes

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Boundedness of Time

- Assume a linear time structure
- Boundedness
  - Unbounded
  - Time origin exists (bounded from the left)
  - Bounded time (bounds on two ends)
- Nature of bound
  - Unspecified
  - Specified
- Physicists believe that the universe is bounded by the "Big Bang" (12-18 billions years ago) and by the "Big Crunch" (? billion years in the future)

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

POLITECHNIKA POZNAŃSKA

Time Density

- **Discrete**
  - Time line is isomorphic to the integers
  - Time line is composed of a sequence of non-decomposable time periods, of some fixed minimal duration, termed **chronons**
  - Between each pair of chronons is a finite number of other chronons
- **Dense**
  - Time line is isomorphic to the rational numbers
  - Infinite number of instants between each pair of chronons
- **Continuous**
  - Time line is isomorphic to the real numbers
  - Infinite number of instants between each pair of chronons
- **Distance** may optionally be defined

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

TSQL2: Time Ontology

- Structure
  - TSQL2 uses a linear time structure
- Boundedness
  - TSQL2 time line is bounded on both ends, from the start of time to a point far in the future
- Density
  - TSQL2 do not differentiate between discrete, dense, and continuous time ontologies
  - No questions can be asked that give different answers
    - E.g., instant a precedes instant b at some specified granularity. Different granularities give different answers
  - Distance is defined in terms of numbers of chronons

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Ontological Temporal Types

- **Instant**: chronon in the time line
  - **Event**: instantaneous fact, something occurring at an instant
  - **Event occurrence time**: valid-time instant at which the event occurs in the real world
- **Instant Set**: set of instants
- **Time period**: time between two instants
  - Also called interval, but conflicts with SQL data type `INTERVAL`
- **Time interval**: a directed duration of time
- **Duration**: amount of time with a known length, but no specific starting or ending instants
  - **positive interval**: forward motion time
  - **negative interval**: backward motion time
- **Temporal element**: finite union of periods

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Representing Time in TSQL2

- TSQL2 supports a bounded discrete representation of the time line
- Time line composed of chronons, which is the smallest granularity
- Consecutive chronons may be grouped together into granules, yielding multiple granularities
- Different granularities are available, and it is possible to convert from one granularity to another (via scaling)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Data Types in SQL-92 and TSQL2

- SQL92
  - DATE (YYYY-MM-DD)
  - TIME (HH:MM:SS)
  - DATETIME (YYYY-MM-DD HH:MM:SS)
  - INTERVAL (no default granularity)
- TSQL2
  - PERIOD: DATETIME - DATETIME

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Time and Facts

- **Valid time** of a fact: when the fact is true in the modeled reality
  - Independently of its recording in the database
  - Past, present, future
- **Transaction time** of a fact: when the fact is current in the database and may be retrieved
  - Identify the transactions that inserted and deleted the fact
- Two dimensions are orthogonal
- Four kinds of tables
  - Snapshot
  - Valid time
  - Transaction time
  - Bitemporal

Snapshot Tables

- May be modified
- Used for static queries
- What is John's title?

```
SELECT Title
FROM Faculty
WHERE Name = 'John'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Snapshot Tables, cont.

- Analogy: Nameplate on door

John Assistant	John 1st Assistant	John 1st Assistant	John Lecturer
Jan. 84	Dec. 87	March 89	July 89

- On January 1st, 1984, John is hired as assistant
- On December 1st, 1987, John finishes his doctorate and is promoted as 1st Assistant retroactively on July 1st, 1987
- On March 1st, 1989, John is promoted as Lecturer, proactively on July 1st, 1989

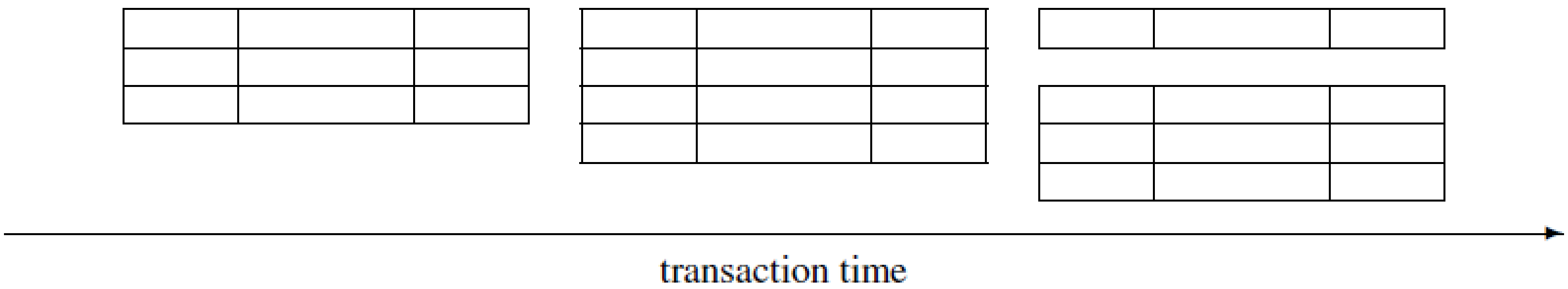
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

POLITECHNIKA POZNAŃSKA

Transaction Time Tables



- Append-only: correction to previous snapshot states is not permitted
- Allow retrospective queries (“rollback”)
- What did we believe John’s rank was on October 1st, 1984?

```
SELECT Title
FROM Faculty
WHERE Name = 'John' AND
      TRANSACTION(Faculty) OVERLAPS DATE '01-10-1984'
```

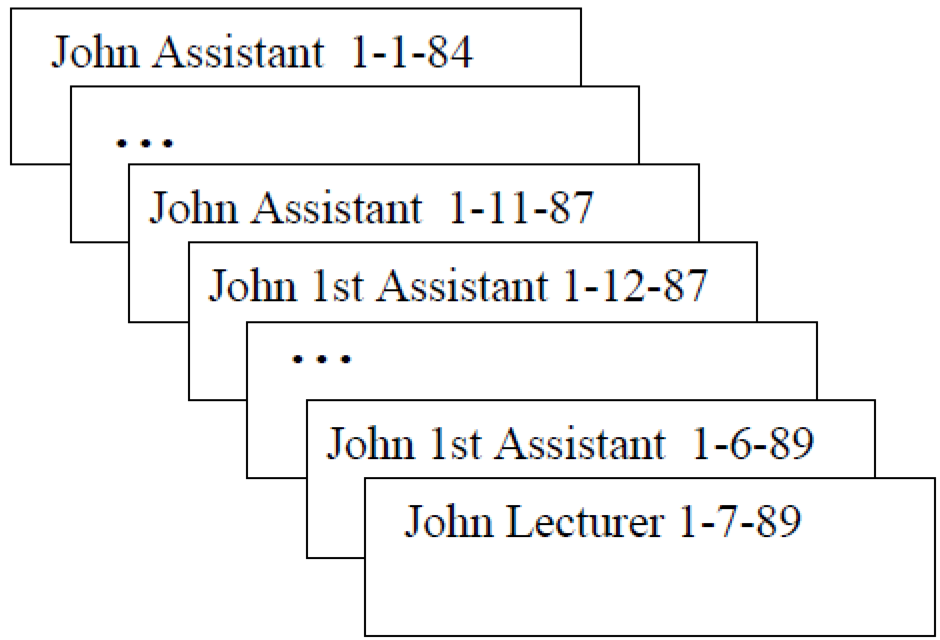
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Transaction Time Tables, cont.

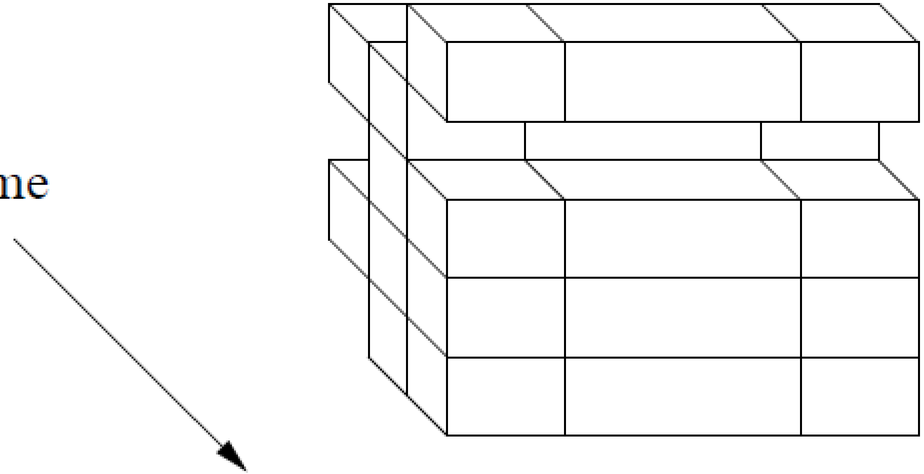
- Analogy: Pay stubs



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Valid Time Tables

Valid Time



- May be modified
- Allow historical queries
- What was John's title on October 1st, 1984 (as best known)?

```
SELECT Title
FROM Faculty
WHERE Name = 'John' AND
      VALID(Faculty) OVERLAPS DATE '01-10-1984'
```

Valid Time Tables, cont.

- Analogy: Curriculum Vitæ

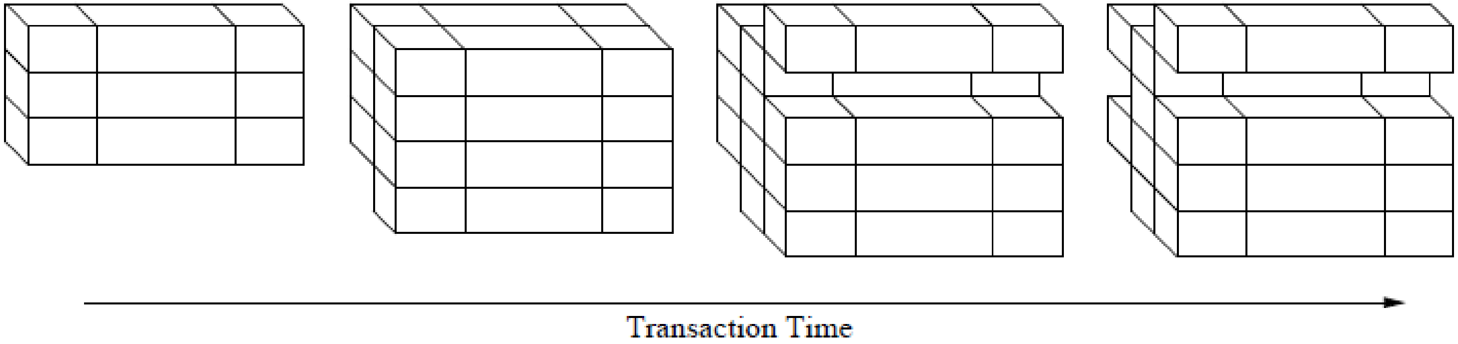
John	
Titles	
Lecturer	July 1989
1st Assistant	July 1987
Assistant	January 1984

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Bitemporal Tables



- Append-only
- Transaction and valid time
- Allow coupled historical and retrospective queries
- On October 1st, 1984, what did we think John's rank was at that date?

```
SELECT Title
FROM Faculty AS E
WHERE Name = 'John' AND
      VALID(E) OVERLAPS DATE '01-10-1984' AND
      TRANSACTION(E) OVERLAPS DATE '01-10-1984'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Bitemporal Tables, cont.

- Analogy: Stack of CVs

January 1984 Version

John
Titles
Assistant January 1984

December 1987 Version

John
Titles
1st Assistant July 1987
Assistant January 1984

March 1989 Version

John
Titles
Lecturer July 1989
1st Assistant July 1987
Assistant January 1984

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Time Ontology: Summary

- Several different structures of time
  - Linear is simplest and most common
- 5 fundamental temporal data types
- Several dimensions of time
  - TSQL2 supports transaction and valid time

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Topics

- Introduction
- Time Ontology
- ➡ **Temporal Conceptual Modeling**
- Manipulating Temporal Databases with SQL-92
- Temporal Support in SQL 2011
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

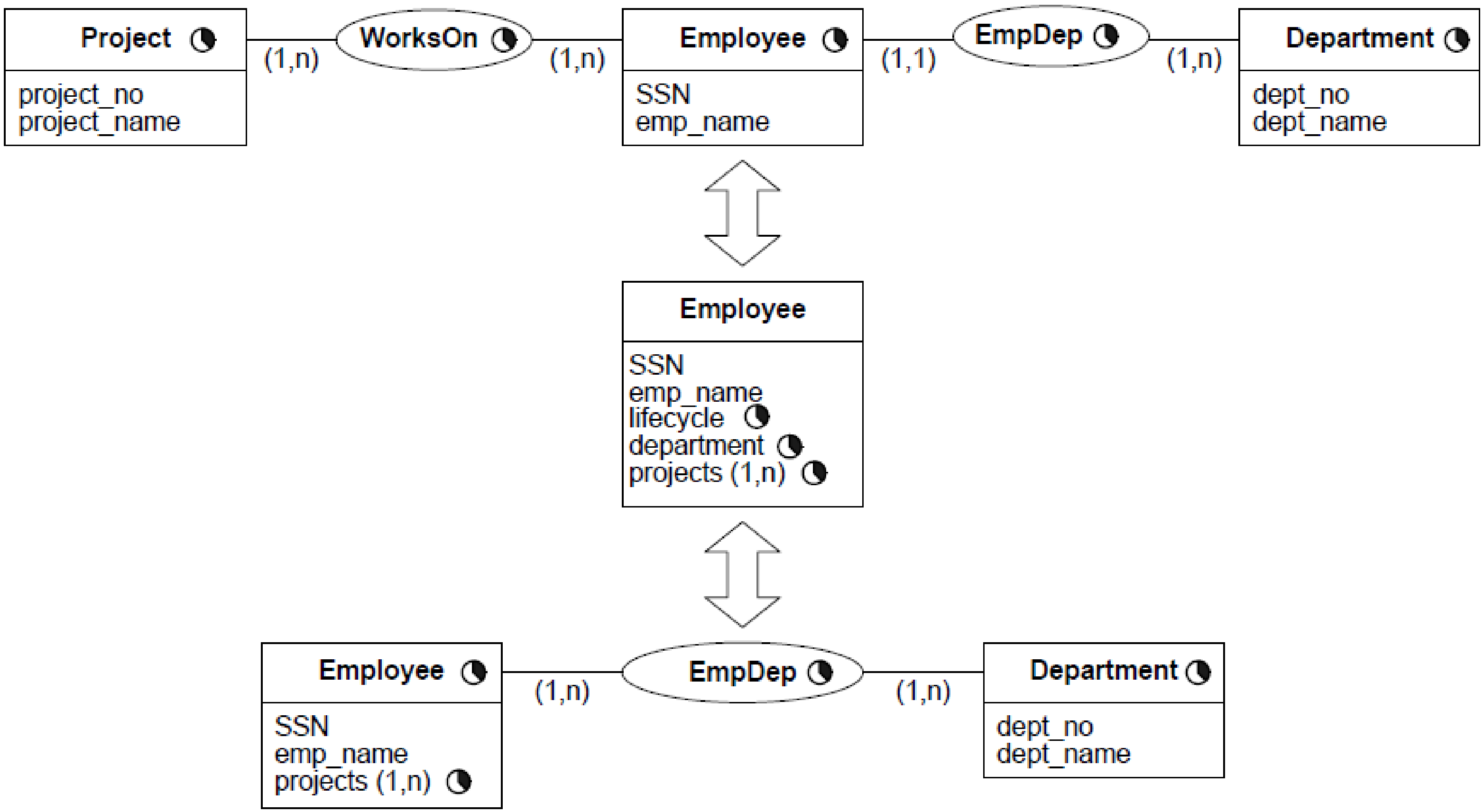
Why Conceptual Modeling ?

- Focuses on the application
- Technology independent
  - portability, durability
- User oriented
- Formal, unambiguous specification
- Supports visual interfaces
  - data definition and manipulation
- Best vehicle for information exchange/integration

The Conceptual Manifesto (1)

- Semantically powerful data structures
- Simple (understandable) data model
  - few clean concepts, with standard, well-known semantics
- No artificial time objects
- Time orthogonal to data structures
- Various granularities
- Clean, visual notations
- Intuitive icons / symbols

Orthogonality



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

The Conceptual Manifesto (2)

- Explicit temporal relationships and integrity constraints
- Support of valid time and transaction time
- Past to future
- Co-existence of temporal and traditional data
- Query languages
- Complete and precise definition of the model

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Information Describes ...

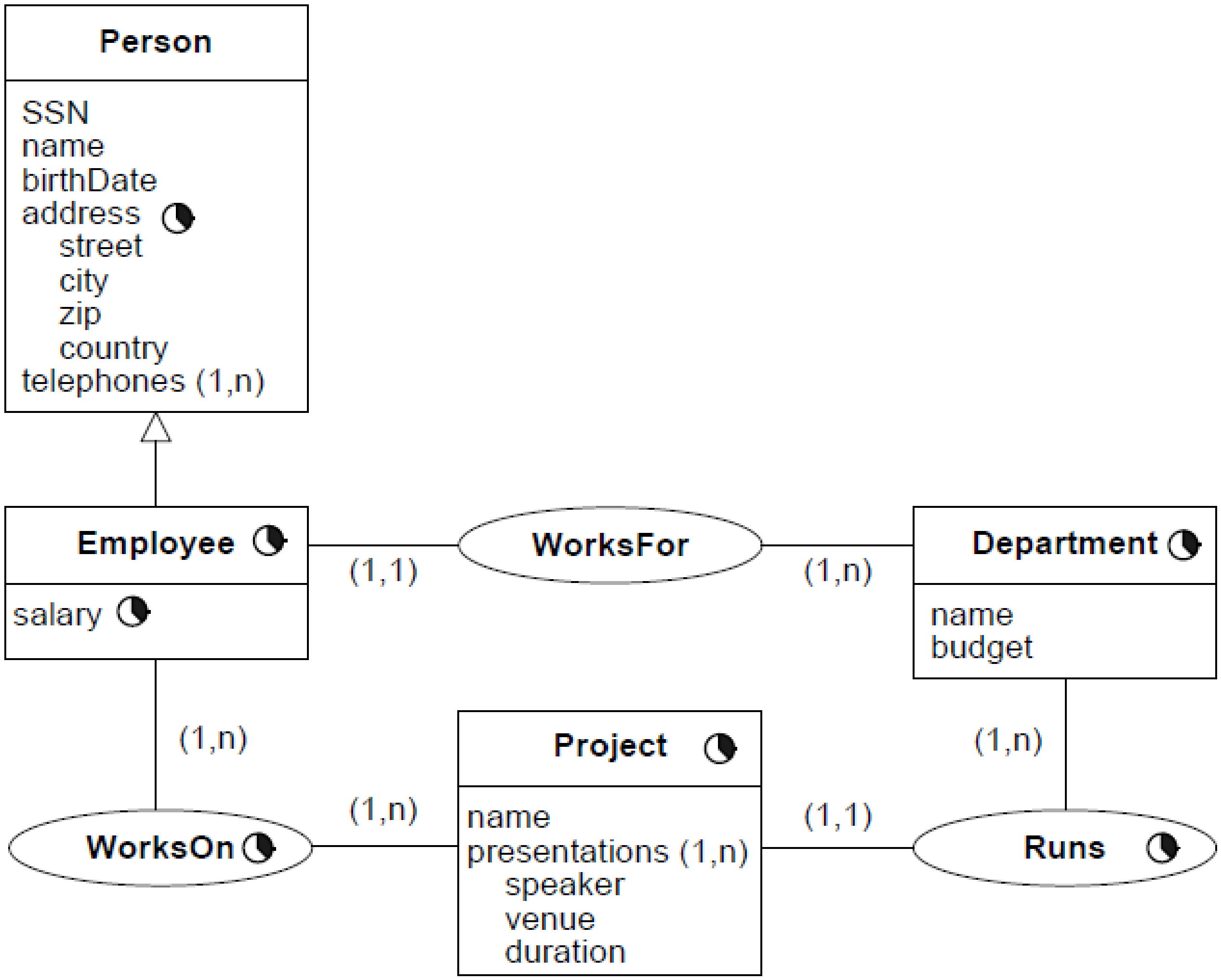
- Life cycles of objects and relationships
- Validity of information values
  - Timestamps
- Temporal relationships
  - Temporal links
  - Temporal integrity constraints

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Schema: Example

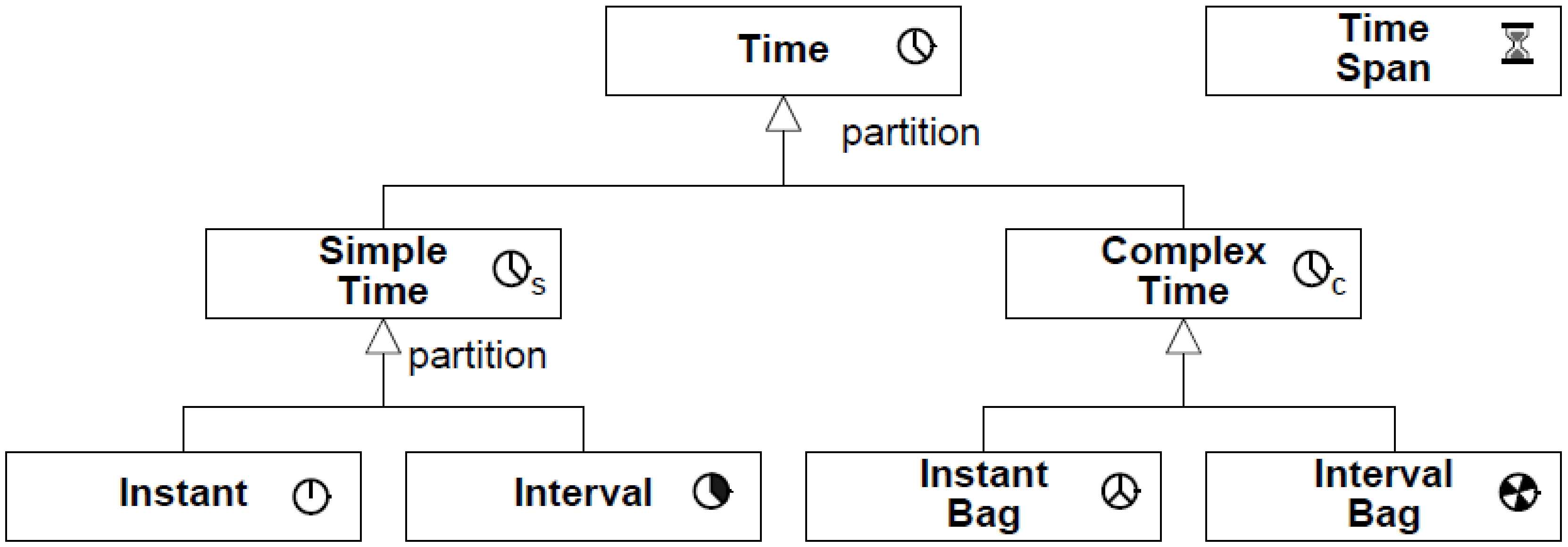


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

MADS Temporal Data Types



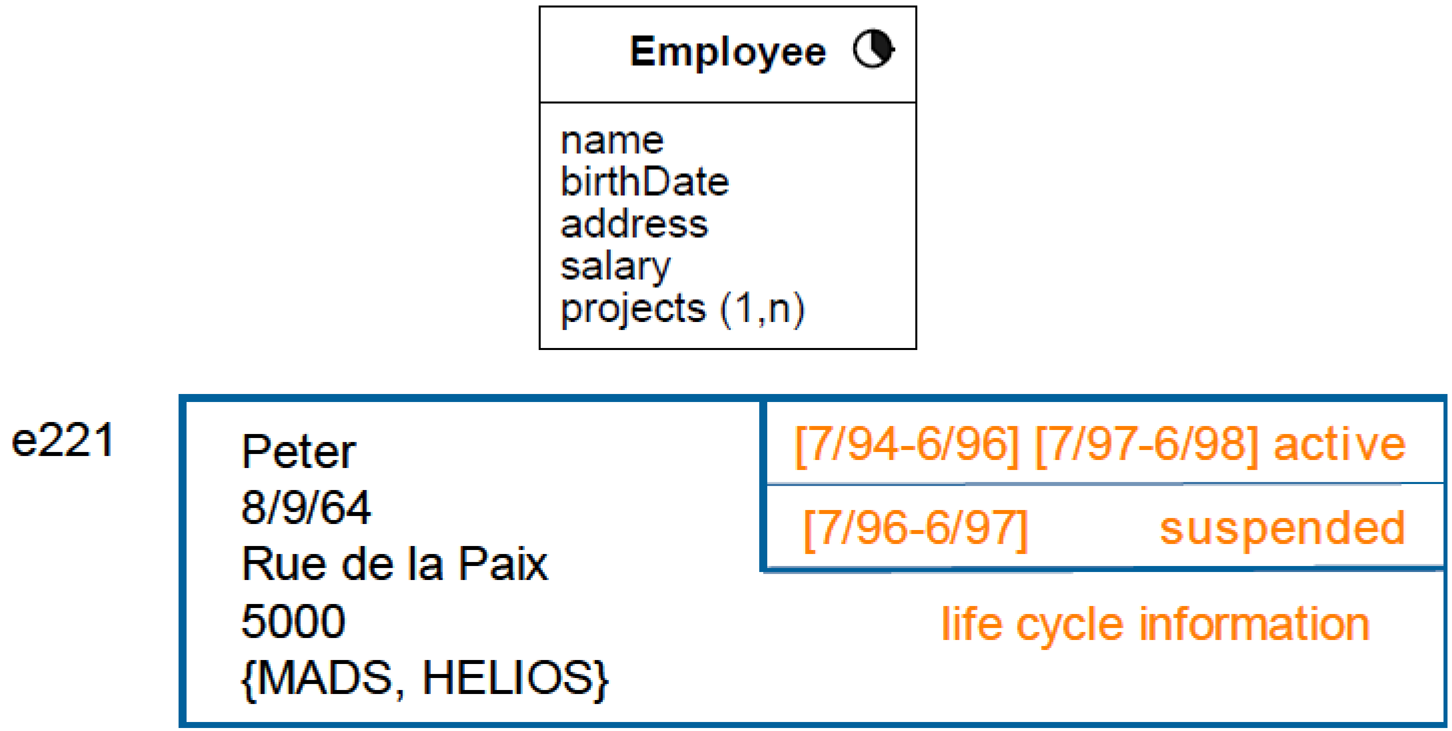
- `Time`, `SimpleTime`, and `ComplexTime` are abstract classes

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

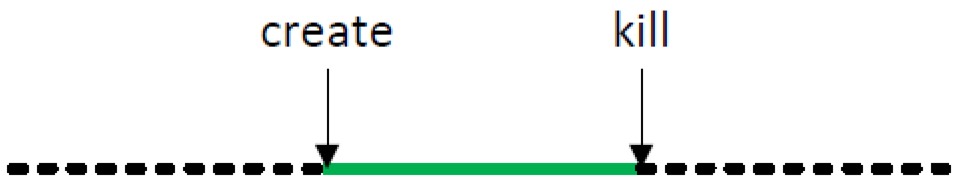
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Objects

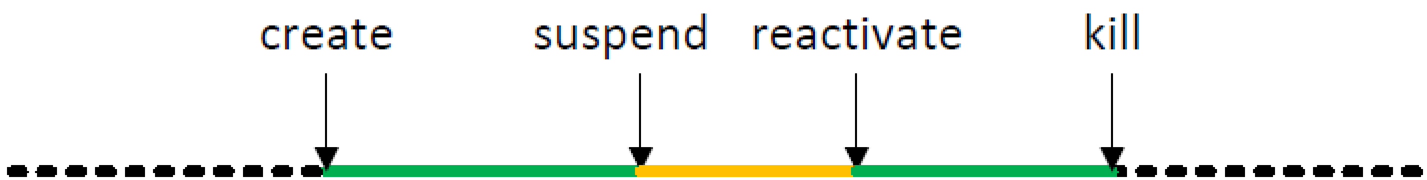


Object / Relationship Life Cycle

- Continuous



- Discontinuous



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Non-Temporal Objects ?

- No life cycle, or
- Default life cycle
  - active → [0, now]
  - active → [now, now]
  - active → [0, ∞]
- Coexistence
  - temporal → non temporal (snapshot)
  - non-temporal → temporal (default life cycle)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

TSQL2 Policy

- Temporal operators not allowed on non-temporal relations
  - no life cycle
- Joins between temporal and non-temporal relations are allowed
  - default life cycle: active → $[0, \infty]$

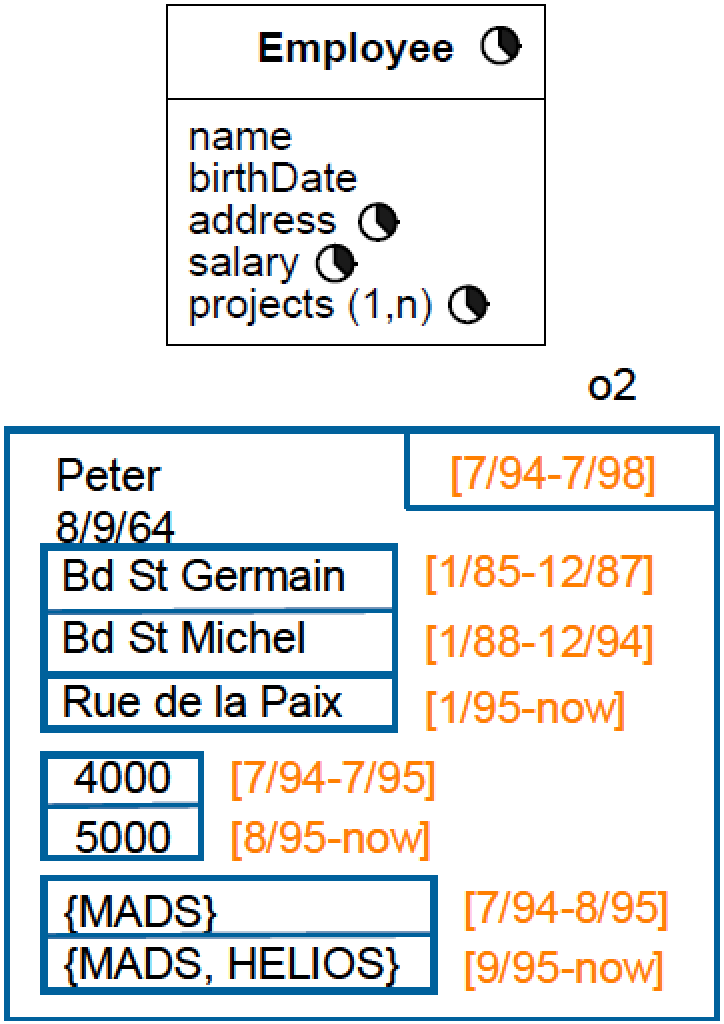
```
SELECT Department.Name, COUNT (PID)
FROM Department, Employee
WHERE Employee.dept # = Department.dept #
   AND VALID(Employee) OVERLAPS PERIOD '[1/1/96-31/12/96]'
GROUP BY dept #
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Attributes

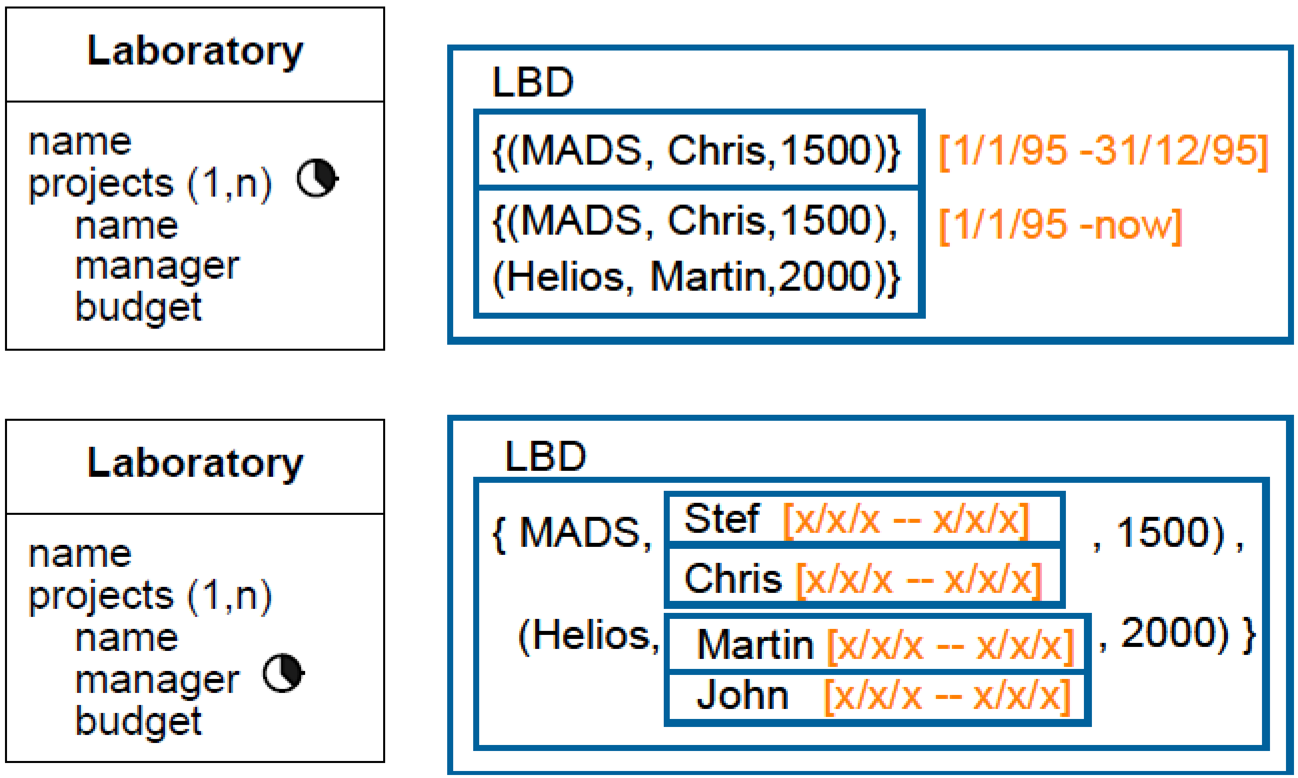


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

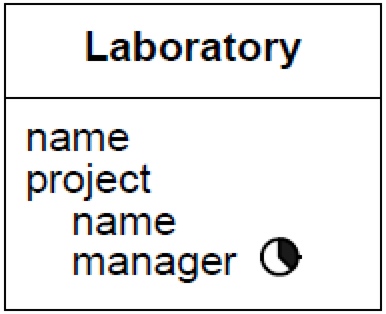
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Complex Attributes (1)



Temporal Complex Attributes (2)



- "Updating" manager ⇒ add element to manager history
- "Updating" projName (name of project has changed) ⇒ update name
- "Updating" project (laboratory changed project) ⇒ update name, start new history for manager

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Attribute Timestamping Properties

- Attribute types / timestamping
  - none, irregular, regular, instants, durations, ...
- Cardinalities
  - snapshot and DBlifespan
- Identifiers
  - snapshot or DBlifespan

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Attribute Timestamping Issues

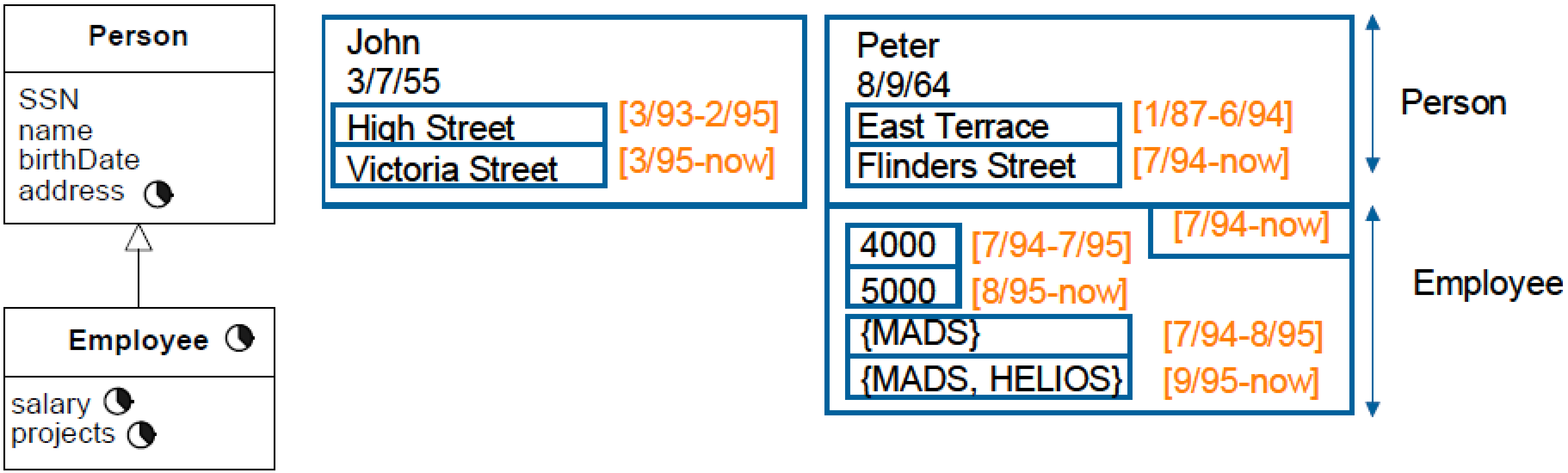
- Constraints?
  - the validity period of an attribute must be within the life cycle of the object it belongs to
  - the validity period of a complex attribute is the union of the validity periods of its components
- MADS : no implicit constraint

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

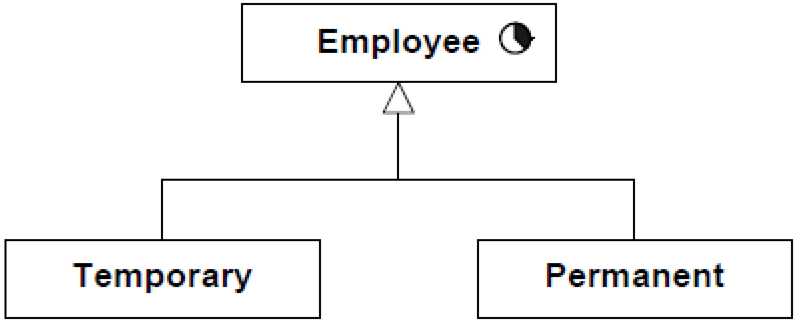
Temporal Generalization



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

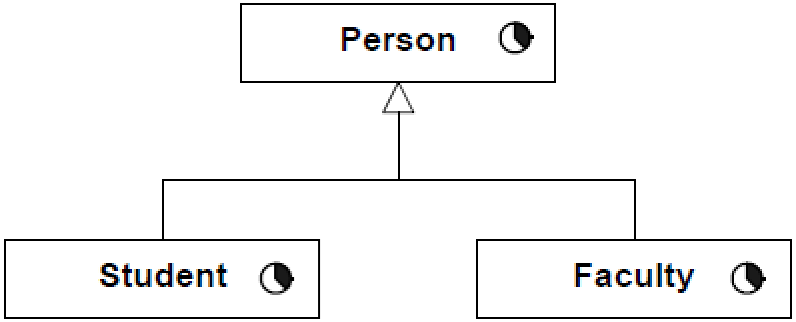
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Static Temporal Generalization



- Temporary and Permanent are implicitly temporal
  - they inherit their life cycle from Employee

Dynamic Temporal Generalization



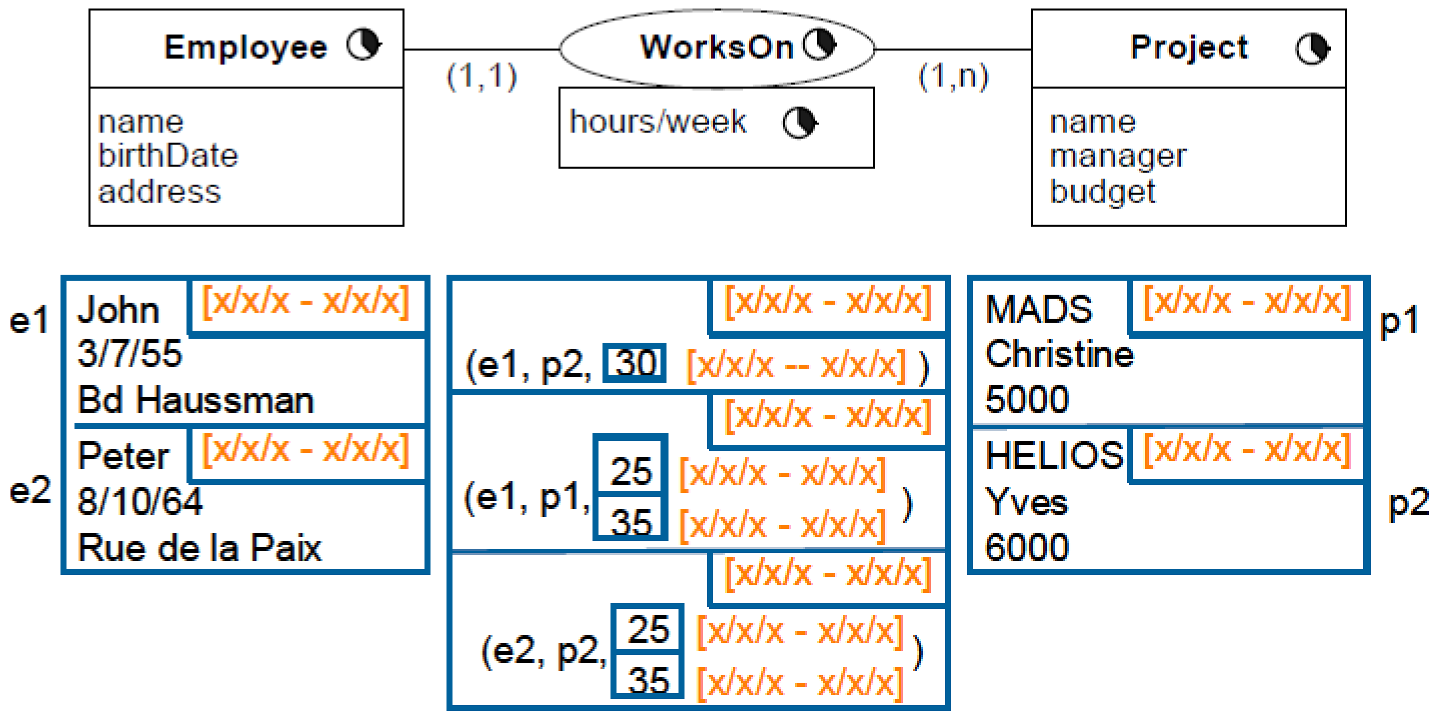
- Student and Faculty have two life cycles:
  - an inherited one (the one of Person)
  - a redefined one (the one of Student/Faculty)
- The redefined life cycle has to be included in the one of the corresponding Person
  - lifespan and active periods

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Relationships (1)



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Relationship Timestamping Issues

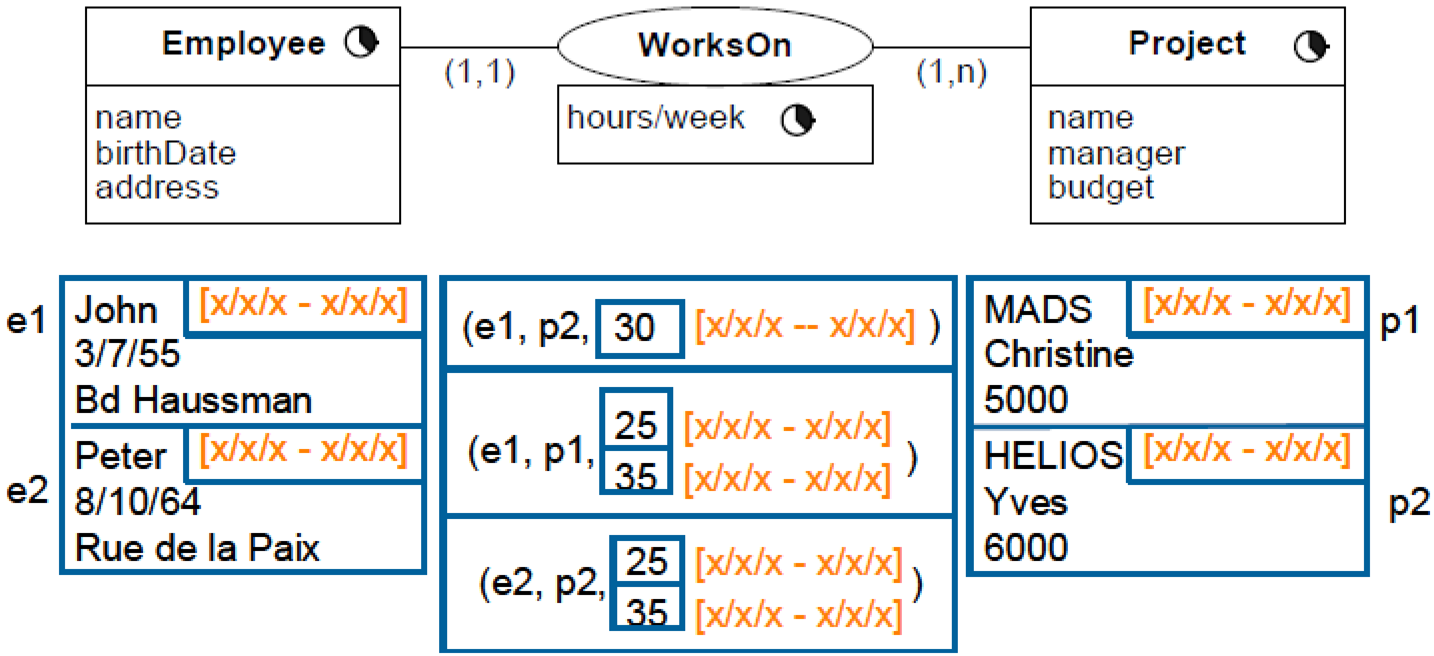
- Constraints?
  - the validity period of a relationship must be within the intersection of the life cycles of the objects it links
  - a temporal relationship can only link temporal objects
- MADS : no implicit constraint

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Relationships (2)



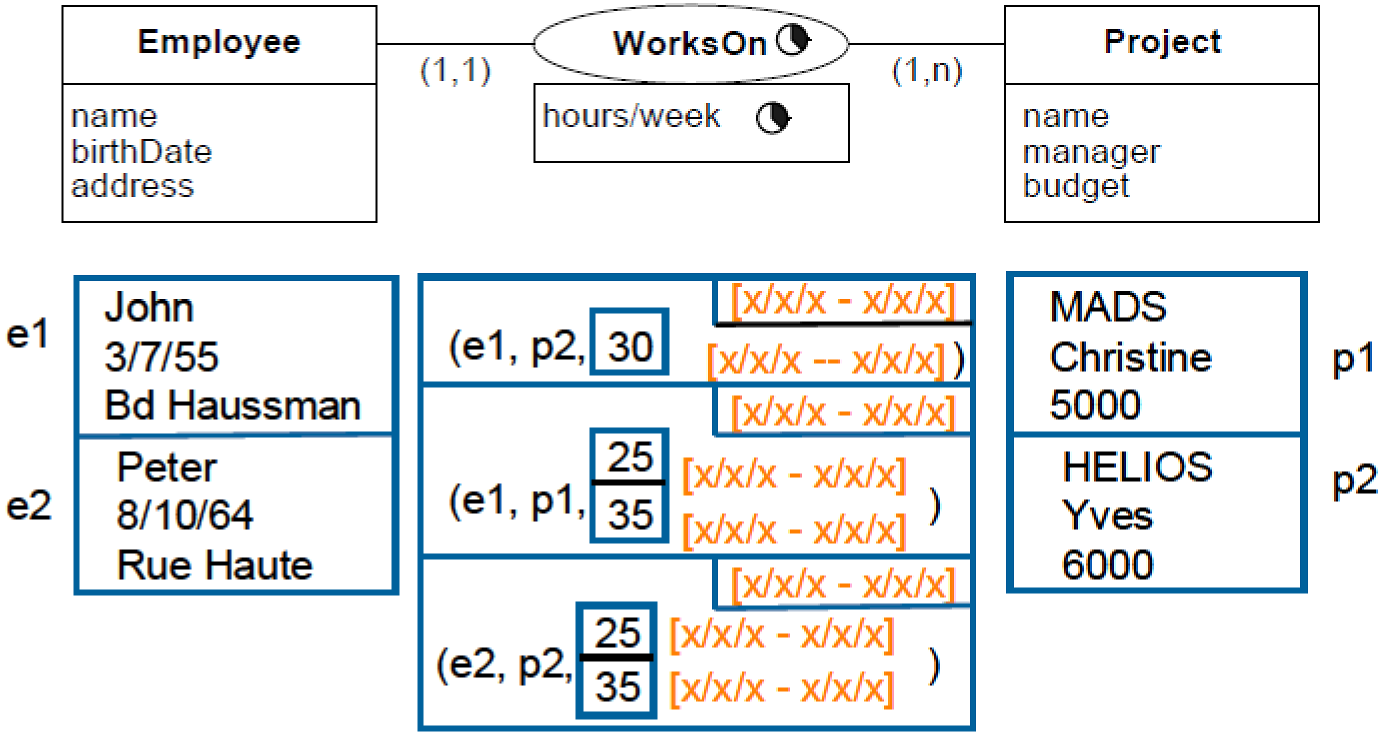
- Only **currently valid couples** are kept in the relationship

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Relationships (3)



- Only **currently valid objects** participate in the relationship

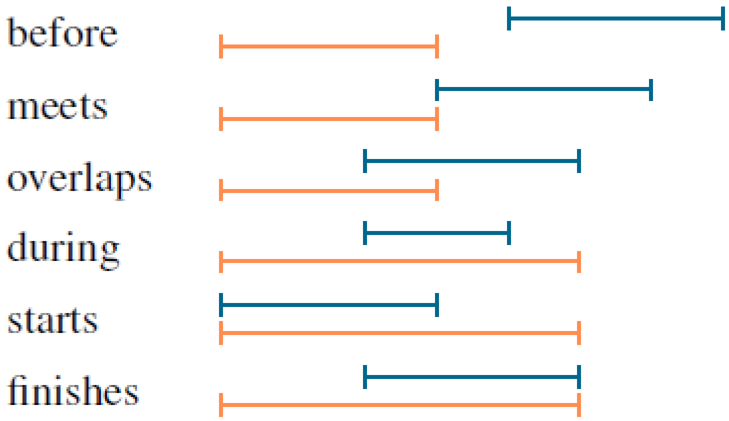
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

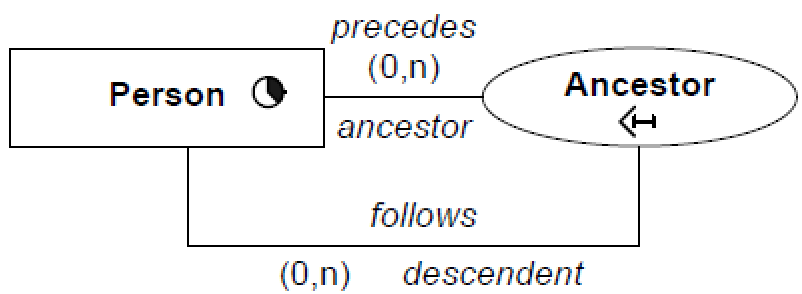
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Synchronization Relationships (1)

- Describe temporal constraints between the life cycles of two objects
- Expressed with Allen's operator extended for temporal elements



Synchronization Relationships (2)



Synchronization Relationship	Icon	Synchronization Relationship	Icon
SyncGeneric		SyncStart	
SyncDisjoint		SyncFinishes	
SyncOverlap		SyncEqual	
SyncWithin		SyncPrecede	
SyncMeet		SyncFollow	

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Synchronization Relationships (3)

- Express a temporal constraint between
  - the whole life cycles, or
  - the active periods
- Temporal constraint defined with
  - extended Allen's operators
  - application-defined operators, e.g., 9 months later
- They are relationships
  - may have attributes, cardinalities

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Conceptual Models: Conclusion

- Conceptual models must be extended with temporal features
- Orthogonality is the answer for achieving maximal expressive power
- Semantics of temporal features must be explicitly defined
- This semantics generalizes that of the traditional conceptual models
- Temporal conceptual models are easily understood by users

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Topics

- Introduction
- Time Ontology
- Temporal Conceptual Modeling
- **Manipulating Temporal Databases with SQL-92**
- Temporal Support in SQL 2011
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Defining Valid-Time Tables in SQL

Employee

SSN	FirstName	LastName	BirthDate

Position

PCN	JobTitle

Incumbents

SSN	PCN	FromDate	ToDate

Salary

SSN	Amount	FromDate	ToDate

- Incumbents and Salary are valid-time tables
  - FromDate indicates when the information in the row is valid, i.e. when the employee was assigned to that position
  - ToDate indicates when the information in the row was no longer valid
- Data type for periods is not available in SQL-92 $\Rightarrow$ a period is simulated with two Date columns

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

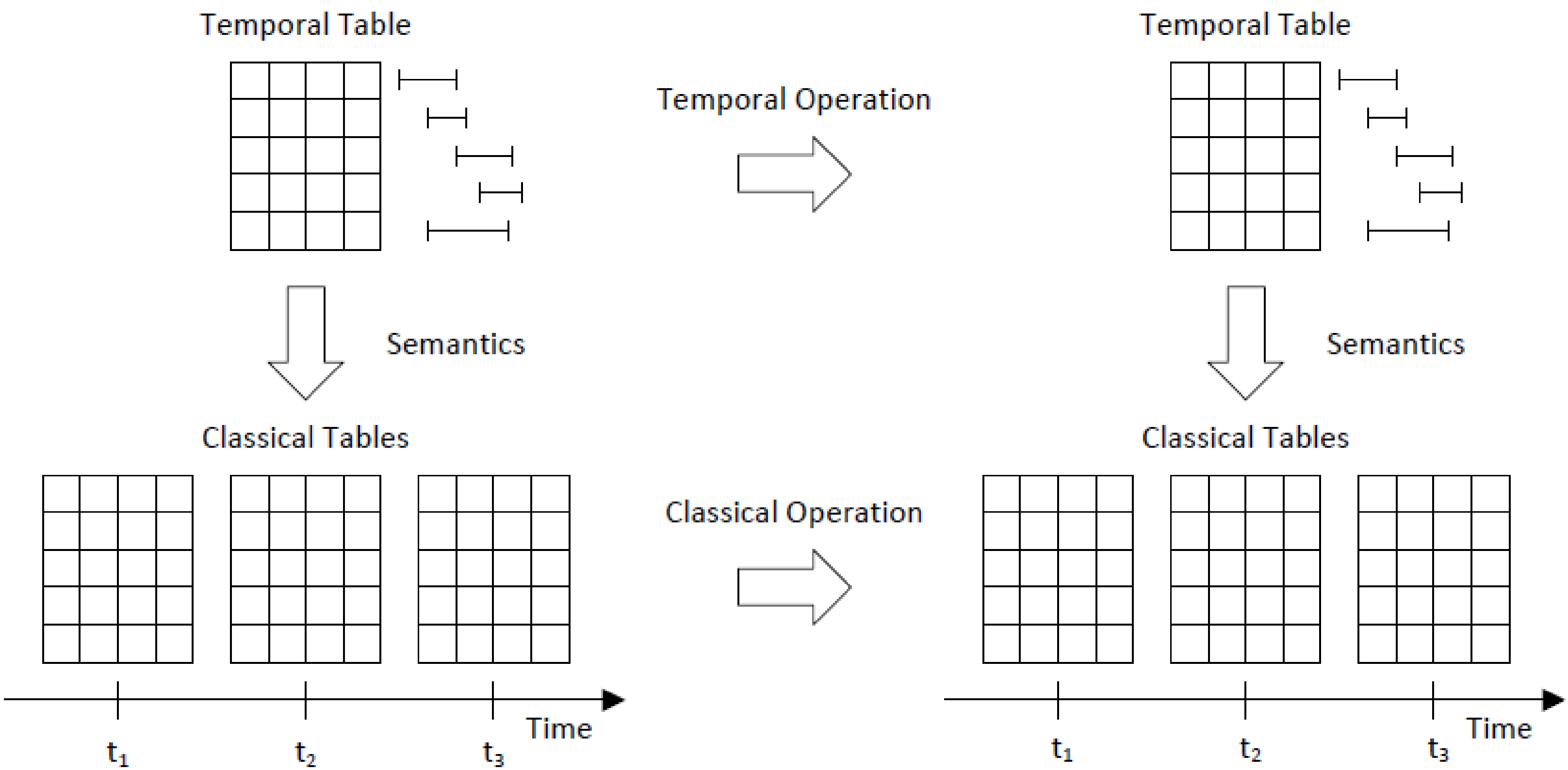
Example of a Valid-Time Table

Incumbents

SSN	PCN	FromDate	ToDate
111223333	900225	1996-01-01	1996-06-01
111223333	900225	1996-06-01	1996-08-01
111223333	900225	1996-08-01	1996-10-01
111223333	900225	1996-10-01	3000-01-01
111223333	900225	1997-01-01	3000-01-01

- Special date '3000-01-01' denotes currently valid
- Closed-open periods used, e.g., validity of first tuple is [1996-01-01,1996-06-01)
- Table can be viewed as a compact representation of a sequence of snapshot tables, each valid on a particular day
- Constraint: Employees do not have gaps in their position history
- Last two rows may be replaced with a single row valid at [1996-06-01,3000-10-01)

Manipulating Temporal Tables: Semantics



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Types of Temporal Statements

- Applies to queries, modifications, views, integrity constraints
- **Current**: Applies to the current point in time (now)
  - What is Bob's current position ?
- **Time-sliced**: Applies to some point in time in the past or the future
  - What was Bob's position on January 1st, 2007?
- **Sequenced**: Applies to each point in time
  - What is Bob's position history ?
- **Non-sequenced**: Applies to all points in time, ignoring the time-varying nature of tables
  - When did Bob changed history ?

Temporal Keys

Incumbents

SSN	PCN	FromDate	ToDate
111223333	900225	1996-01-01	1996-06-01
111223333	900225	1996-04-01	1996-10-01

- **Constraint**: Employees have only one position at a point in time
- In the corresponding non-temporal table the key is (SSN,PCN)
- Candidate keys on Incumbents: (SSN,PCN,FromDate), (SSN,PCN,ToDate), and (SSN,PCN,FromDate,ToDate)
- None captures the constraint: there are overlapping periods associated with the same SSN
- What is needed: **sequenced** constraint, applied at each point in time
- All constraints specified on a snapshot table have sequenced counterparts, specified on the analogous valid-time table

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Primary Key

- **Constraint**: Employees have only one position at a point in time

```
CREATE  TRIGGER Seq_Primary_Key ON Incumbents
FOR INSERT, UPDATE AS
IF EXISTS ( SELECT * FROM Incumbents AS I1 WHERE 1 <
            ( SELECT COUNT(I2.SSN) FROM Incumbents AS I2
              WHERE I1.SSN = I2.SSN AND I1.PCN = I2.PCN
              AND I1.FromDate < I2.ToDate
              AND I2.FromDate < I1.ToDate ) )
   OR
   EXISTS ( SELECT * FROM Incumbents AS I
            WHERE I.SSN IS NULL OR I.PCN IS NULL )
BEGIN
  RAISERROR('Violation of sequenced primary key constraint',1,2)
  rollback transaction
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Handling Now

- What should the timestamp be for current data ?
- One alternative: using `NULL`
- Allows to indentify current records: `WHERE Incumbents.ToDate IS NULL`
- Disadvantages
  - users get confused with a data of `NULL`
  - in SQL any comparison with a null value returns false
    ⇒ rows with null values will be absent from the result of many queries
  - other uses of `NULL` are not available
- Another approach: set the end date to largest value in the timestamp domain, e.g., `'3000-01-01'`
- Disadvantages
  - DB states that something will be true in the far future
  - represent '**now**' and '**forever**' in the same way

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Types of Duplicates

Incumbents

	SSN	PCN	FromDate	ToDate
1	111223333	120033	1996-01-01	1996-06-01
2	111223333	120033	1996-04-01	1996-10-01
3	111223333	120033	1996-04-01	1996-10-01
4	111223333	120033	1996-10-01	1998-01-01
5	111223333	120033	1997-12-01	1998-01-01

- Two rows are **value equivalent** if the values of their nontimestamp columns are equivalent
- Two rows are **sequenced duplicates** if they are duplicates at some instant: 1+2 $\Rightarrow$ employee has two positions for the months of April and May of 1996
- Two rows are **current duplicates** if they are sequenced duplicates at the current instant: 4+5 $\Rightarrow$ in December 1997 a current duplicate will suddenly appear
- Two rows are **nonsequenced duplicates** if the values of all columns are identical: 2+3

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Preventing Duplicates (1)

- Preventing value-equivalent rows: define secondary key using `UNIQUE(SSN,PCN)`
- Preventing nonsequenced duplicates: `UNIQUE(SSN,PCN,FromDate,ToDate)`
- Preventing current duplicates: No employee can have two identical positions at the current time

```
CREATE TRIGGER Current_Duplicates ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS ( SELECT I1.SSN FROM Incumbents AS I1 WHERE 1 <
            ( SELECT COUNT(I2.SSN) FROM Incumbents AS I2
              WHERE I1.SSN = I2.SSN AND I1.PCN=I2.PCN
              AND I1.FromDate <= CURRENT_DATE
              AND CURRENT_DATE < I1.ToDate
              AND I2.FromDate <= CURRENT_DATE
              AND CURRENT_DATE < I2.ToDate ) )
BEGIN
     RAISERROR('Transaction allows current duplicates',1,2)
     rollback transaction
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Preventing Duplicates (2)

- Preventing current duplicates, assuming no future data: current data will have the same `ToDate` (`'3000-01-01'`) ⇒ `UNIQUE(SSN,PCN,ToDate)`
- Preventing sequenced duplicates: since a primary key is a combination of `UNIQUE` and `NOT NULL`, remove the `NOT NULL` portion of code for keys in the previous trigger

```
CREATE  TRIGGER Seq_Primary_Key ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS ( SELECT I1.SSN FROM Incumbents AS I1 WHERE 1 <
            ( SELECT COUNT(I2.SSN) FROM Incumbents AS I2
              WHERE I1.SSN = I2.SSN AND I1.PCN=I2.PCN
              AND I1.FromDate < I2.ToDate
              AND I2.FromDate < I1.ToDate ) )
BEGIN
  RAISERROR('Transaction allows sequenced duplicates',1,2)
  rollback transaction
END
```

- Preventing sequenced duplicates, asumming only current modifications: `UNIQUE(SSN,PCN,ToDate)`

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Uniqueness (1)

- Constraint: Each employee has at most one position
- Snapshot table: `UNIQUE(SSN)`
- Sequenced constraint: At any time each employee has at most one position, i.e., `Incumbents.SSN` is sequenced unique

```
CREATE  TRIGGER Seq_Unique ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS ( SELECT I1.SSN FROM Incumbents AS I1 WHERE 1 <
            ( SELECT COUNT(I2.SSN) FROM Incumbents AS I2
              WHERE I1.SSN = I2.SSN
              AND I1.FromDate < I2.ToDate
              AND I2.FromDate < I1.ToDate ) )
   OR
   EXISTS ( SELECT * FROM Incumbents AS I
                    WHERE I.SSN IS NULL )
BEGIN
  RAISERROR('Transaction violates sequenced unique constraint',1,2)
  rollback transaction
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Uniqueness (2)

- Nonsequenced constraint: an employee cannot have more than one position over two identical periods, i.e., Incumbents.SSN is nonsequenced unique:

```
UNIQUE(SSN,FromDate,ToDate)
```

- Current constraint: an employee has at most one position, i.e., Incumbents.SSN is current unique:

```
CREATE  TRIGGER Current_Unique ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS ( SELECT I1.SSN FROM Incumbents AS I1 WHERE 1 <
            ( SELECT COUNT(I2.SSN) FROM Incumbents AS I2
              WHERE I1.SSN = I2.SSN
              AND I1.FromDate <= CURRENT_DATE
              AND CURRENT_DATE < I1.ToDate ) )
BEGIN
  RAISERROR('Transaction violates current unique constraint',1,2)
  rollback transaction
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Referential Integrity (1)

- Incumbents.PCN is a foreign key for Position.PCN
- **Case 1**: Neither table is temporal

```
CREATE TABLE Incumbents ( ...
    PCN CHAR(6) NOT NULL REFERENCES Position,  ... )
```

- **Case 2**: Both tables are temporal

The PCN of all current incumbents must be listed in the current positions

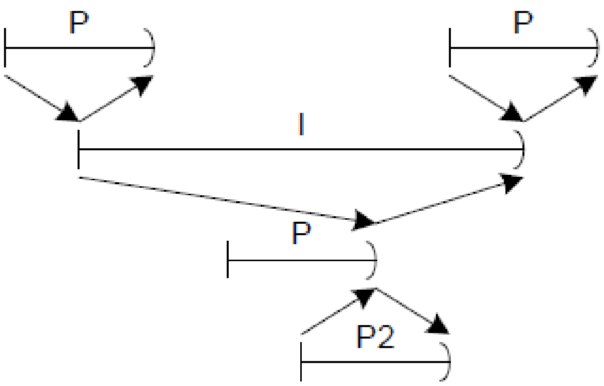
```
CREATE TRIGGER Current_Referential_Integrity ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS ( SELECT * FROM Incumbents AS I
            WHERE I.ToDate = '3000-01-01'
            AND NOT EXISTS (
              SELECT * FROM Position AS P
              WHERE I.PCN = P.PCN AND P.ToDate = '3000-01-01' ) )
BEGIN
  RAISERROR('Violation of current referential integrity',1,2)
  ROLLBACK TRANSACTION
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Referential Integrity (2)

- `Incumbents.PCN` is a sequenced foreign key for `Position.PCN`

```
CREATE TRIGGER Sequenced_Ref_Integrity ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS (
  SELECT * FROM Incumbents AS I
  WHERE NOT EXISTS (
    SELECT * FROM Position AS P
    WHERE I.PCN = P.PCN AND P.FromDate <= I.FromDate
    AND I.FromDate < P.ToDate )
  OR NOT EXISTS (
    SELECT * FROM Position AS P
    WHERE I.PCN = P.PCN AND P.FromDate < I.ToDate
    AND I.ToDate <= P.ToDate )
  OR EXISTS (
    SELECT * FROM Position AS P
    WHERE I.PCN = P.PCN AND I.FromDate < P.ToDate
    AND P.ToDate < I.ToDate AND NOT EXISTS (
      SELECT * FROM Position AS P2
      WHERE P2.PCN = P.PCN AND P2.FromDate <= P.ToDate
      AND P.ToDate < P2.ToDate ) ) )
BEGIN
  RAISERROR('Violation of sequenced referential integrity',1,2)
  ROLLBACK TRANSACTION
END
```

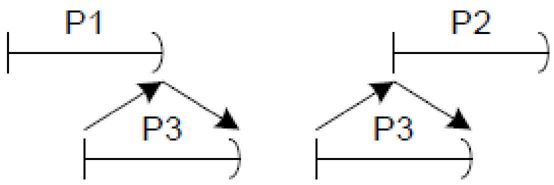


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Contiguous History

- Incumbents.PCN defines a contiguous history

```
CREATE TRIGGER Contiguous_History ON Position
FOR INSERT, UPDATE, DELETE AS
IF EXISTS (
  SELECT * FROM Position AS P1, Position AS P2
  WHERE P1.PCN = P2.PCN AND P1.ToDate < P2.FromDate
  AND NOT EXISTS (
    SELECT * FROM Position AS P3
    WHERE P3.PCN = P1.PCN
    AND ( ( P3.FromDate <= P1.ToDate
            AND P1.ToDate < P3.ToDate )
    OR ( P3.FromDate < P2.FromDate
            AND P2.FromDate <= P3.ToDate ) ) ) )
BEGIN
  RAISERROR('Transaction violates contiguous history',1,2)
  ROLLBACK TRANSACTION
END
```



- This is a **nonsequenced constraint**: it require examining the table at multiple points of time

Referential Integrity (3)

- Incumbents.PCN is a sequenced foreign key for Position.PCN, and Incumbents.PCN defines a contiguous history

```
CREATE TRIGGER Sequenced_Ref_Integrity ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS (
    SELECT * FROM Incumbents AS I
    WHERE NOT EXISTS (
      SELECT * FROM Position AS P WHERE I.PCN = P.PCN
      AND P.FromDate <= I.ToDate AND I.FromDate < P.ToDate )
    OR NOT EXISTS (
      SELECT * FROM Position AS P WHERE I.PCN = P.PCN
      AND P.FromDate < I.ToDate AND I.ToDate <= P.ToDate ) )
BEGIN
  RAISERROR('Violation of sequenced referential integrity',1,2)
  ROLLBACK TRANSACTION
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Referential Integrity (4)

- **Case 4:** Only the referenced table is temporal
- `Incumbents.PCN` is a current foreign key for `Position.PCN`

```
CREATE TRIGGER Current_Referential_Integrity ON Incumbents
FOR INSERT, UPDATE, DELETE AS
IF EXISTS (
    SELECT * FROM Incumbents AS I
    WHERE NOT EXISTS (
      SELECT * FROM Position AS P
      WHERE I.PCN = P.PCN AND P.ToDate = '3000-01-01' ) )
BEGIN
  RAISERROR('Violation of current referential integrity',1,2)
  ROLLBACK TRANSACTION
END
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying Valid-Time Tables

Employee

SSN	FirstName	LastName	BirthDate

Position

PCN	JobTitle

Incumbents

SSN	PCN	FromDate	ToDate

Salary

SSN	Amount	FromDate	ToDate

- As for constraints, queries and modifications can be of three kinds
  - **current**, **sequenced**, and **nonsequenced**
- Extracting the current state: What is Bob's current position

```
SELECT JobTitle
FROM Employee E, Incumbents I, Position P
WHERE E.FirstName = 'Bob'
AND E.SSN = I.SSN AND I.PCN = P.PCN
AND I.ToDate = '3000-01-01'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Extracting Current State (1)

- Another alternative for obtaining Bob's current position

```
SELECT JobTitle
FROM Employee E, Incumbents I, Position P
WHERE E.FirstName = 'Bob'
AND E.SSN = I.SSN AND I.PCN = P.PCN
AND I.FromDate <= CURRENT_DATE AND CURRENT_DATE < I.ToDate
```

- Current joins over two temporal tables are not too difficult

- What is Bob's current position and salary ?

```
SELECT JobTitle, Amount
FROM Employee E, Incumbents I, Position P, Salary S
WHERE FirstName = 'Bob'
AND E.SSN = I.SSN AND I.PCN = P.PCN AND E.SSN = S.SSN
AND I.FromDate <= CURRENT_DATE AND CURRENT_DATE < I.ToDate
AND S.FromDate <= CURRENT_DATE AND CURRENT_DATE < S.ToDate
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Extracting Current State (2)

- What employees currently have no position?

```
SELECT FirstName
FROM Employee E
WHERE NOT EXISTS (
  SELECT *
  FROM Incumbents I
  WHERE E.SSN = I.SSN
  AND I.FromDate <= CURRENT_DATE AND CURRENT_DATE < I. ToDate )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Extracting Prior States

- **Timeslice queries**: extracts a state at a particular point in time
- Timeslice queries over a previous state requires an additional predicate for each temporal table
- What was Bob's position at the beginning of 1997?

```
SELECT JobTitle
FROM Employee E, Incumbents I, Position P
WHERE E.FirstName = 'Bob'
AND E.SSN = I.SSN AND I.PCN = P.PCN
AND I.FromDate <= '1997-01-01' AND '1997-01-01' < I.ToDate
```

Sequenced Queries

- Queries whose result is a valid-time table
- Use sequenced variants of basic operations
  - Selection, projection, union, sorting, join, difference, and duplicate elimination
- **Sequenced selection**: no change is necessary
- Who makes or has made more than 50K annually

```
SELECT *
FROM Salary
WHERE Amount > 50000
```

- **Sequenced projection**: include the timestamp columns in the select list
- List the social security numbers of current and past employees

```
SELECT SSN, FromDate, ToDate
FROM Salary
```

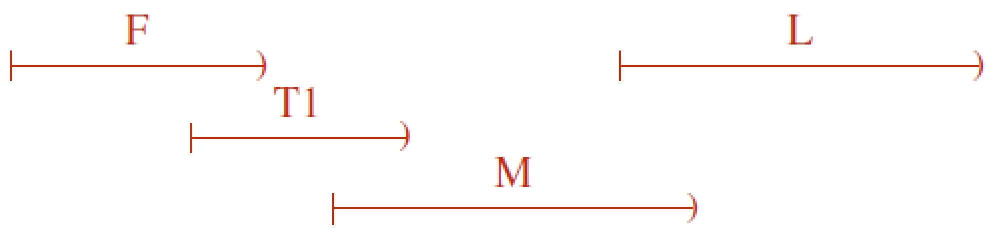
- Duplications resulting from the projection are retained
- To eliminate them **coalescing** is needed (see next)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Coalescing entirely in SQL



- Select those start and end dates such that
  - there are not gaps between these dates
  - no value-equivalent row overlaps the period between the selected start and end dates and has an earlier start date or a later end date
- Search two value-equivalent rows `F`(irst) and `L`(ast) defining the start and end points of a coalesced row
- First `NOT EXISTS` ensures that there are no gaps between `F.ToDate` and `L.FromDate`
- Second `NOT EXIST` ensures that only maximal periods result, i.e. `F` and `L` cannot be part of a larger value-equivalent row `T2`

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Coalescing while Removing Duplicates

```
SELECT DISTINCT F.SSN, F.FromDate, L.ToDate
FROM Salary F, Salary L
WHERE F.FromDate < L.ToDate
AND F.SSN = L.SSN
AND NOT EXISTS ( SELECT * FROM Salary AS M
  WHERE M.SSN = F.SSN
  AND F.FromDate < M.FromDate AND M.FromDate <= L.ToDate
  AND NOT EXISTS ( SELECT * FROM Salary AS T1
    WHERE T1.SSN = F.SSN AND
    AND T1.FromDate < M.FromDate AND M.FromDate <= T1.ToDate ) )
AND NOT EXISTS ( SELECT * FROM Salary AS T2
  WHERE T2.SSN = F.SSN AND
  AND ( (T2.FromDate < F.FromDate AND F.FromDate <= T2.ToDate)
     OR (T2.FromDate <= L.ToDate AND L.ToDate < T2.ToDate) ) )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Sort

- Requires the result to be ordered at each point in time
- This can be accomplished by appending the start and end time columns in the `ORDER BY` clause
- Sequenced sort `Incumbents` on the position code (first version)

```
SELECT *
FROM Incumbents
ORDER BY PCN, FromDate, ToDate
```

- Sequenced sorting can also be accomplished by omitting the timestamp columns

```
SELECT *
FROM Incumbents
ORDER BY PCN
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Union

- A UNION ALL (retaining duplicates) over temporal tables is automatically sequenced if the timestamp columns are kept
- Who makes or has made annually more than 50,000 or less than 10,000?

```
SELECT *
FROM Salary
WHERE Amount > 50000
UNION ALL
SELECT *
FROM Salary
WHERE Amount < 10000
```

- A UNION without ALL eliminates duplicates but is difficult to express in SQL (see later)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

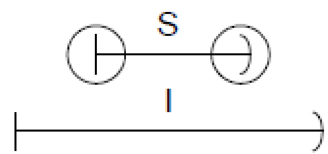
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Join (1)

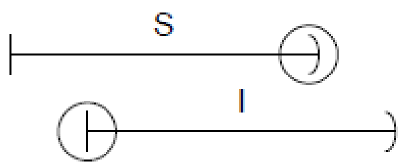
- Example: determine the salary and position history for each employee
- Implies a sequenced join between `Salary` and `Incumbents`
- It is supposed that there are no duplicate rows in the tables: at each point in time an employee has **one** salary and **one** position
- In SQL a sequenced join requires four select statements and complex inequality predicates
- The following code does not generates duplicates
- For this reason `UNION ALL` is used which is more efficient than `UNION`, which does a lot of work for remove the nonocccurring duplicates

Sequenced Join (2)

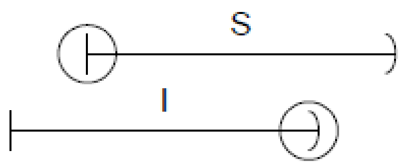
```
SELECT S.SSN, Amount, PCN, S.FromDate, S.ToDate
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND I.FromDate < S.FromDate AND S.ToDate <= I.ToDate
    UNION ALL
```



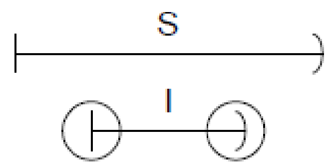

```
SELECT S.SSN, Amount, PCN, S.FromDate, I.ToDate
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND S.FromDate >= I.FromDate
AND S.FromDate < I.ToDate AND I.ToDate < S.ToDate
    UNION ALL
```




```
SELECT S.SSN, Amount, PCN, I.FromDate, S.ToDate
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND I.FromDate >= S.FromDate
AND I.FromDate < S.ToDate AND S.ToDate < I.ToDate
    UNION ALL
```




```
SELECT S.SSN, Amount, PCN, I.FromDate, I.ToDate
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND I.FromDate > S.FromDate AND I.ToDate < S.ToDate
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Join using CASE

```
SELECT S.SSN, Amount, PCN, "Start Date"=
    CASE WHEN S.FromDate > I.FromDate
         THEN S.FromDate ELSE I.FromDate
    END, "End Date"=
    CASE WHEN S.ToDate > I.ToDate
         THEN I.ToDate ELSE S.ToDate
    END
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND (CASE WHEN S.FromDate > I.FromDate
          THEN S.FromDate ELSE I.FromDate
     END)
<   (CASE WHEN S.ToDate > I.ToDate
          THEN I.ToDate ELSE S.ToDate
     END)
```

- CASE allows to write this query in a single statement
- First CASE simulates a maxDate function of the two arguments, the second one a minDate function
- Condition in the WHERE ensures that the period of validity is well formed

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Join using Functions: SQL Server Example

```
create function minDate(@one smalldatetime, @two smalldatetime)
returns smalldatetime as
begin
  return CASE WHEN @one < @two then @one else @two end
end

create function maxDate(@one smalldatetime, @two smalldatetime)
returns smalldatetime as
begin
  return CASE WHEN @one > @two then @one else @two end
end

SELECT S.SSN, Amount, PCN,
    "Start Date" = maxDate(S.FromDate,I.FromDate),
    "End Date" = minDate(S.ToDate,I.ToDate)
FROM Salary S, Incumbents I
WHERE S.SSN = I.SSN
AND maxDate(S.FromDate,I.FromDate) < minDate(S.ToDate,I.ToDate)
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Difference

- Implemented in SQL with EXCEPT, NOT EXISTS, or NOT IN
- List the employees who are department heads (PCN=1234) but are not also professors (PCN=5555)
- Nontemporal version

```
SELECT SSN
FROM Incumbents I1
WHERE I1.PCN = 1234
AND NOT EXISTS ( SELECT * FROM Incumbents I2
  WHERE I1.SSN = I2.SSN AND I2.PCN = 5555 )
```

- Using EXCEPT (not available in SQL Server)

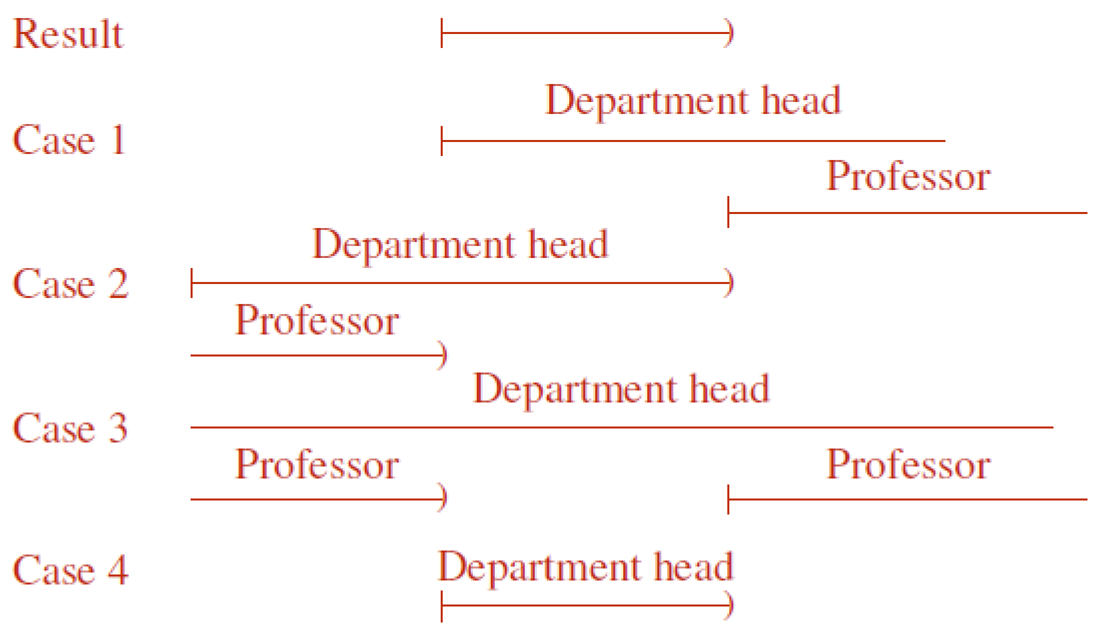
```
SELECT SSN
FROM Incumbents
WHERE PCN = 1234
EXCEPT
SELECT SSN
FROM Incumbents
WHERE PCN = 5555
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Difference (1)

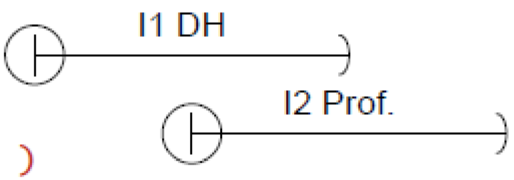


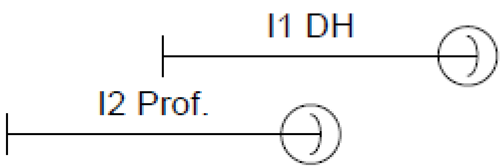
- Sequenced version: Identify when the department heads were not professors
- Four possible cases should be taken into account
- Each of them requires a separate SELECT statement

Sequenced Difference (2)

- List the employees who are or were department heads (PCN=1234) but not also professors (PCN=5555)

```
SELECT I1.SSN, I1.FromDate, I2.FromDate AS ToDate
FROM Incumbents I1, Incumbents I2
WHERE I1.PCN = 1234 AND I2.PCN = 5555 AND I1.SSN = I2.SSN
AND I1.FromDate < I2.FromDate AND I2.FromDate < I1.ToDate
AND NOT EXISTS ( SELECT * FROM Incumbents I3
  WHERE I1.SSN = I3.SSN AND I3.PCN = 5555
  AND I1.FromDate < I3.ToDate AND I3.FromDate < I2.FromDate )
UNION
SELECT I1.SSN, I2.ToDate AS FromDate, I1.ToDate
FROM Incumbents I1, Incumbents I2
WHERE I1.PCN = 1234 AND I2.PCN = 5555 AND I1.SSN = I2.SSN
AND I1.FromDate < I2.ToDate AND I2.ToDate < I1.ToDate
AND NOT EXISTS ( SELECT * FROM Incumbents I3
  WHERE I1.SSN = I3.SSN AND I3.PCN = 5555
  AND I2.ToDate < I3.ToDate AND I3.FromDate < I1.ToDate )
UNION
...
```

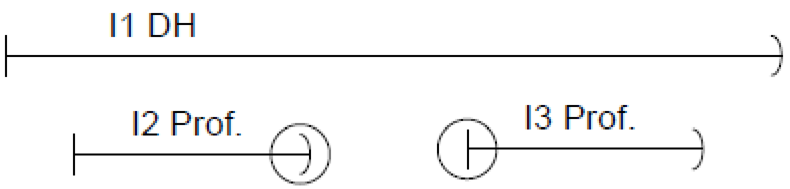


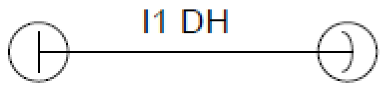


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Difference (3)

```
...
SELECT I1.SSN, I2.ToDate AS FromDate, I3.FromDate AS ToDate
FROM Incumbents I1, Incumbents I2, Incumbents I3
WHERE I1.PCN = 1234 AND I2.PCN = 5555 AND I3.PCN = 5555
AND I1.SSN = I2.SSN AND I1.SSN = I3.SSN
AND I2.ToDate < I3.FromDate
AND I1.FromDate < I2.ToDate
AND I3.FromDate < I1.ToDate
AND NOT EXISTS ( SELECT * FROM Incumbents I4
  WHERE I1.SSN = I4.SSN AND I4.PCN = 5555
  AND I2.ToDate < I4.ToDate AND I4.FromDate < I3.FromDate )
UNION
SELECT SSN, FromDate, ToDate
FROM Incumbents I1
WHERE I1.PCN = 1234
AND NOT EXISTS ( SELECT * FROM Incumbents I4
  WHERE I1.SSN=I4.SSN AND I4.PCN = 5555
  AND I1.FromDate < I4.ToDate AND I4.FromDate < I1.ToDate )
```





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Nonsequenced Variants

- Nonsequenced operators (selection, join , . . .) are straightforward
  - They ignore the time-varying nature of tables
- List all the salaries, past and present, of employees who had been lecturer at some time

```
SELECT Amount
FROM Incumbents I, Position P, Salary S
WHERE I.SSN = S.SSN AND I.PCN = P.PCN
AND JobTitle = 'Lecturer'
```

- When did employees receive raises?

```
SELECT S2.SSN, S2.FromDate AS RAISE_DATE
FROM Salary S1, Salary S2
WHERE S2.Amount > S1.Amount
AND S1.SSN = S2.SSN
AND S1.ToDate = S2.FromDate
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Eliminating Duplicates

- Remove nonsequenced duplicates from Incumbents

```
SELECT DISTINCT *
FROM Incumbents
```

- Remove value-equivalent rows from Incumbents

```
SELECT DISTINCT SSN,PCN
FROM Incumbents
```

- Remove current duplicates from Incumbents

```
SELECT DISTINCT SSN,PCN
FROM Incumbents
WHERE ToDate = '3000-01-01'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Aggregation Functions

Affiliation

SSN	DNumber	FromDate	ToDate

Salary

SSN	Amount	FromDate	ToDate

- SQL provides aggregation functions: COUNT, MIN, MAX, AVG, . . .
- List the maximum salary: non-temporal version

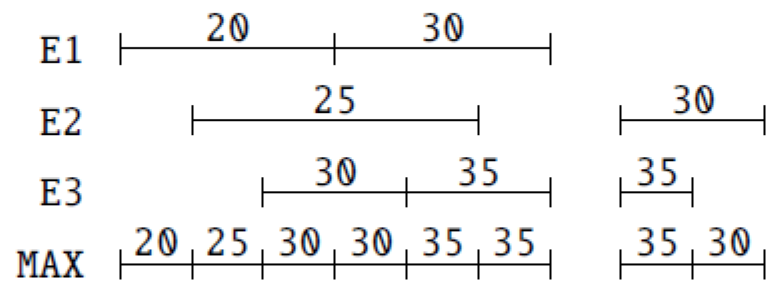
```
SELECT MAX(Amount)
FROM Salary
```

- List by department the maximum salary: non-temporal version

```
SELECT DNumber, MAX(Amount)
FROM Affiliation A, Salary S
WHERE A.SSN = S.SSN
GROUP BY DNumber
```


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Maximum Salary: Temporal Version (1)



- **First step**: Compute the periods on which a maximum must be calculated

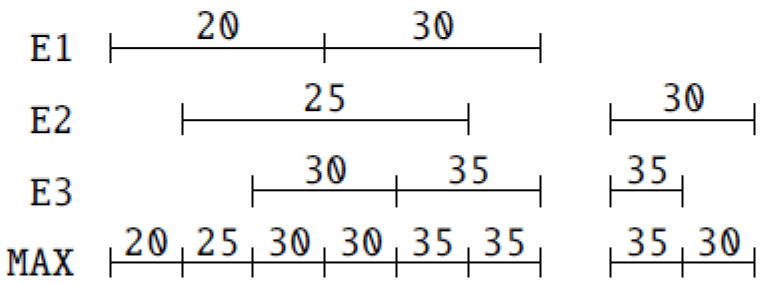
```
CREATE VIEW SalChanges(Day) as
  SELECT DISTINCT FromDate FROM Salary
  UNION
  SELECT DISTINCT ToDate FROM Salary
CREATE VIEW SalPeriods(FromDate, ToDate) as
  SELECT P1.Day, P2.Day
  FROM SalChanges P1, SalChanges P2
  WHERE P1.Day < P2.Day
  AND NOT EXISTS ( SELECT * FROM SalChanges P3
    WHERE P1.Day < P3.Day AND P3.Day < P2.Day )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Maximum Salary: Temporal Version (2)

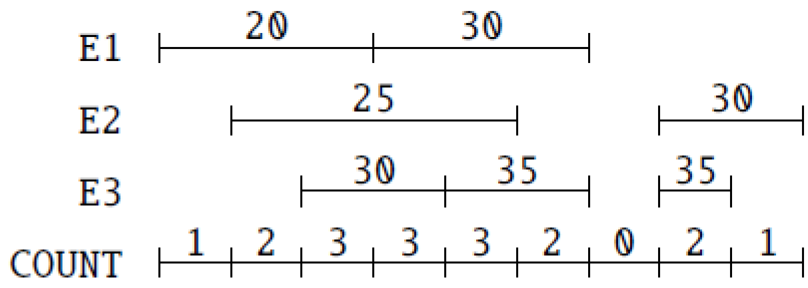


- **Second step**: Compute the maximum salary for these periods

```
CREATE VIEW TempMax(MaxSalary, FromDate, ToDate) as
    SELECT MAX(E.Amount), I.FromDate, I.ToDate
    FROM Salary E, SalPeriods I
    WHERE E.FromDate <= I.FromDate AND I.ToDate <= E.ToDate
    GROUP BY I.FromDate, I.ToDate
```

- **Third step**: Coalesce the above view (as seen before)

Number of Employees: Temporal Version

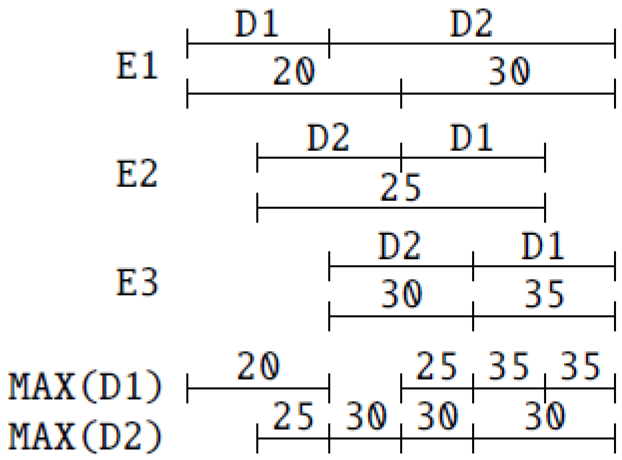


- **Second step**: Compute the number of employees for these periods

```
CREATE VIEW TempCount(NbEmp, FromDate, ToDate) as
    SELECT COUNT(*), P.FromDate, P.ToDate
    FROM Salary S, SalPeriods P
    WHERE S.FromDate<=P.FromDate AND P.ToDate<=S.ToDate
    GROUP BY P.FromDate, P.ToDate
UNION ALL
    SELECT 0, P.FromDate, P.ToDate
    FROM SalPeriods P
    WHERE NOT EXISTS (
        SELECT * FROM Salary S
        WHERE S.FromDate<=P.FromDate AND P.ToDate<=S.ToDate )
```

- **Third step**: Coalesce the above view (as seen before)

Maximum Salary by Department: Temporal Version (1)



- Hypothesis: Employees have salary only while they are affiliated to a department

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Maximum Salary by Department: Temporal Version (2)

- **First step**: Compute by department the periods on which a maximum must be calculated

```
CREATE VIEW Aff_Sal(DNumber, Amount, FromDate, ToDate) as
  SELECT DISTINCT A.DNumber, S.Amount,
    maxDate(S.FromDate,A.FromDate), minDate(S.ToDate,A.ToDate)
  FROM Affiliation A, Salary S
  WHERE A.SSN=S.SSN
  AND maxDate(S.FromDate,A.FromDate) < minDate(S.ToDate,A.ToDate)
CREATE VIEW SalChanges(DNumber, Day) as
  SELECT DISTINCT DNumber, FromDate FROM Aff_Sal
  UNION
  SELECT DISTINCT DNumber, ToDate FROM Aff_Sal
CREATE VIEW SalPeriods(DNumber, FromDate, ToDate) as
  SELECT P1.DNumber, P1.Day, P2.Day
  FROM SalChanges P1, SalChanges P2
  WHERE P1.DNumber = P2.DNumber AND P1.Day < P2.Day
  AND NOT EXISTS ( SELECT * FROM SalChanges P3
    WHERE P1.DNumber = P3.DNumber AND P1.Day < P3.Day
    AND P3.Day < P2.Day )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Maximum Salary by Department: Temporal Version (3)

- **Second step**: Compute the maximum salary for these periods

```
CREATE VIEW TempMaxDep(DNumber, MaxSalary,
    FromDate, ToDate) as
  SELECT P.DNumber, MAX(Amount), P.FromDate, P.ToDate
  FROM Aff_Sal A, SalPeriods P
  WHERE A.DNumber = P.DNumber
  AND A.FromDate <= P.FromDate AND P.ToDate <= A.ToDate
  GROUP BY P.DNumber, P.FromDate, P.ToDate
```

- **Third step**: Coalesce the above view (as seen before)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division

Affiliation

SSN	DNumber	FromDate	ToDate

Controls

PNumber	DNumber	FromDate	ToDate

WorksOn

SSN	PNumber	FromDate	ToDate

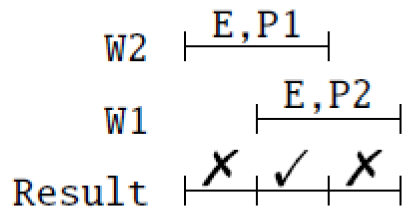
- Implemented in SQL with two nested `NOT EXISTS`
- List the employees that work in all projects of the department to which they are affiliated: non-temporal version

```
SELECT SSN
FROM Affiliation A
WHERE NOT EXISTS ( SELECT * FROM Controls C
  WHERE A.DNumber = C.DNumber
  AND NOT EXISTS ( SELECT * FROM WorksOn W
     WHERE C.PNumber = W.PNumber
     AND A.SSN = W.SSN ) )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 1

- Only WorksOn is temporal
- **First step**: Compute the division




```
Affiliation(E,D)
Controls(D,P1)
Controls(D,P2)
```

```
CREATE VIEW TempUnivQuant(SSN, FromDate, ToDate) as
  SELECT DISTINCT W1.SSN, W1.FromDate, W2.ToDate
  FROM WorksOn W1, WorksOn W2, Affiliation A
  WHERE W1.SSN = W2.SSN AND W1.SSN = A.SSN
  AND W1.FromDate < W2.ToDate
  AND NOT EXISTS ( SELECT * FROM Controls C
    WHERE A.DNumber = C.DNumber
    AND NOT EXISTS ( SELECT * FROM WorksOn W
      WHERE C.PNumber = W.PNumber AND A.SSN = W.SSN
      AND W.FromDate <= W1.FromDate
      AND W2.ToDate <= W.ToDate ) )
```

- **Second step**: Coalesce the above view

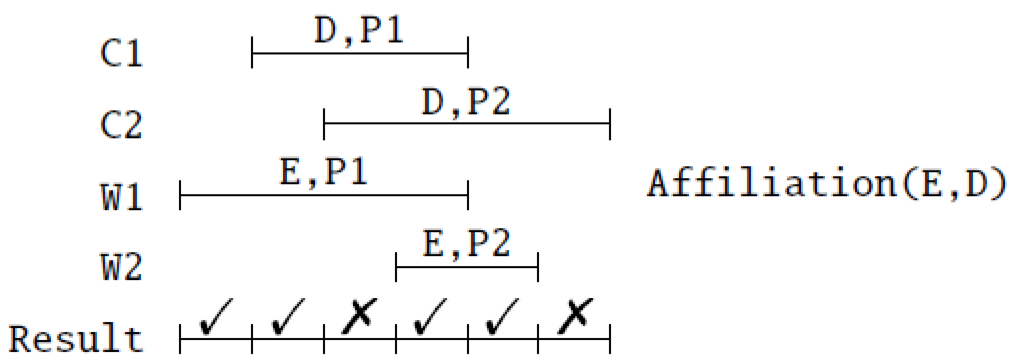
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 2 (1)

- Only Controls and WorksOn are temporal
- Employees may work in projects controlled by departments different from the department to which they are affiliated
- **First step**: Construct the periods on which the division must be computed



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 2 (2)

```
CREATE VIEW ProjChangesC2(SSN,Day) as
  SELECT DISTINCT SSN,FromDate
    FROM Affiliation A, Controls C
    WHERE A.DNumber=C.DNumber UNION
  SELECT DISTINCT SSN,ToDate
    FROM Affiliation A, Controls C
    WHERE A.DNumber=C.DNumber UNION
  SELECT DISTINCT SSN,FromDate FROM WorksOn UNION
  SELECT SSN,ToDate FROM WorksOn
CREATE VIEW ProjPeriodsC2(SSN,FromDate,ToDate) as
  SELECT P1.SSN,P1.Day,P2.Day
  FROM ProjChangesC2 P1, ProjChangesC2 P2
  WHERE P1.SSN=P2.SSN AND P1.Day<P2.Day
  AND NOT EXISTS ( SELECT * FROM ProjChangesC2 P3
    WHERE P1.SSN=P3.SSN AND P1.Day<P3.Day
    AND P3.Day<P2.Day )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 2 (3)

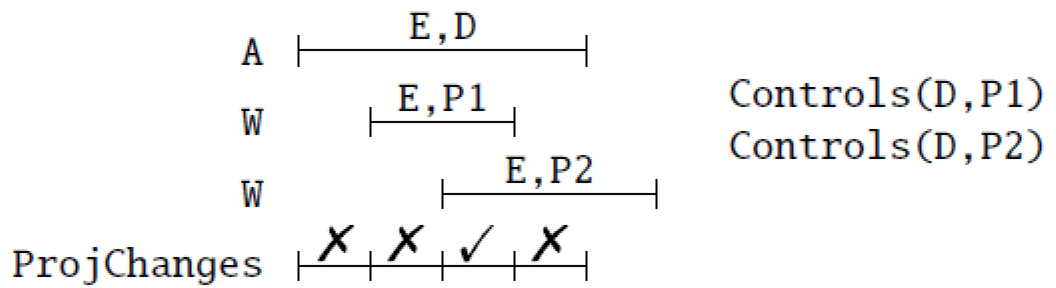
- **Second step**: Compute the division of these periods

```
CREATE VIEW TempUnivC2(SSN,FromDate,ToDate) as
  SELECT DISTINCT P.SSN,P.FromDate,P.ToDate
  FROM ProjPeriodsC2 P, Affiliation A
  WHERE P.SSN=A.SSN AND NOT EXISTS (
    SELECT * FROM Controls C WHERE A.DNumber=C.DNumber
    AND C.FromDate<=P.FromDate AND P.ToDate<=C.ToDate
    AND NOT EXISTS ( SELECT * FROM WorksOn W
      WHERE C.PNumber=W.PNumber AND P.SSN=W.SSN
      AND W.FromDate<=P.FromDate
      AND P.ToDate<=W.ToDate ) )
```

- **Third step**: Coalesce the above view

Sequenced Division: Case 3 (1)

- Only `Affiliation` and `WorksOn` are temporal
- Employees may work in projects controlled by departments different from the department to which they are affiliated
- **First step**: Construct the periods on which the division must be computed



Sequenced Division: Case 3 (2)

```
CREATE VIEW Aff_WO(SSN, DNumber, PNumber, FromDate, ToDate) as
  SELECT DISTINCT A.SSN, A.DNumber, W.PNumber,
    maxDate(A.FromDate,W.FromDate), minDate(A.ToDate,W.ToDate)
  FROM Affiliation A, WorksOn W
  WHERE A.SSN=W.SSN
  AND maxDate(A.FromDate,W.FromDate) < minDate(A.ToDate,W.ToDate)
CREATE VIEW ProjChangesC3(SSN, DNumber, Day) as
  SELECT DISTINCT SSN, DNumber, FromDate FROM Aff_WO UNION
  SELECT DISTINCT SSN, DNumber, ToDate FROM Aff_WO UNION
  SELECT SSN, DNumber, FromDate FROM Affiliation UNION
  SELECT SSN, DNumber, ToDate FROM Affiliation
CREATE VIEW ProjPeriodsC3(SSN, DNumber, FromDate, ToDate) as
  SELECT P1.SSN, P1.DNumber, P1.Day, P2.Day
  FROM ProjChangesC3 P1, ProjChangesC3 P2
  WHERE P1.SSN = P2.SSN
  AND P1.DNumber = P2.DNumber AND P1.Day < P2.Day
  AND NOT EXISTS ( SELECT * FROM ProjChangesC3 P3
    WHERE P1.SSN = P3.SSN AND P1.DNumber = P3.DNumber
    AND P1.Day < P3.Day AND P3.Day < P2.Day )
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 3 (3)

- **Second step**: Compute the division of these periods

```
CREATE VIEW TempUnivQuant(SSN, FromDate, ToDate) as
  SELECT DISTINCT P.SSN, P.FromDate, P.ToDate
  FROM ProjPeriodsC3 P
  WHERE NOT EXISTS ( SELECT * FROM Controls C
    WHERE P.DNumber=C.DNumber
    AND NOT EXISTS ( SELECT * FROM WorksOn W
      WHERE C.PNumber=W.PNumber AND P.SSN=W.SSN
      AND W.FromDate<=P.FromDate
      AND P.ToDate<=W.ToDate ) )
```

- **Third step**: Coalesce the above view

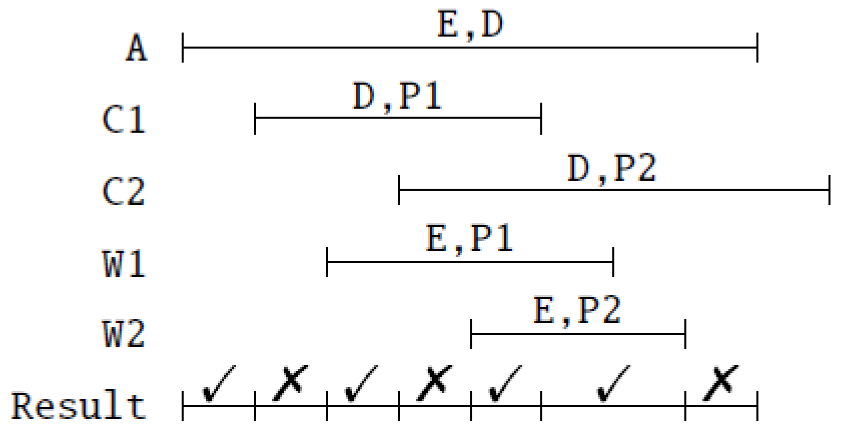
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 4 (1)

- `Affiliation`, `Controls`, and `WorksOn` are all temporal
- **First step**: Construct the periods on which the division must be computed



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sequenced Division: Case 4 (2)

```
CREATE VIEW Aff_Cont(SSN, DNumber, PNumber, FromDate, ToDate) as
  SELECT DISTINCT A.SSN, A.DNumber, C.PNumber,
    maxDate(A.FromDate,C.FromDate), minDate(A.ToDate,C.ToDate)
  FROM Affiliation A, Controls C WHERE A.DNumber=C.DNumber
  AND maxDate(A.FromDate,C.FromDate) < minDate(A.ToDate,C.ToDate)
CREATE VIEW Aff_Cont_WO(SSN, DNumber, PNumber, FromDate, ToDate) as
  SELECT DISTINCT A.SSN, A.DNumber, W.PNumber,
    maxDate(A.FromDate,W.FromDate), minDate(A.ToDate,W.ToDate)
  FROM Aff_Cont A, WorksOn W WHERE A.PNumber=W.PNumber AND A.SSN=W.SSN
  AND maxDate(A.FromDate,W.FromDate) < minDate(A.ToDate,W.ToDate)
CREATE VIEW ProjChangesC4(SSN, DNumber, Day) as
  SELECT DISTINCT SSN, DNumber, FromDate FROM Aff_Cont UNION
  SELECT DISTINCT SSN, DNumber, ToDate FROM Aff_Cont UNION
  SELECT DISTINCT SSN, DNumber, FromDate FROM Aff_Cont_WO UNION
  SELECT DISTINCT SSN, DNumber, ToDate FROM Aff_Cont_WO UNION
  SELECT SSN, DNumber, FromDate FROM Affiliation UNION
  SELECT SSN, DNumber, ToDate FROM Affiliation
CREATE VIEW ProjPeriodsC4(SSN, DNumber, FromDate, ToDate) as
  SELECT P1.SSN, P1.DNumber, P1.Day, P2.Day
  FROM ProjChangesC4 P1, ProjChangesC4 P2 WHERE P1.SSN = P2.SSN
  AND P1.DNumber = P2.DNumber AND P1.Day < P2.Day
  AND NOT EXISTS ( SELECT * FROM ProjChangesC4 P3
    WHERE P1.SSN = P3.SSN AND P1.DNumber = P3.DNumber
    AND P1.Day < P3.Day AND P3.Day < P2.Day )
```

Sequenced Division: Case 4 (3)

- **Second step**: Compute the division of these periods

```
CREATE VIEW TempUnivQuant(SSN, FromDate, ToDate) as
  SELECT DISTINCT P.SSN, P.FromDate, P.ToDate
  FROM ProjPeriodsC4 P
  WHERE NOT EXISTS ( SELECT * FROM Controls C
    WHERE P.DNumber = C.DNumber
    AND C.FromDate <= P.FromDate
    AND P.ToDate <= C.ToDate
    AND NOT EXISTS ( SELECT * FROM WorksOn W
      WHERE C.PNumber = W.PNumber AND P.SSN=W.SSN
      AND W.FromDate <= P.FromDate
      AND P.ToDate <= W.ToDate ) )
```

- **Third step**: Coalesce the above result

Temporal Support in Oracle

- Oracle 9i, released in 2001, included support for transaction time
- Its flashback queries allow the application to access prior transaction-time states of their database; they are transaction timeslice queries
- Database modifications and conventional queries are temporally upward compatible
- Oracle 10g, released in 2006, extended flashback queries to retrieve all the versions of a row between two transaction times (a key-transaction-time-range query)
- It also allowed tables and databases to be rolled back to a previous transaction time, discarding all changes after that time
- Oracle 10g Workspace Manager includes the period data type, valid-time support, transaction-time support, bitemporal support, and support for sequenced primary keys, sequenced uniqueness, sequenced referential integrity, and sequenced selection and projection
- These facilities permit tracing of actions on data as well as the ability to perform database forensics
- Oracle 11g, released in 2007, does not rely on transient storage like the undo segments, it records changes in the Flashback Recovery Area
- Validtime queries were also enhanced

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Topics

- Introduction
- Time Ontology
- Temporal Conceptual Modeling
- Manipulating Temporal Databases with SQL-92
- **Temporal Support in SQL 2011**
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Facilities in the SQL 2011

- ISQL:2011 Part 2: SQL/Foundation, published on December 2011 (1434 pages!) has temporal support
- **Application-time period tables** (essentially valid-time tables)
  - Have sequenced primary and foreign keys
  - Support single-table valid-time sequenced insertions, deletions, and updates
  - Nonsequenced valid-time queries are supported
- **System-versioned tables** (essentially transaction-time tables)
  - Have transaction-time current primary and foreign keys
  - Support transaction-time current insertions, deletions, and updates
  - Support transaction-time current and nonsequenced queries
- **System-versioned application-time period tables** (essentially bitemporal tables)
  - Support temporal queries and modifications of combinations of the valid-time and transaction-time variants

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Support in the SQL Standard: A Short History

- First work started in July 1993 under the TSQL2 initiative led by Richard Snodgrass
- Definitive version of the TSQL2 Language Specification published in September 1994
- Book "The TSQL2 Temporal Query Language", edited by Richard Snodgrass and published by Kluwer Academic Publishers appeared in 1995
- Then work to transfer some of the constructs and insights of TSQL2 into SQL3 started
- A new part to SQL3, termed SQL/Temporal, was accepted in January, 1995 as Part 7 of the SQL3 specification
- Discussions then commenced on adding valid-time and transaction-time support to SQL/Temporal. Two change proposals, ANSI-96-501 and ANSI-96-502, were unanimously accepted by ANSI and forwarded to ISO in early 1997
- Due to disagreements within the ISO committee, the project responsible for temporal support was canceled in 2001
- Concepts and constructs from SQL/Temporal were subsequently included in SQL:2011 and have been implemented in IBM DB2, Oracle, Teradata Database, and PolarLake
- Other products have included temporal support

Brief Description of the SQL Standard (1)

- ISO/IEC 9075, Database Language SQL is the dominant database language de-jure standard
- First published in 1987, revised versions published in 1989, 1992, 1999, 2003, 2008, and 2011
- Multi-part standard with 9 Parts
  - Part 1 - Framework (SQL/Framework)
  - Part 2 - Foundation (SQL/Foundation)
  - Part 3 - Call-Level Interface (SQL/CLI)
  - Part 4 - Persistent Stored Modules (SQL/PSM)
  - Part 9 - Management of External Data (SQL/MED)
  - Part 10 - Object Language Bindings (SQL/OLB)
  - Part 11 - Information and Definition Schemas (SQL/Schemata)
  - Part 13 - SQL Routines and Types using the Java Programming Language (SQL/JRT
  - Part 14 - XML-Related Specifications (SQL/XML)
- Parts 3, 9, 10, and 13 are currently inactive

Brief Description of the SQL Standard (2)

- Part 2 - SQL/Foundation: Largest and the most important part SQL
  - General-purpose programming constructs: Data types, expressions, predicates, etc.
  - Data definition: `CREATE`/`ALTER`/`DROP` of tables, views, constraints, triggers, stored procedures, stored functions, etc.
  - Query constructs: `SELECT`, joins, etc.
  - Data manipulation: `INSERT`, `UPDATE`, `MERGE`, `DELETE`, etc.
  - Access control: `GRANT`, `REVOKE`, etc.
  - Transaction control: `COMMIT`, `ROLLBACK`, etc.
  - Connection management: `CONNECT`, `DISCONNECT`, etc.
  - Session management: `SET SESSION` statement
  - Exception handling: `GET DIAGNOSTICS` statement

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Brief Description of the SQL Standard (3)

- For conformance purpose, SQL is divided into a list of "features", grouped under two categories:
  - Mandatory features
  - Optional features
- To claim conformance, an implementation must conform to all mandatory features
- An implementation may conform to any number of optional features
- Both are listed in Annex F of each part of the SQL standard
- SQL/Foundation:2008 specifies 164 mandatory features and 280 optional features
- SQL/Foundation:2011 added a total 34 new features, including
  - System-versioned tables
  - Application-time period tables

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Application-Time Period Tables

- Contain a PERIOD clause (newly-introduced) with an user-defined period name
- Currently restricted to temporal periods only; may be relaxed in the future
- Must contain two additional columns, to store the start time and the end time of a period associated with the row
- Values of both start and end columns are set by the users
- Users can specify primary key/unique constraints to ensure that no two rows with the same key value have overlapping periods
- Users can specify referential constraints to ensure that the period of every child row is completely contained in the period of exactly one parent row or in the combined period of two or more consecutive parent rows
- Queries, inserts, updates and deletes on application-time period tables behave exactly like queries, inserts, updates and deletes on regular tables
- Additional syntax is provided on UPDATE and DELETE statements for partial period updates and deletes

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Creating an Application-Time Period Table

```
CREATE TABLE employees
(emp_name VARCHAR(50) NOT NULL PRIMARY KEY,
dept_id VARCHAR(10),
start_date DATE NOT NULL,
end_date DATE NOT NULL,
PERIOD FOR emp_period (start_date, end_date),
PRIMARY KEY (emp_name, emp_period WITHOUT OVERLAPS),
FOREIGN KEY (dept_id, PERIOD emp_period) REFERENCES
    departments (dept_id, PERIOD dept_period));
```

- `PERIOD` clause automatically enforces the constraint `end_date` > `start_date`
- The name of the period can be any user-defined name
- The period starts on the `start_date` value and ends on the value just prior to `end_date` value
- This corresponds to the `[closed, open)` model of periods

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Inserting Rows into an Application-Time Period Table (1)

- On an insertion, user provides the start and end time of the period for each row
- User-supplied time values can be either in the past, current, or in the future
- Example

```
INSERT INTO employees (emp_name, dept_id, start_date, end_date)
VALUES ('John', 'J13', DATE '1995-11-15', DATE '1996-11-15'),
       ('Tracy','K25', DATE '1996-01-01', DATE '1997-11-15')
```

emp_name	dept_id	start_date	end_date
John	J13	1995-11-15	1996-11-15
Tracy	K25	1996-01-01	1997-11-15

- Periods are (closed, open)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Inserting Rows into an Application-Time Period Table (2)

emp_name	dept_id	start_date	end_date
John	J13	1995-11-15	1996-11-15
Tracy	K25	1996-01-01	1997-11-15

- Given the above table, the following INSERT will succeed

```
INSERT INTO employees (emp_name, dept_id, start_date, end_date)
VALUES ('John', 'J13', DATE '1996-11-15', DATE '1997-11-15'),
       ('John','J12', DATE '1997-11-15', DATE '1998-11-15')
```

- The following INSERT will not, because of the inclusion of emp_period WITHOUT OVERLAPS in the primary key definition

```
INSERT INTO employees (emp_name, dept_id, start_date, end_date)
VALUES ('John', 'J13', DATE '1996-01-01', DATE '1996-12-31')
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Updating Rows in an Application-Time Period Table (1)

- All rows can be potentially updated
- Users are allowed to update the start and end columns of the period associated with each row
- When a row from an application-time period table is updated using the regular `UPDATE` statements, the regular semantics apply
- Additional syntax is provided for `UPDATE` statements to specify the time period during which the update applies
- Only those rows that lie within the specified period are impacted
- May lead to row splits, i.e., update of a row may cause insertion of up to two rows to preserve the information for the periods that lie outside the specified period
- Users are not allowed to update the start and end columns of the period associated with each row under this option

Updating Rows in an Application-Time Period Table (2)

emp_name	dept_id	start_date	end_date
John	J13	15/11/1995	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Given the above table, the following UPDATE

```
UPDATE employees
SET dept_id = 'J15'
WHERE emp_name = 'John'
```

will lead the following table

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- No changes to the period values

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Updating Rows in an Application-Time Period Table (3)

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Given the above table, the following UPDATE

```
UPDATE employees FOR PORTION OF emp_period FROM
DATE '1996-03-01' TO DATE '1996-07-01'
SET dept_id = 'M12'
WHERE emp_name = 'John'
```

will lead the following table

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	01/03/1996
John	M12	01/03/1996	01/07/1996
John	J15	01/07/1996	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Automatic row splitting: 1 update and 2 inserts

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Deleting Rows from an Application-Time Period Table (1)

- All rows can be potentially deleted
- When a row from an application-time period table is deleted using the regular DELETE statements, the regular semantics apply
- Additional syntax is provided for DELETE statements to specify the time period during which the delete applies
- Only those rows that lie within the specified period are impacted
- May lead to row splits, i.e., delete of a row may cause insertion of up to two rows to preserve the information for the periods that lie outside the specified period

Deleting Rows from an Application-Time Period Table (1)

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	01/03/1996
John	M12	01/03/1996	01/07/1996
John	J15	01/07/1996	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Given the above table, the following DELETE

```
DELETE FROM employees FOR PORTION OF emp_period FROM
DATE '1996-08-01' TO DATE '1996-09-01'
WHERE emp_name = 'John'
```

will lead the following table

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	01/03/1996
John	M12	01/03/1996	01/07/1996
John	J15	01/07/1996	01/08/1996
John	J15	01/09/1996	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Automatic row splitting: 1 delete and 2 inserts

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Deleting Rows from an Application-Time Period Table (3)

emp_name	dept_id	start_date	end_date
John	J15	15/11/1995	01/03/1996
John	M12	01/03/1996	01/07/1996
John	J15	01/07/1996	01/08/1996
John	J15	01/09/1996	15/11/1996
Tracy	K25	01/01/1996	15/11/1997

- Given the above table, the following DELETE

```
DELETE FROM employees
WHERE emp_name = 'John'
```

will lead the following table

emp_name	dept_id	start_date	end_date
Tracy	K25	01/01/1996	15/11/1997

- All rows pertaining to John are deleted

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying an Application-Time Period Table (1)

emp_name	dept_id	start_date	end_date
John	J13	15/11/1995	31/01/1998
John	M24	31/01/1998	31/12/9999
Tracy	K25	01/01/1996	31/03/2000

- Existing syntax for querying regular tables is applicable to application-time period tables also
- Which department was John in on Dec. 1, 1997?

```
SELECT dept_id
FROM employees
WHERE emp_name = 'John' AND start_date <= DATE '1997-12-01'
AND end_date > DATE '1997-12-01'
```

- Answer: J13

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying an Application-Time Period Table (2)

emp_name	dept_id	start_date	end_date
John	J13	15/11/1995	31/01/1998
John	M24	31/01/1998	31/12/9999
Tracy	K25	01/01/1996	31/03/2000

- Which department is John in currently?

```
SELECT dept_id
FROM employees
WHERE emp_name = 'John' AND start_date <= CURRENT_DATE
AND end_date > CURRENT_DATE;
```

- Answer: M24

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying an Application-Time Period Table (3)

emp_name	dept_id	start_date	end_date
John	J13	15/11/1995	31/01/1998
John	M24	31/01/1998	31/12/9999
Tracy	K25	01/01/1996	31/03/2000

- How many departments has John worked in since Jan. 1, 1996?

```
SELECT count(distinct dept_id)
FROM employees WHERE emp_name = 'John' AND start_date <= CURRENT_DATE
AND end_date > DATE '1996-01-01';
```

- Answer: 2

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Benefits of Application-Time Period Tables

- Most business data is time sensitive, i.e., need to track the time period during when a data item is deemed valid or effective from the business point of view
- Database systems today offer no support for
  - Associating user-maintained time periods with rows
  - Enforcing constraints such as "an employee can be in only one department in any given period"
- Updating/deleting a row for a part of its validity period
- Currently, applications take on the responsibility for managing such requirements
- Major issues
  - Complexity of code
  - Poor performance
- Use of application-time period tables provides
  - Significant simplification of application code
  - Significant improvement in performance
  - Transparent to legacy applications

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

System-Versioned Tables

- System-versioned tables are tables that contain a PERIOD clause with a pre-defined period name (SYSTEM_TIME) and specify WITH SYSTEM VERSIONING
- System-versioned tables must contain two additional columns, to store the start time and the end time of the SYSTEM_TIME period
- Values of both start and end columns are set by the system, users are not allowed to supply values for these columns
- Unlike regular tables, system-versioned tables preserve the old versions of rows as the table is updated
- Rows whose periods intersect the current time are called **current system rows**, all others are called **historical system rows**
- Only current system rows can be updated or deleted
- All constraints are enforced on current system rows only

Creating a System-Versioned Table

```
CREATE TABLE employees
(emp_name VARCHAR(50) NOT NULL, dept_id VARCHAR(10),
system_start TIMESTAMP(6) GENERATED ALWAYS AS ROW START,
system_end TIMESTAMP(6) GENERATED ALWAYS AS ROW END,
PERIOD FOR SYSTEM_TIME (system_start, system_end),
PRIMARY KEY (emp_name),
FOREIGN KEY (dept_id) REFERENCES  departments (dept_id);
)  WITH SYSTEM VERSIONING;
```

- `PERIOD` clause automatically enforces the constraint `system_end` > `system_start`
- The name of the period must be `SYSTEM_TIME`
- The period starts on the `system_start` value and ends on the value just prior to `system_end` value
- This corresponds to the `[closed, open)` model of periods

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Inserting Rows into a System-Versioned Table

- When a row is inserted into a system-versioned table, the SQL-implementation sets the start time to the transaction time and the end time to the largest timestamp value
- All rows inserted in a transaction will get the same values for the start and end columns
- The following INSERT executed at timestamp 15/11/1995

```
INSERT INTO emp (emp_name, dept_id)
VALUES ('John', 'J13'), ('Tracy','K25')
```

  leads to the following table

emp_name	dept_id	system_start	system_end
John	J13	15/11/1995	31/12/9999
Tracy	K25	15/11/1995	31/12/9999

- Values of system_start and system_end are set by DBMS
- N.B. Ony date components of system_start and system_end values are shown for the purpose of simplifying display

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Updating Rows in a System-Versioned Table

- When a row from a system-versioned table is updated, the SQL-implementation inserts the "old" version of the row into the table before updating the row
- SQL-implementation sets the end time of the old row and the start time of the updated row to the transaction time
- Users are not allowed to update the start and end columns
- The following UPDATE executed at timestamp 31/1/1998

```
UPDATE emp
SET dept_id = 'M24'
WHERE emp_name = 'John'
```

leads to the following table

emp_name	dept_id	system_start	system_end
John	M24	31/01/1998	31/12/9999
John	J13	15/11/1995	31/01/1998
Tracy	K25	15/11/1995	31/12/9999

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Deleting Rows from a System-Versioned Table

- When a row from a system-versioned table is deleted, the SQL-implementation does not actually delete the row; it simply sets its end time to the transaction time
- The following DELETE executed on 31/3/2000

```
DELETE FROM emp
WHERE emp_name = 'Tracy'
```

leads to the following table

emp_name	dept_id	system_start	system_end
John	M24	31/01/1998	31/12/9999
John	J13	15/11/1995	31/01/1998
Tracy	K25	15/11/1995	31/03/2000

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying System-Versioned Tables (1)

- Existing syntax for querying regular tables is applicable to system-versioned tables also
- Additional syntax is provided for expressing queries involving system-versioned tables in a more succinct manner:
  - FOR SYSTEM_TIME AS OF <datetime value expression >
  - FOR SYSTEM_TIME BETWEEN < datetime value expression 1 > AND < datetime value expression 2 >
  - FOR SYSTEM_TIME FROM < datetime value expression 1 > TO < datetime value expression 2 >

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying System-Versioned Tables (2)

emp_name	dept_id	system_start	system_end
John	M24	31/01/1998	31/12/9999
John	J13	15/11/1995	31/01/1998
Tracy	K25	15/11/1995	31/03/2000

- Which department was John in on Dec. 1, 1997?

```
SELECT Dept
FROM employees FOR SYSTEM_TIME AS OF DATE '1997-12-01'
WHERE emp_name = 'John'
```

- Answer: `J13`

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Querying System-Versioned Tables (3)

emp_name	dept_id	system_start	system_end
John	M24	31/01/1998	31/12/9999
John	J13	15/11/1995	31/01/1998
Tracy	K25	15/11/1995	31/03/2000

- Which department is John in currently?

```
SELECT Dept
FROM employees
WHERE emp_name = 'John'
```

- Answer: `M24`
- If `AS OF` clause is not specified, only current system rows are returned
  ⇒ `FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP` is the default

Querying System-Versioned Tables (4)

emp_name	dept_id	system_start	system_end
John	M24	31/01/1998	31/12/9999
John	J13	15/11/1995	31/01/1998
Tracy	K25	15/11/1995	31/03/2000

- How many departments has John worked in since Jan. 1, 1996?

```
SELECT count(distinct dept_id)
FROM employees
FOR SYSTEM_TIME BETWEEN DATE '1996-01-01' AND CURRENT_DATE
WHERE emp_name = 'John'
```

- Answer: 2

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Benefits of System-Versioned Tables

- Today's database systems focus mainly on managing current data; they provide almost no support for managing historical data
- Some applications have an inherent need for preserving old data. Examples: job histories, salary histories, account histories, etc.
- Regulatory and compliance laws require keeping old data around for certain length of time
- Currently, applications take on the responsibility for preserving old data
- Major issues
  - Complexity of code
  - Poor performance
- System-versioned tables provides
  - Significant simplification of application code
  - Significant improvement in performance
  - Transparent to legacy applications

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

System-Versioned Application-Time Period Tables

- A table that is both an application-time period table and a system-versioned table
- Such a table supports features of both application-time period tables and system-versioned tables
- Creating a system-versioned application-time period table

```
CREATE TABLE employees
(emp_name VARCHAR(50) NOT NULL PRIMARY KEY,
dept_id VARCHAR(10),
start_date DATE NOT NULL,
end_date DATE NOT NULL,
system_start TIMESTAMP(6) GENERATED ALWAYS AS ROW START,
System_end TIMESTAMP(6) GENERATED ALWAYS AS ROW END,
PERIOD FOR emp_period (start_date, end_date),
PERIOD FOR SYSTEM_TIME (system_start, system_end),
PRIMARY KEY (emp_name, emp_period WITHOUT OVERLAPS),
FOREIGN KEY (dept_id, PERIOD emp_period) REFERENCES
    departments (dept_id, PERIOD dept_period)
) WITH SYSTEM VERSIONING;
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Insert

- On 11/01/1995, employees table was updated to show that John and Tracy will be joining the departments J13 and K25, respectively, starting from 15/11/1995

```
INSERT INTO employees (emp_name, dept_id, start_date, end_date)
VALUES ('John', 'J13', DATE '1995-11-15', DATE '9999-12-31'),
       ('Tracy','K25', DATE '1995-11-15', DATE '9999-12-31')
```

emp_name	dept_id	start_date	end_date	system_start	system_end
John	J13	15/11/1995	31/12/9999	11/1/1995	31/12/9999
Tracy	K25	15/11/1995	31/12/9999	11/1/1995	31/12/9999

- `system_start` and `system_end` values are set by the system
- N.B. `DATE` type is used in examples instead of `TIMESTAMP` type to simplify display

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Update

- On 11/10/1995, it was discovered that John was assigned to the wrong department; it was changed to department J15 on that day

```
UPDATE employees
SET dept_id = 'J15'
WHERE emp_name = 'John'
```

- This leads to the following table

emp_name	dept_id	start_date	end_date	system_start	system_end
John	J15	15/11/1995	31/12/9999	11/10/1995	31/12/9999
John	J13	15/11/1995	31/12/9999	11/01/1995	11/10/1995
Tracy	K25	15/11/1995	31/12/9999	11/1/1995	31/12/9999

Partial Period Update

- On 15/12/1997, John is loaned to department M12 starting from 1/1/1998 to 7/1/1998

```
UPDATE employees FOR PORTION OF emp_period
FROM DATE '1998-01-01' TO DATE '1998-07-01'
SET dept_id = 'M12' WHERE emp_name = 'John'
```

- This leads to the following table

emp_name	dept_id	start_date	end_date	system_start	system_end
John	J15	01/07/1998	31/12/9999	15/12/1997	31/12/9999
John	M12	15/11/1995	31/12/9999	15/12/1997	31/12/9999
John	J15	15/11/1995	31/12/9999	15/12/1997	31/12/9999
John	J15	15/11/1995	31/12/9999	11/10/1995	15/12/1997
John	J15	15/11/1995	31/12/9999	11/10/1995	31/12/9999
Tracy	K25	15/11/1995	31/12/9999	11/1/1995	31/12/9999

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Partial Period Delete

- On 15/12/1998, John is approved for a leave of absence from 1/1/1999 to 1/1/2000

```
DELETE FROM employees
FOR PORTION OF emp_period FROM DATE '1999-01-01' TO DATE '2000-01-01'
WHERE emp_name = 'John'
```

- This leads to the following table

emp_name	dept_id	start_date	end_date	system_start	system_end
John	J15	01/01/2000	31/12/9999	15/12/1998	31/12/9999
John	J15	01/07/1998	31/12/9999	15/12/1998	31/12/9999
John	J15	01/07/1998	31/12/9999	15/12/1997	31/12/9999
John	M12	15/11/1995	31/12/9999	15/12/1997	31/12/9999
John	J15	01/07/1998	31/12/9999	15/12/1997	15/12/1998
John	J15	15/11/1995	31/12/9999	11/10/1995	15/12/1997
John	J15	15/11/1995	31/12/9999	11/10/1995	31/12/9999
Tracy	K25	15/11/1995	31/12/9999	11/01/1995	31/12/9999

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Delete

- On 1/6/2000, John resigns from the company

```
DELETE FROM employees
WHERE emp_name = 'John'
```

- This leads to the following table

emp_name	dept_id	start_date	end_date	system_start	system_end
John	J15	01/01/2000	31/12/9999	15/12/1998	01/06/9999
John	J15	01/07/1998	31/12/9999	15/12/1998	01/06/9999
John	J15	01/07/1998	31/12/9999	15/12/1997	01/06/9999
John	M12	15/11/1995	31/12/9999	15/12/1997	01/06/9999
John	J15	01/07/1998	31/12/9999	15/12/1997	15/12/1998
John	J15	15/11/1995	31/12/9999	11/10/1995	15/12/1997
John	J13	15/11/1995	31/12/9999	11/10/1995	31/12/9999
Tracy	K25	15/11/1995	31/12/9999	11/01/1995	31/12/9999

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Temporal Databases: Conclusion

- Temporal information is ubiquitous in every application domain
- Such information should be included in the overall software lifecyle: from design to implementation
- Necessity of a temporal conceptual model for discussing requirements with users
- Manipulating temporal information in standard SQL is
  - Very difficult to program
  - Very inefficient
- Native temporal capabilities are needed in DBMSs
- Recent SQL standard has introduced such capabilities after more than a decade of debates
- Such capabilities have still to be implemented in the different platforms
- Data warehouses have included temporal capabilities since their begining a few decades ago
- Temporal capabilities are usally combined with spatial capabilities $\Rightarrow$ spatio-temporal databases

Spatial Databases: Topics

- **Introduction**
- Georeferences and Coordinate Systems
- Conceptual Modeling for Spatial Databases
- Logical Modeling for Spatial Databases
- SQL/MM
- Representative Systems
- Summary

Foreword

- In 1854 in Soho (London) doctor John Snow was able to identify the water point at the origin of the cholera epidemy by correlating the location of the water points and the location of the victims

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

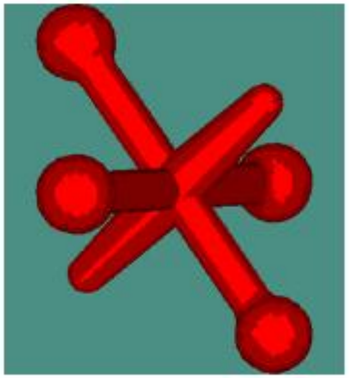
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

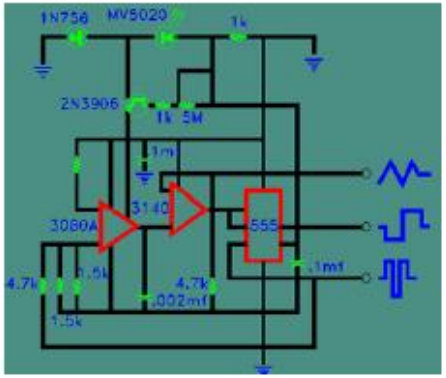
Spatial Databases

- A database that needs to store and query spatial objects, e.g.
  - Point: a house, a monument
  - Line: a road segment, a road network
  - Polygon: a county, a voting area
- Types of spatial data

GIS Data



CAD Data



CAM Data

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Database Management Systems

- A Database Management System that manages data existing in some space
- 2D or 2.5D
  - Integrated circuits: VLSI design
  - Geographic space (surface of the Earth): GIS, urban planning
- 2.5 D: Elevation
- 3D
  - Medicine: Brain models
  - Biological research: Molecule structures
  - Architecture: CAD
  - Ground models: Geology
- Supporting technology able to manage large collections of geometric objects
- Major commercial and open-source DBMSs provide spatial support

Why Spatial Databases?

- Queries to databases are posed in high level declarative manner (usually using SQL)
- SQL is popular in the commercial database world
- Standard SQL operates on relatively simple data types
- Additional spatial data types and operations can be defined in spatial database
- SQL was extended to support spatial data types and operations, e.g., OGC Simple Features for SQL
- A DBMS is a way of storing information in a manner that
  - Enforces consistency
  - Facilitates access
  - Allows users to relate data from multiple tables together

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Application Areas

- Street network-based
  - Vehicle routing and scheduling (cars, planes, trains)
  - Location analysis, ...
- Natural resource-based
  - Management of areas: agricultural lands, forests, recreation resources, wildlife habitat analysis, migration routes planning...
  - Environmental impact analysis
  - Toxic facility siting
  - Groundwater modeling, ...
- Land parcel-based
  - Zoning, subdivision plan review
  - Environmental impact statements
  - Water quality management
  - Facility management: electricity, gaz, clean water, used water, network

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Interaction with End Users

- Display data (e.g., maps) on the screen
- Access other data by clicking on it (hypermaps)
- Address queries
- Perform operations

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Example of Queries

- On a subway map of Brussels, what is the shortest way from here to the Grand Place?
  $\Rightarrow$ shortest path algorithm
- Overlay the land use map with the map of districts in Belgium
- Display today's weather forecast in the Brussels Region
- Given the map of a neighborhood, find the best spot for opening a drugstore (based on a given set of optimality criteria) $\Rightarrow$ allocation problem

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Geographic Information Systems

- A system designed to capture, store, manipulate, analyze, manage, and present geographically-referenced data as well as non-spatial data
- Connection between system elements is geography, e.g., location, proximity, spatial distribution
- Common purpose: decision making for managing use of land, resources, ocean data, transportation, geomarketing, urban planning, etc.
- Many commercial and open source systems, but limited temporal support

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

GIS as a Set of Subsystems

- Data processing
  - Data acquisition (from maps, images): input, store
- Data analysis
  - Retrieval, analysis (answers to queries, complex statistical analyses)
- Information use
  - Users: Researchers, planners, managers
  - Interaction needed between GIS group and users to plan analytical procedures and data structures
- Management system
  - Organizational role: separate unit in a resource management
  - Agency offering spatial database and analysis services
  - System manager, system operator, system analysts, digitizer operators

Contributing Disciplines and Technologies (1)

- Convergence of technological fields and traditional disciplines
- **Geography**: The study of the Earth
- **Cartography**: Display of spatial information
  - Computer cartography (digital, automated cartography): methods for digital representation, manipulation and visualization of cartographic features
- **Remote sensing**: Techniques for image acquisition and processing (space images)
- **Photogrammetry**: Aerial photos and techniques for accurate measurements
- **Geodesy**: The study of the size and shape of the earth
- **Statistics**: Statistical techniques for analysis + errors and uncertainty in GIS data
- **Operations Research**: Optimizing techniques for decision making
- **Mathematics**: (Computational) geometry and graph theory for analysis of spatial data
- **Civil Engineering**: Transportation, urban engineering

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Contributing Disciplines and Technologies (2)

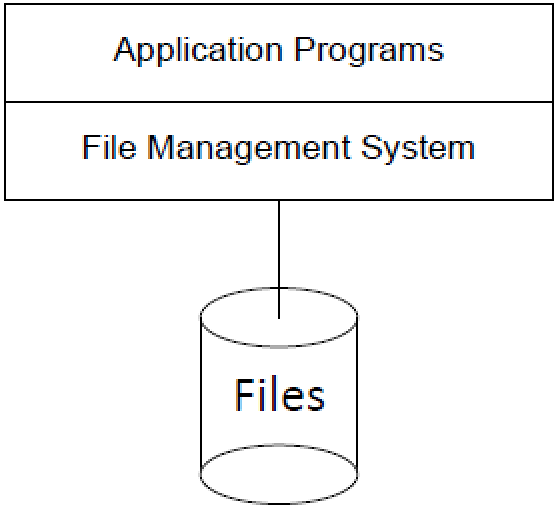
- **Computer Science**
  - Computer Aided Design (CAD): Software, techniques for data input, display, representation, visualization
  - Computer graphics: Hardware, software for handling graphic objects
  - Artificial Intelligence: Emulate human intelligence and decision making (computer = "expert").
  - Database Management Systems (DBMS): Representing, handling large volumes of data (access, updates)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

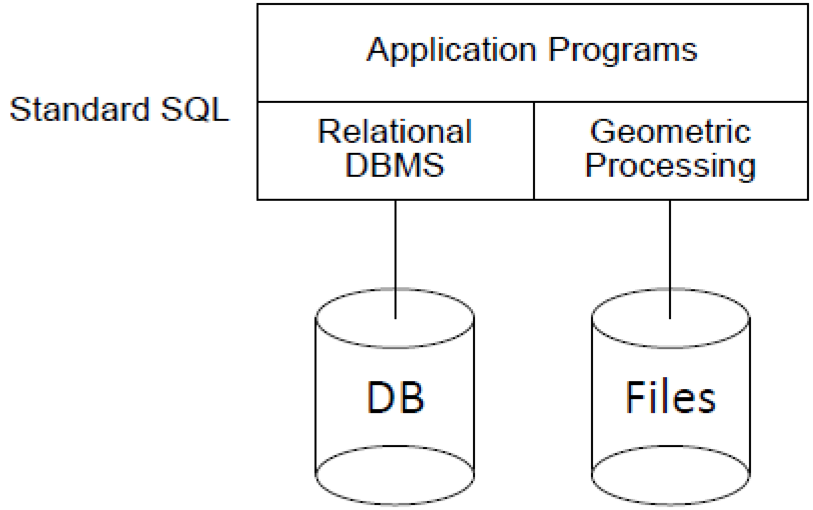
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

GIS Architectures: Ad Hoc Systems



- Not modular
- Not reusable
- Not extensible
- Not friendly
- Very efficient

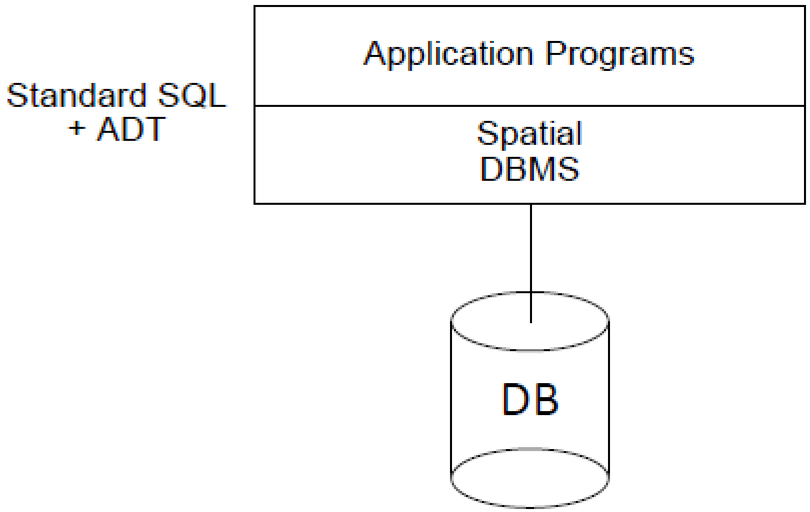
GIS Architectures: Loosely Coupled Approach



- Structured information and geometry stored at different places
  - Relational DBMS for alphanumerical (non spatial) data
  - Specific module for spatial data management
- Modularity (use of a DBMS) BUT:
  - Heterogeneous models! Difficult to model, integrate and use
  - Partial loss of basic DBMS functionalities e.g., concurrency, optimization, recovery, querying

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

GIS Architectures: Integrated Approach



- “Extended relational” system
- Modular
- Extensible (add a new operation, a new type)
- Reusable
- Friendly (easy change, easy use)

Main Database Issues

- Data modeling
- (Spatial) query language
- Query optimization
- Spatial access methods define index structure to accelerate the retrieval of objects
- Versioning
- Data representation (raster/vector)
- Graphical user interfaces (GUI)
- Computational geometry for GIS

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Topics

- Introduction
- **Georeferences and Coordinate Systems**
- Conceptual Modeling for Spatial Databases
- Logical Modeling for Spatial Databases
- SQL/MM
- Representative Systems
- Summary

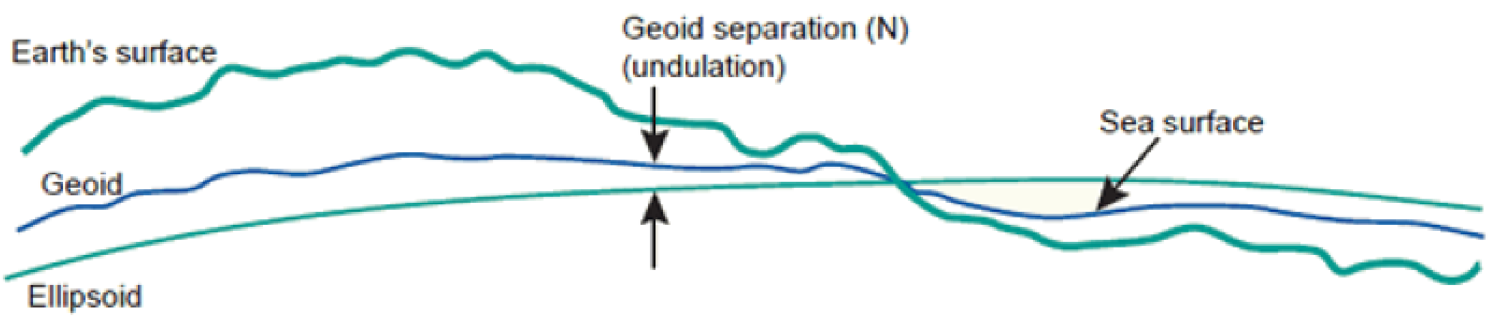
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

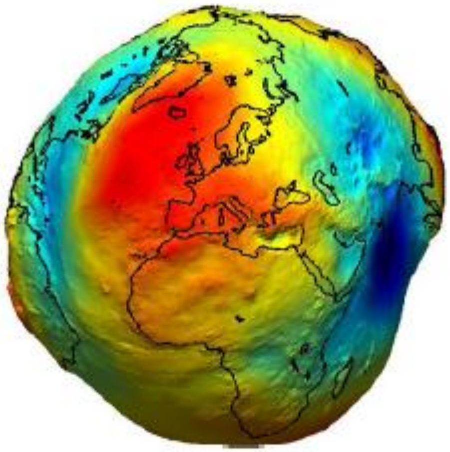
Projected Coordinates Systems

- Why focus on projections?
  - Going from 3 to 2 dimensions for Earth representation always involves a projection
  - Information on the projection is essential for applications analyzing spatial relationships
  - Choice of the projection can influence the results
- The Earth is a complex surface whose shape and dimensions can not be described with mathematical formulas
- Two main reference surfaces are used to approximate the shape of the Earth: the **ellipsoid** and the **Geoid**



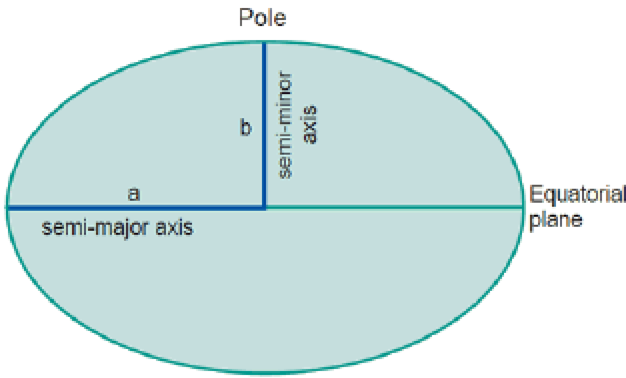
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

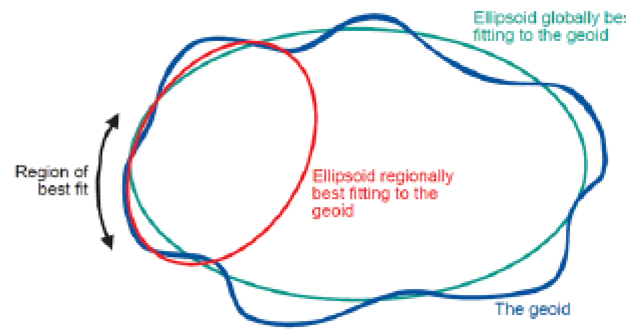
The Geoid



- Reference model for the physical surface of the Earth
- It is the equipotential surface of the Earth's gravity field which best fits the global mean sea level and extended through the continents
- Used in geodesy, e.g., for measuring heights represented on maps
- Not very practical to produce maps
- Since the mathematical description of the geoid is unknown, it is impossible to identify mathematical relationships for moving from the Earth to the map

Ellipsoid

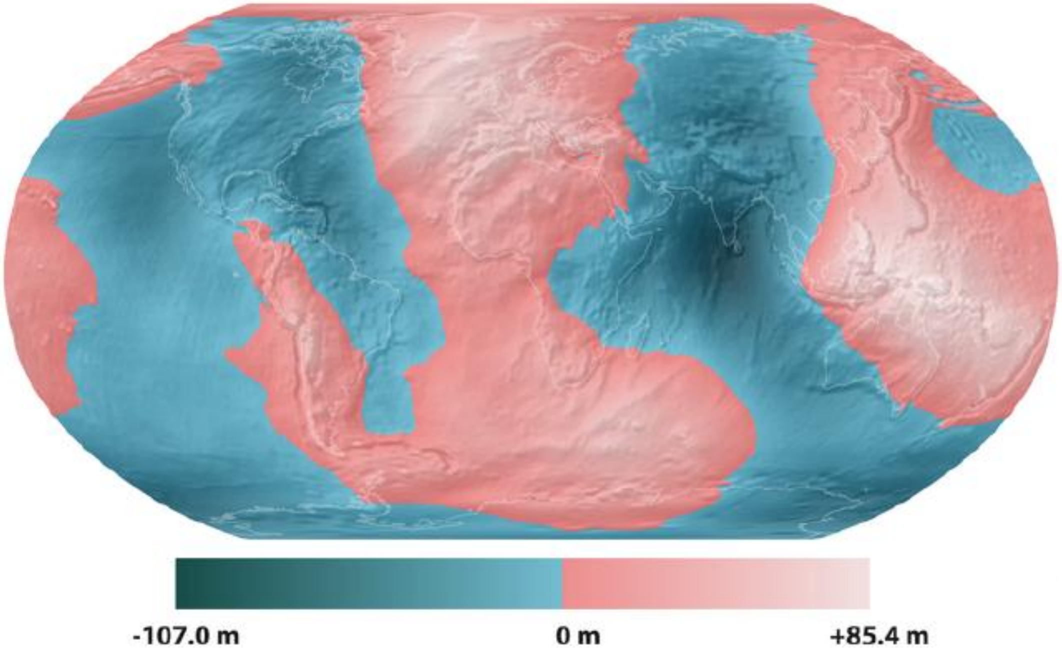




- A mathematically-defined surface that approximates the geoid
- Flattening $f = \frac{a-b}{a}$: how much the symmetry axis is compressed relative to the equatorial radius
- For the Earth, f is around 1/300: difference of the major and minor semi-axes of approx. 21 km
- Ellipsoid used to measure locations, the latitude (ϕ) and longitude (λ), of points of interest
- These locations on the ellipsoid are then projected onto a mapping plane
- Different regions of the world use different reference ellipsoids that minimize the differences between the geoid and the ellipsoid
- For Belgium (since 2008) ellipsoid GRS80 (=WGS84) with $a = 6{,}378{,}137$ m. and $f = 1/298.257222101$

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Difference between the EGM96 Geoide and the WGS84 Ellipsoid



- The physical Earth has excursions of +8,000 m (Mount Everest) and -11,000 m (Mariana Trench)
- The geoid's total variation goes from -107m to +85 m compared to a perfect mathematical ellipsoid

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

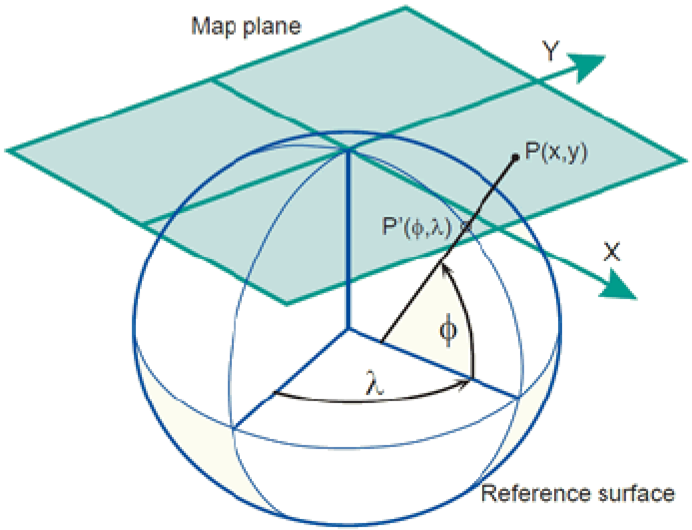
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Latitude and Longitude

- Measures of the angles (in degrees) from the center of the Earth to a point on the Earth's surface
- Latitude measure angles in a North-South direction: the Equator is at an angle of 0
- Longitude measures angles in an East-West direction: the Prime Meridian is at angle of 0
- Prime Meridian: imaginary line running from the North to the South Pole through Greenwich, England
- For example, the location of Brussels, Belgium is 50.8411° N, 4.3564° E
- Longitude and latitude are not uniform units of measure
- Only along the Equator the distance represented by one degree of longitude approximate the distance represented by one degree of latitude

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Map Projections

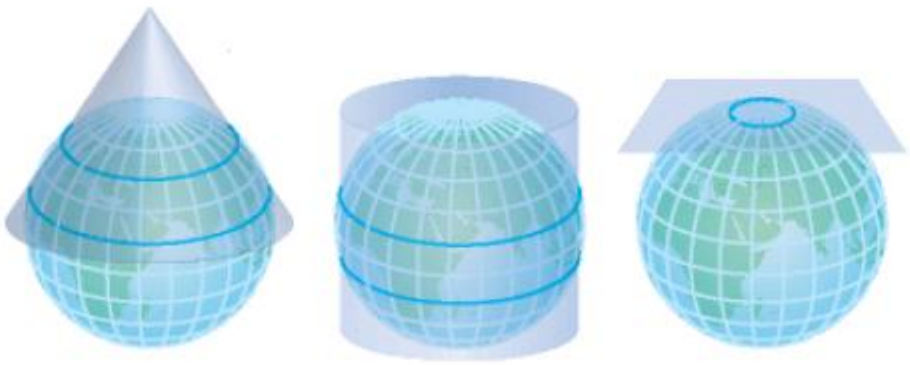


- To produce a map, the curved reference surface of the Earth, approximated by an ellipsoid or a sphere, is transformed into the flat plane of the map by means of a map projection
- A point on the reference surface of the Earth with geographic coordinates (ϕ, λ) is transformed into Cartesian (or map) coordinates (x, y) representing positions on the map plane
- Each projection causes deformations
- Map projections can be categorised in four ways: Class, Angle, Fit and Properties

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Map Projections: Shape of Projection Surface



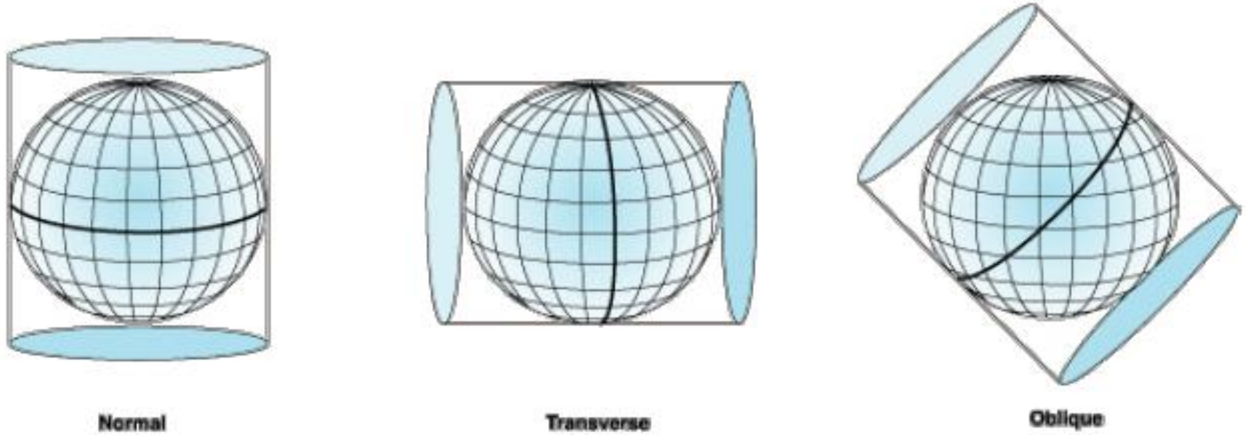
- Shape of the projection surface, commonly either a flat plane, a cylinder or a cone
- Cones and cylinders are not flat shapes, but they can be rolled flat without introducing additional distortion
  - **Cylindrical**: coordinates are projected onto a rolled cylinder
  - **Conical**: coordinates are projected onto a rolled cone
  - **Azimuthal**: coordinates are projected directly onto a flat planar surface
- Azimuthal projections work best for circular areas (e.g., the poles)
- Cylindrical projections work best for rectangular areas (e.g., world maps)
- Conical projections work best for triangle shaped areas (e.g., continents)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Map Projections: Angle



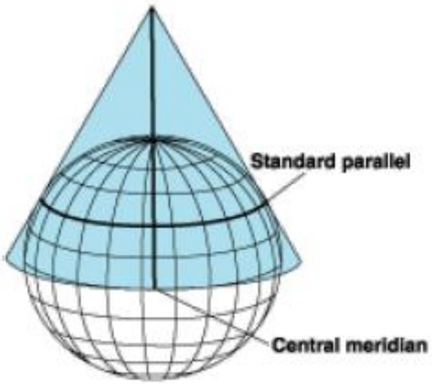
- This refers to the alignment of the projection surface, measured as the angle between the main axis of the earth and the main symmetry axis of the projection surface.
  - **Normal**: the two axes are parallel
  - **Transverse**: the two axes are perpendicular
  - **Oblique**: the two axes are at some other angle
- Ideally the plane of projection is aligned as closely as possible with the main axis of the area to be mapped. This helps to minimise distortion and scale error.

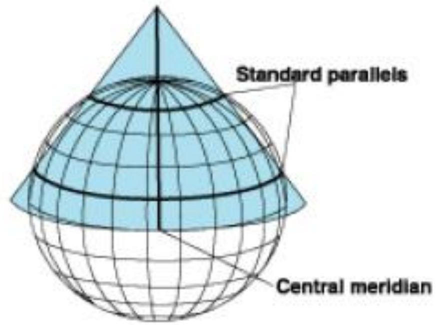
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

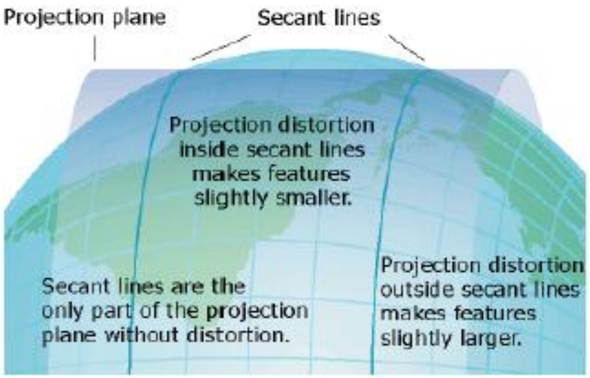
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Map Projections: Fit







- A measure of how closely the projection surface fits the surface of the earth.
  - **Tangent**: the projection surface touches the surface of the earth.
  - **Secant**: the projection surface slices through the earth.
- Distortion occurs wherever the projection surface is not touching or intersecting the surface of the Earth
- Secant projections usually reduce scale error because the two surfaces intersect in more places and the overall fit tends to be closer
- A globe is the only way to represent the entire Earth without any significant scale error

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

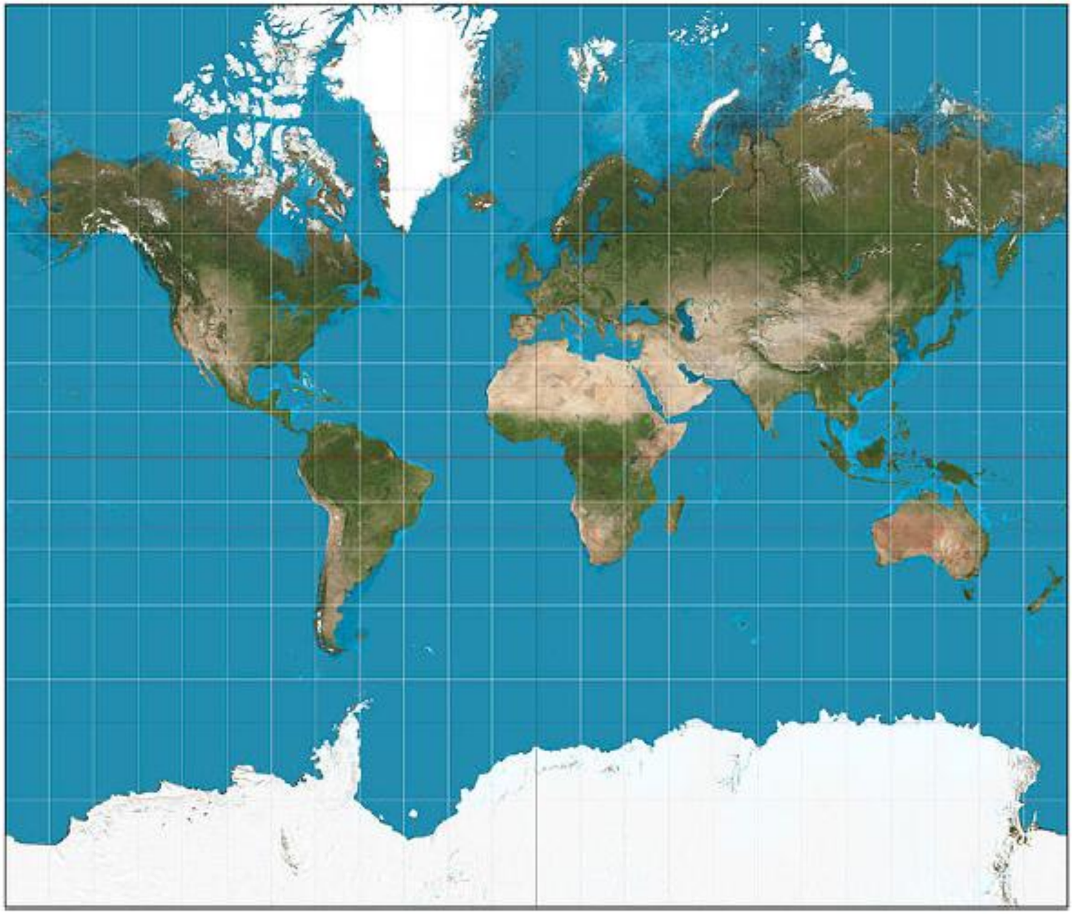
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Map Projections: Geometric Deformations

- What is preserved
  - **Conformal**: preserve shapes and angles
  - **Equal Area** (or **equivalent**): preserve areas in correct relative size (shapes not preserved)
  - **Equidistant**: preserve distance (this is only possible at certain locations or in certain directions)
  - **True-direction** (or **Azimuthal**): preserves accurate directions (e.g., angles preserved, but length of lines is not)
- It is impossible to construct a map that is both equal-area and conformal
- Conformal map projections are recommended for navigational charts and topographic maps
- Equal area projections are generally best for thematic mapping
- Equidistant map projections should be used when measuring distances from a point: air routes, radio propagation strength, radiation dispersal

Conformal Cylindrical Projection: Mercator Projection

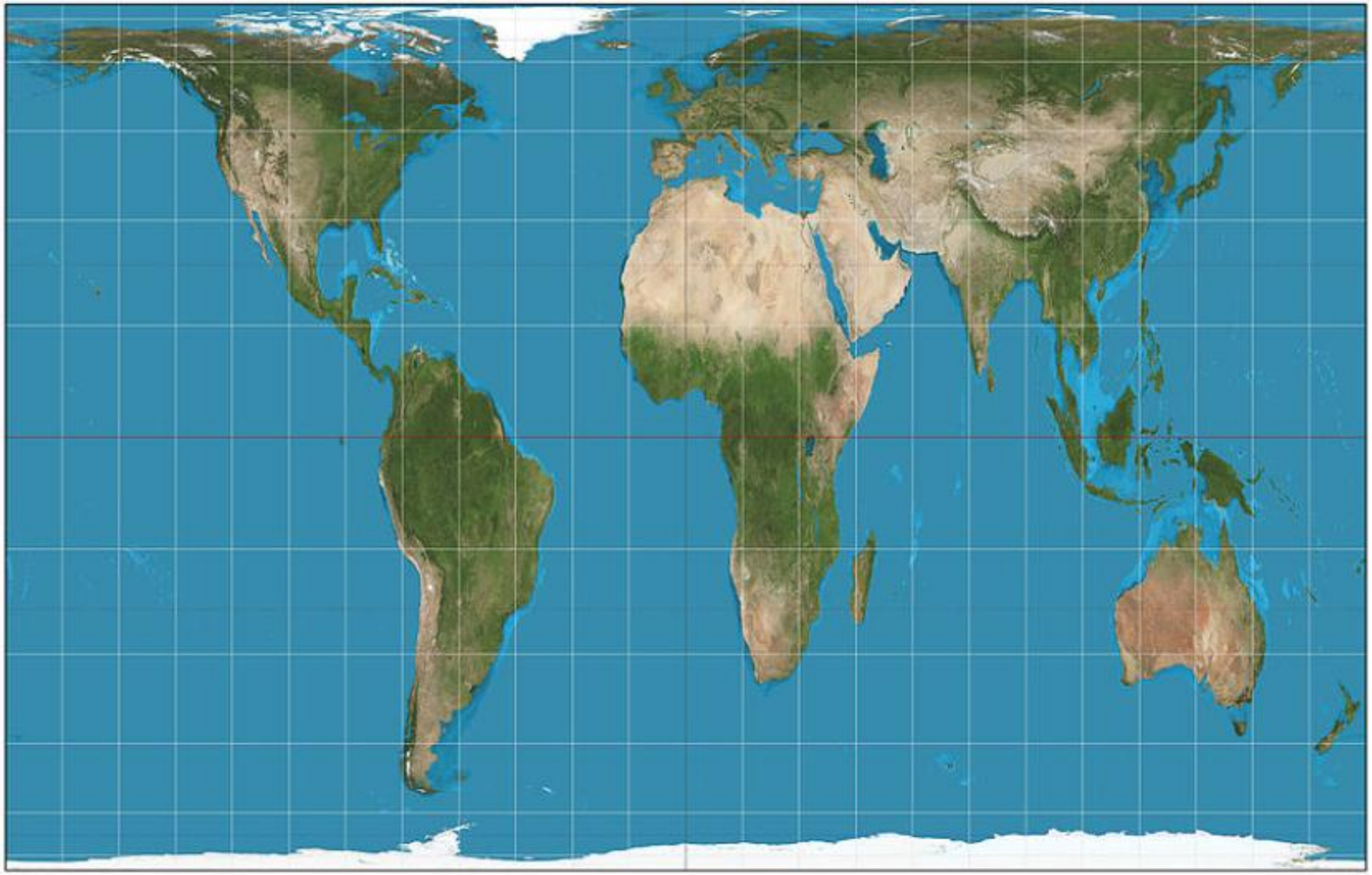


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Cylindrical Equal-Area Projection: Peters Projection

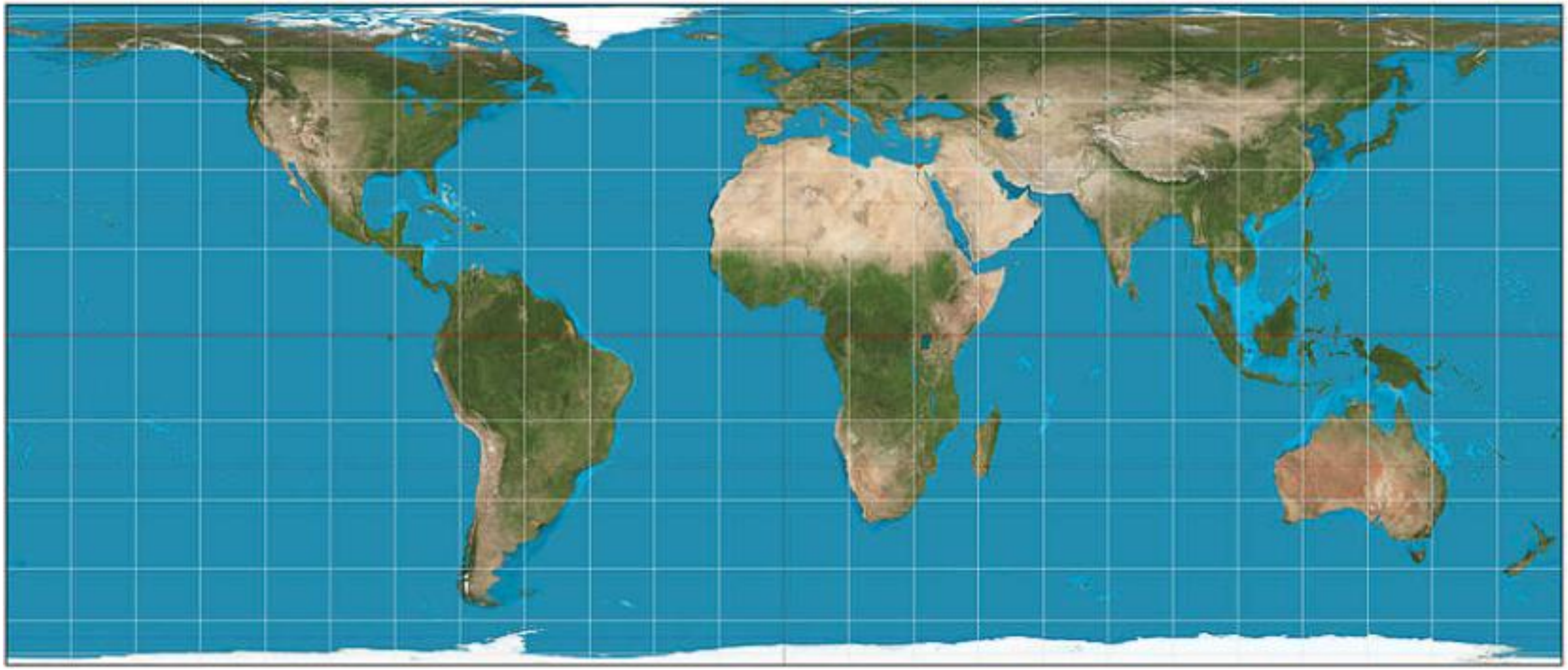


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

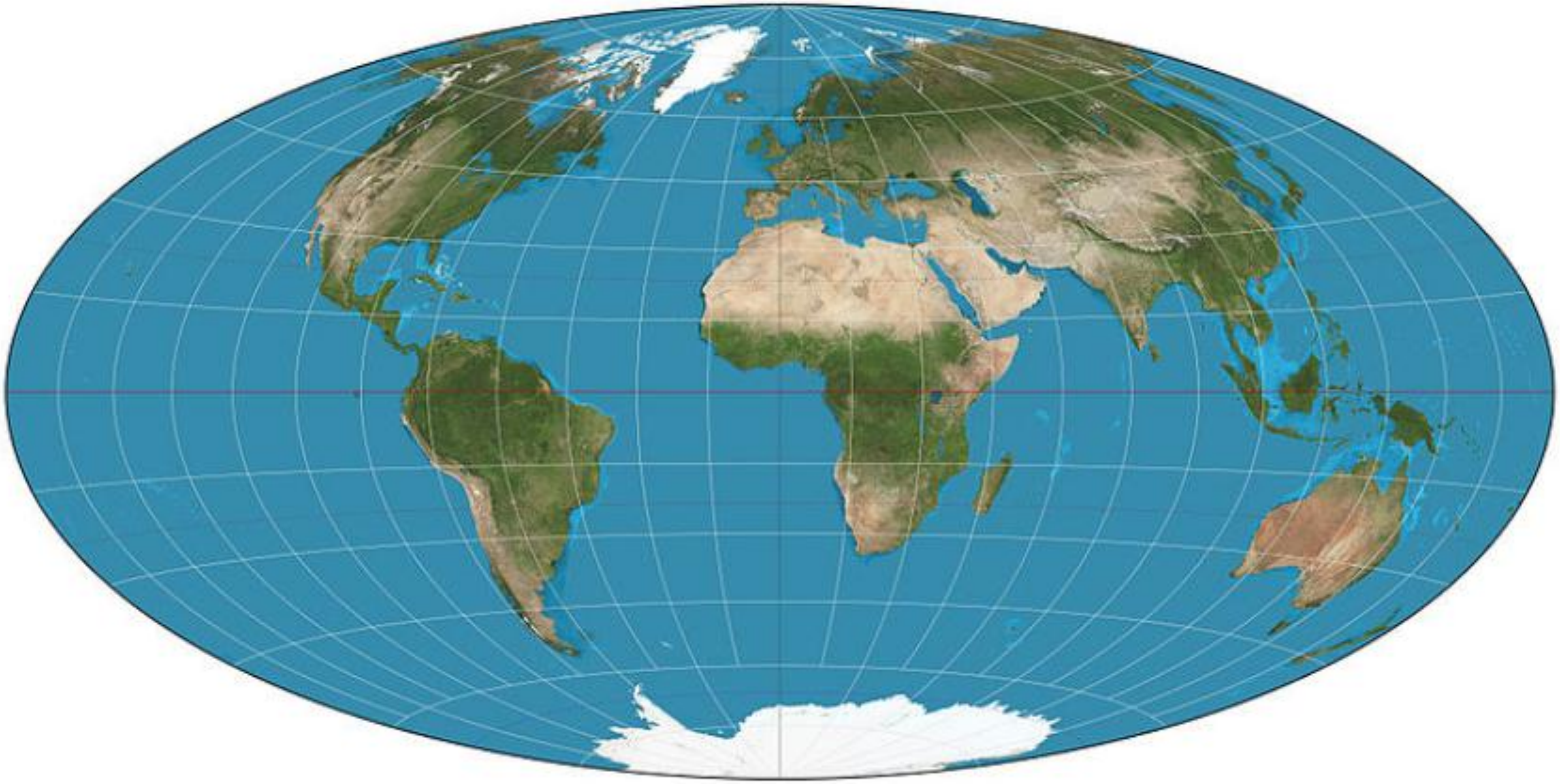
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Cylindrical Equal-Area Projection: Behrmann Projection



Elliptical Equal-Area projection: Hammer Projection



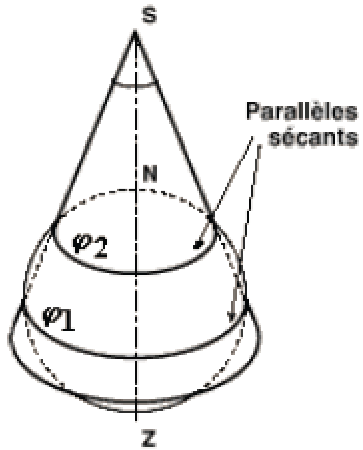
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

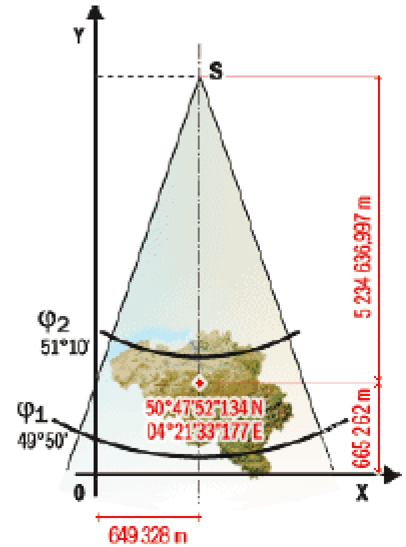
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

True-direction Azimuthal Projection

Belgium: Projection Lambert 2008





- Projection plane: Conical
- Direction of the projection axis: Normal
- Nature of the contact: Secant
- Nature of the deformation: Conformal

Parameters

Ellipsoid	Identity	GRS80
	semi-major axis a	6,378,137 m
	flattening f	1/298.257222101
Standard Parallels	φ_1	49° 50' N
	φ_2	51° 10' N
Origine	Origin Latitude	50° 47' 52" 134 N
	Central meridian	4° 21' 33" 177 E
Origin Coordinates	x_0	649,328 m
	y_0	665,262 m

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Topics

- Introduction
- Georeferences and Coordinate Systems
- ➡ **Conceptual Modeling for Spatial Databases**
- Logical Modeling for Spatial Databases
- SQL/MM
- Representative Systems
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Data Modeling Requirements

- Multiple views of space
  - discrete (objects), continuous (fields)
  - 2D, 2.5D, 3D
- Multiple representations
  - different scales, different viewpoints
- Several spatial abstract data types
  - point, line, area, set of points, set of lines, set of areas, . . .
- Explicit spatial relationships
  - crossing, adjacency, . . .

Interaction Requirements

- Visual interactions
  - map displays
  - information visualization
  - graphical queries on maps
- Flexible, context-dependent interactions
- Multiple user profiles
  - highway: constructor, car driver, truck driver, hiker, ecologist
- Multiple instantiations
  - a building may be a school and a church
  - a road segment may be also a segment of a hiking trail

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Practical Requirements

- Huge data sets
  - Collecting new data is expensive
  - Reusing highly heterogeneous existing data sets is a must ... but is very difficult!
  - Integration requires understanding, hence a conceptual model
- Integration of DB with different space/time granularity
- Coexistence with non-spatial, non-temporal data
- Reengineering of legacy applications
- Interoperability

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Why Conceptual Modeling?

- Focuses on the application
- Technology independent
  - portability, durability
- User oriented
- Formal, unambiguous specification
- Supports visual interfaces
  - data definition and manipulation
- Best vehicle for information exchange/integration

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

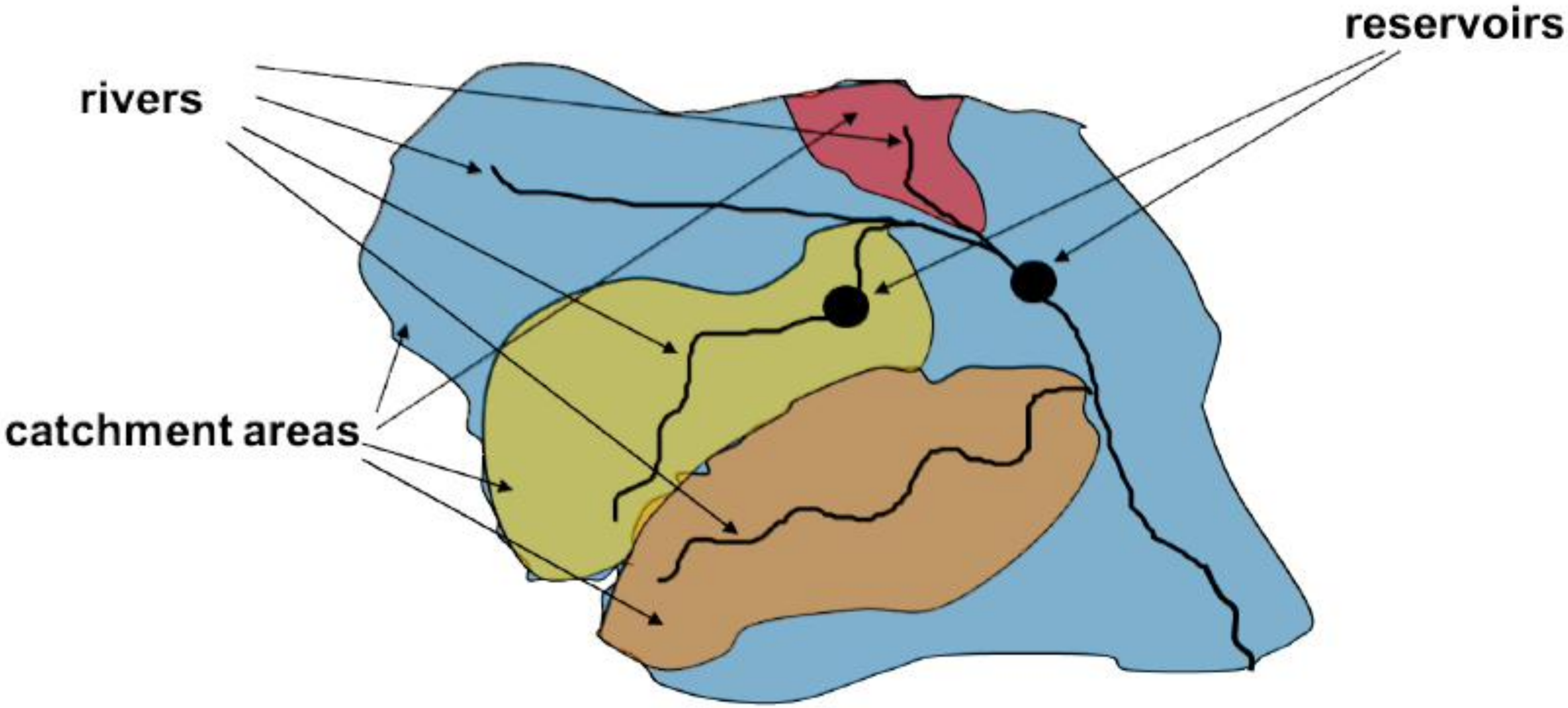
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

The Spatiotemporal Conceptual Manifesto

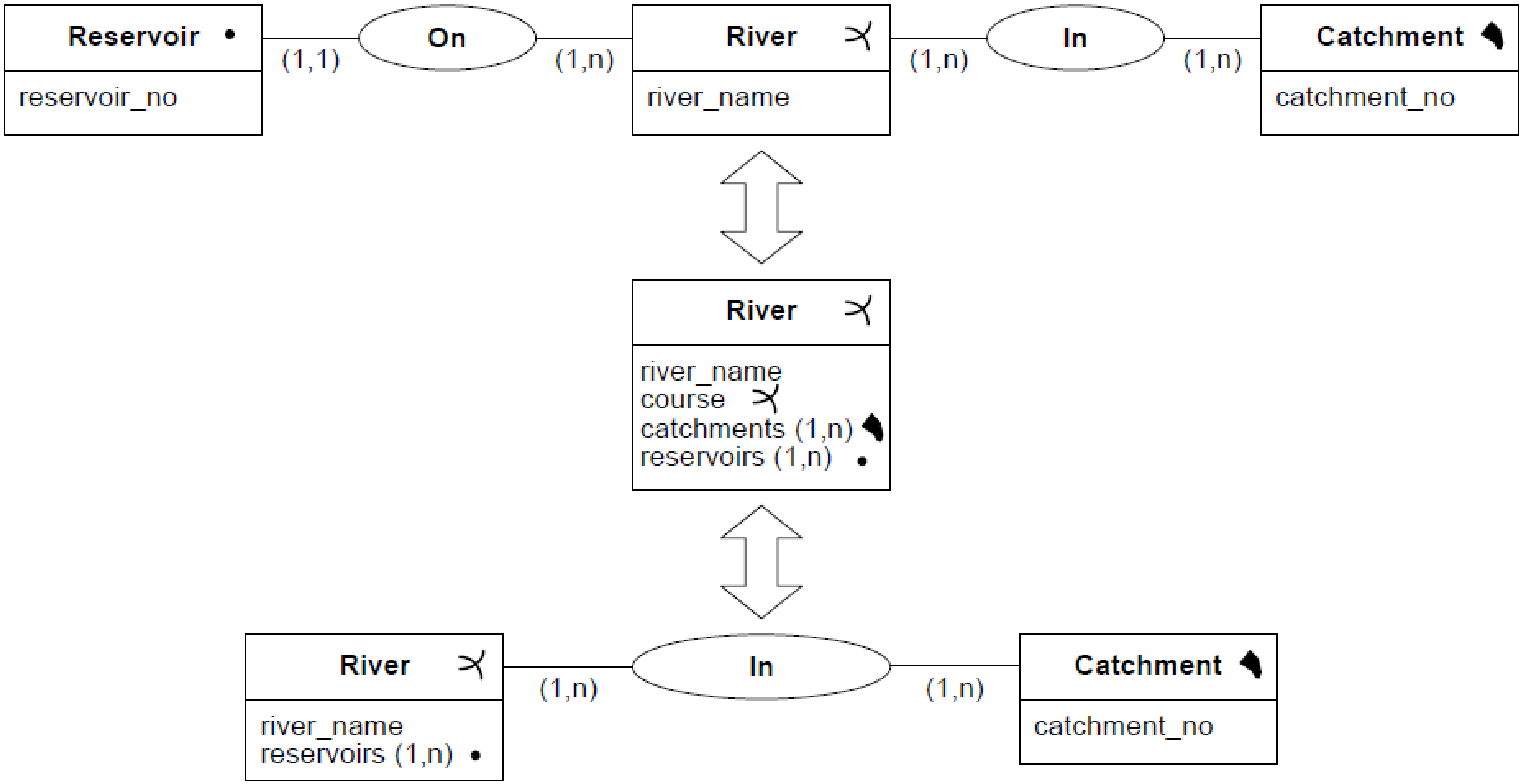
- Good expressive power
- Simple (understandable) data model
  - few clean concepts, with standard, well-known semantics
- No artificial constructs (e.g., space / time objects)
- Orthogonality of space, time and data structures
- Similarity of concepts for space and time
- Clean, visual notations and intuitive icons / symbols
- Formal definition
- Associated query language

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Orthogonality: What is Spatial



Same Space, Different Data Structures

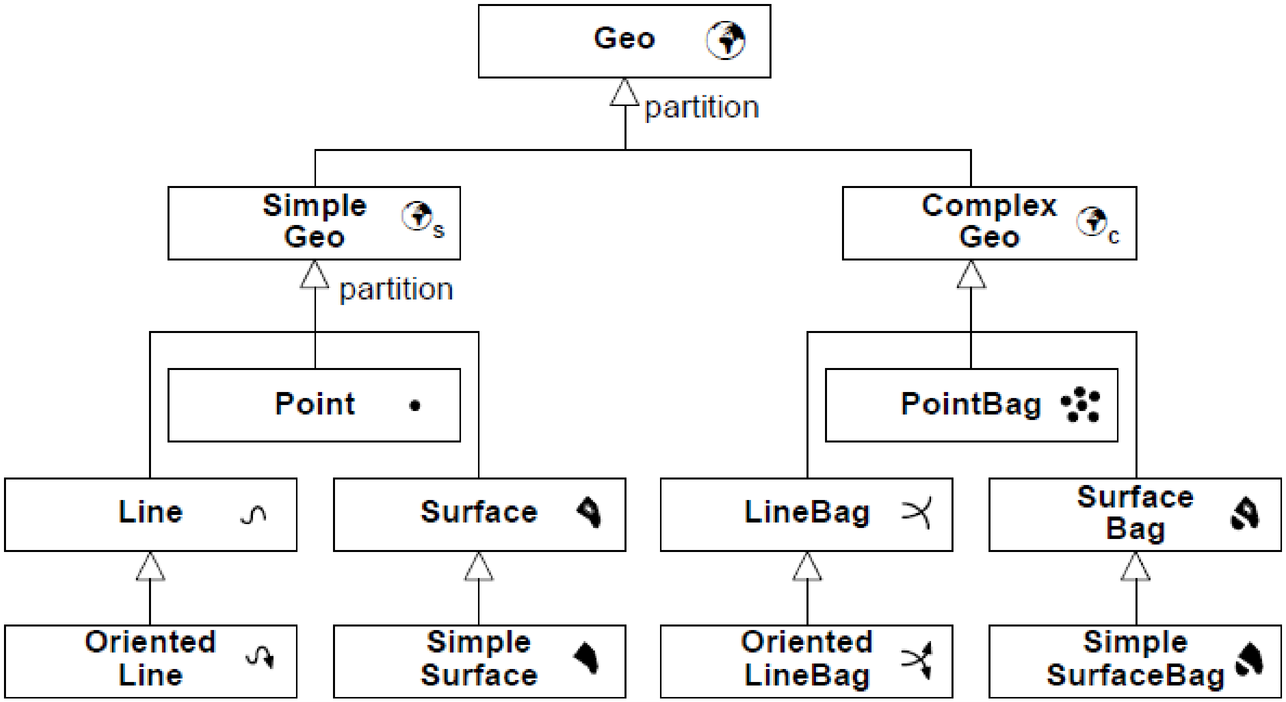


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

MADS Spatial Data Types: Type Hierarchy



- Spatial data types are **topologically closed**: all geometries include their boundary
- `Geo`, `SimpleGeo`, and `ComplexGeo` are abstract classes

MADS Spatial Data Types (1)

Point

- Boundary of a point: empty set

Line

simple and open | non-simple and open | partially closed | closed and simple

- Boundary of a line: the extreme points, if any

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

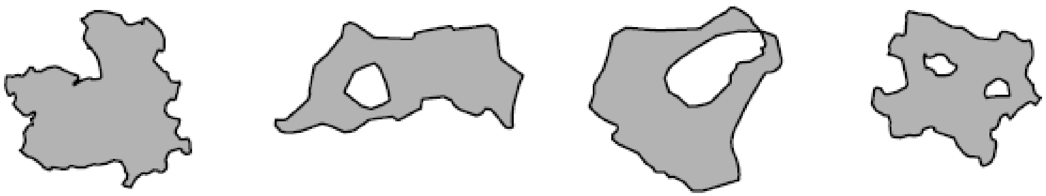
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

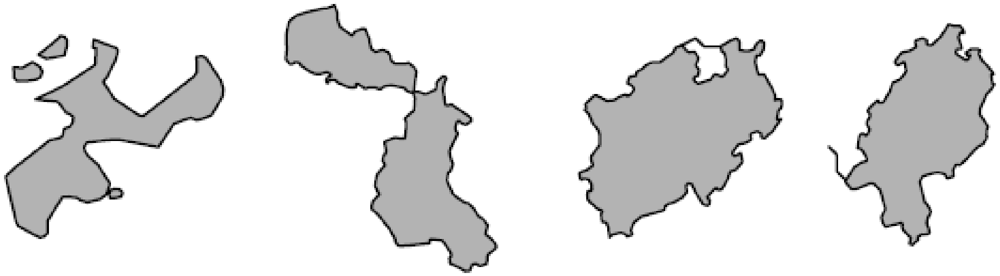
MADS Spatial Data Types (2)

Surface

- Defined by 1 exterior boundary and 0 or more interior boundaries defining its holes



- Examples of geometries that are not surfaces

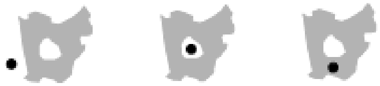
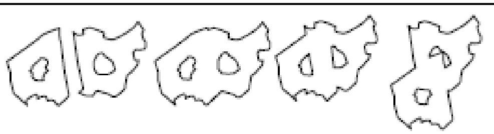


several surfaces several surfaces has a cut line has a spike

Complex Geometries and Their Boundaries

- Boundary of a ComplexGeo value is defined (recursively) by the spatial union of
  - (1) the boundaries of its components that do not intersect with other components
  - (2) the intersecting boundaries that do not lie in the interior of their union

$$B(a \cup b) = B(a) - b \cup B(b) - a \cup (\ (B(a) \cap b) \cup (B(b) \cap a) - I(a \cup b)\ )$$

Types	a ∪ b	B(a ∪ b)
Point/ Point		
Point/ Line		
Point/ Surface		
Line/ Line		
Line/ Surface		
Surface/ Surface		

Topological Predicates

- Specify how two geometries relate to each other
- Based on the definition of their **boundary**, **interior**, and **exterior**, denoted by $I(x)$, $B(x)$, and $E(x)$
- $Dim(x)$: maximum dimension (-1, 0, 1, or 2) of x, -1 corresponds to the dimension of the empty set
- Dimensionally extended 9-intersection matrix (DE-9IM) for defining topological predicates

	Interior	Boundary	Exterior
Interior	$Dim(I(a) \cap I(b))$	$Dim(I(a) \cap B(b))$	$Dim(I(a) \cap E(b))$
Boundary	$Dim(B(a) \cap I(b))$	$Dim(B(a) \cap B(b))$	$Dim(B(a) \cap E(b))$
Exterior	$Dim(E(a) \cap I(b))$	$Dim(E(a) \cap B(b))$	$Dim(E(a) \cap E(b))$

- Dense notation use a string of 9 characters to represent the cells of the matrix
- Possible characters: T (non-empty intersection), F (empty intersection), 0, 1, 2, * (irrelevant)
- Example: a and b are disjoint if their intersection is empty

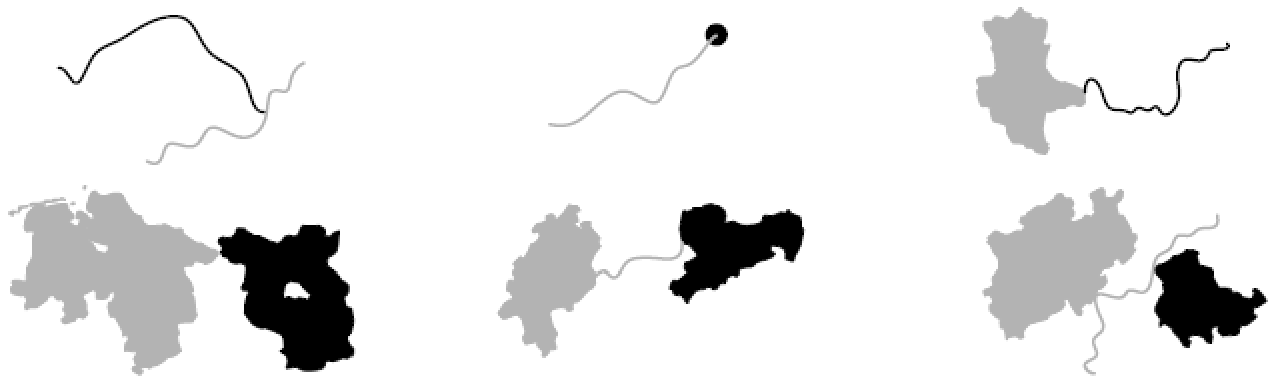
$$I(a) \cap I(b) = \varnothing \wedge I(a) \cap B(b) = \varnothing \wedge B(a) \cap I(b) = \varnothing \wedge B(a) \cap B(b) = \varnothing$$

corresponds to 'FF*FF****'

Topological Predicates (1)

Meets

- a meets $b \Leftrightarrow I(a) \cap I(b) = \varnothing \wedge a \cap b \neq \varnothing \wedge Dim(a \cap b) = 0$

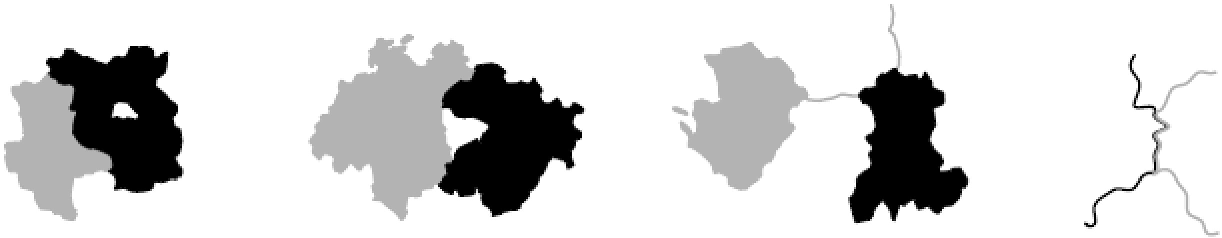


- Examples of geometries that do not satisfies the meets predicate

Topological Predicates (2)

Adjacent

- a adjacent $b \Leftrightarrow I(a) \cap I(b) = \varnothing \wedge a \cap b \neq \varnothing \wedge Dim(a \cap b) = 1$



- The last example does not satisfies the predicate, their intersection is at the interior of both geometries

Touches

- a touches $b \Leftrightarrow I(a) \cap I(b) = \varnothing \wedge a \cap b \neq \varnothing \Leftrightarrow a$ meets $b \vee a$ adjacent b

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

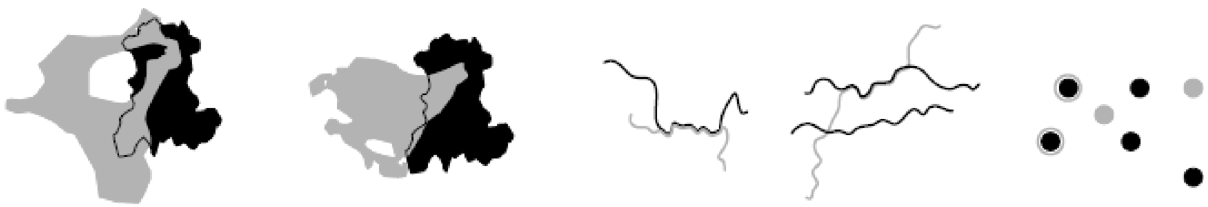
Topological Predicates (3)

Crosses

- a crosses $b \Leftrightarrow Dim(I(a) \cap I(b)) < max(Dim(I(a)), Dim(I(b))) \wedge a \cap b \neq a \wedge a \cap b \neq b \wedge a \cap b \neq \emptyset$

Overlaps

- a overlaps $b \Leftrightarrow Dim(I(a)) = Dim(I(b)) = Dim(I(a) \cap I(b)) \wedge a \cap b \neq a \wedge a \cap b \neq b$



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

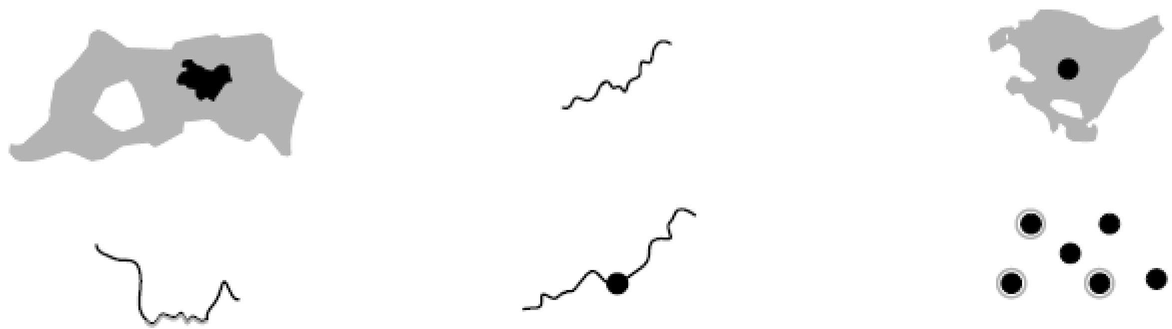
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Topological Predicates (4)

Contains/Within

- a contains $b \Leftrightarrow I(a) \cap I(b) \neq \varnothing \wedge a \cap b = b \Leftrightarrow b$ within a



Disjoint/Intersects

- a disjoint $b \Leftrightarrow a \cap b = \varnothing \Leftrightarrow \neg a$ intersects b

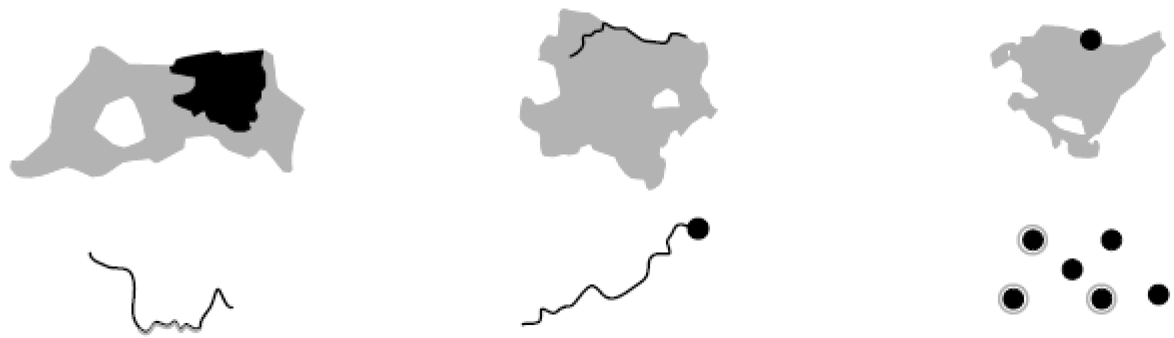
Equals

- a equals $b \Leftrightarrow a \cap b = a \wedge a \cap b = b \Leftrightarrow (a - b) \cup (b - a) = \varnothing$

Topological Predicates (5)

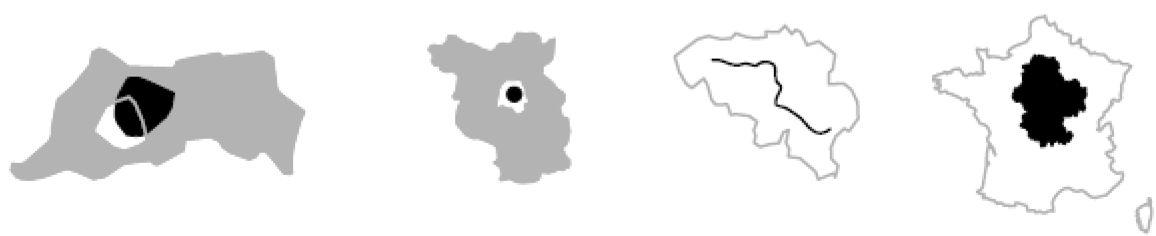
Covers

- a covers $b \Leftrightarrow a \cap b = b \Leftrightarrow b - a = \varnothing$



Encloses/Surrounded

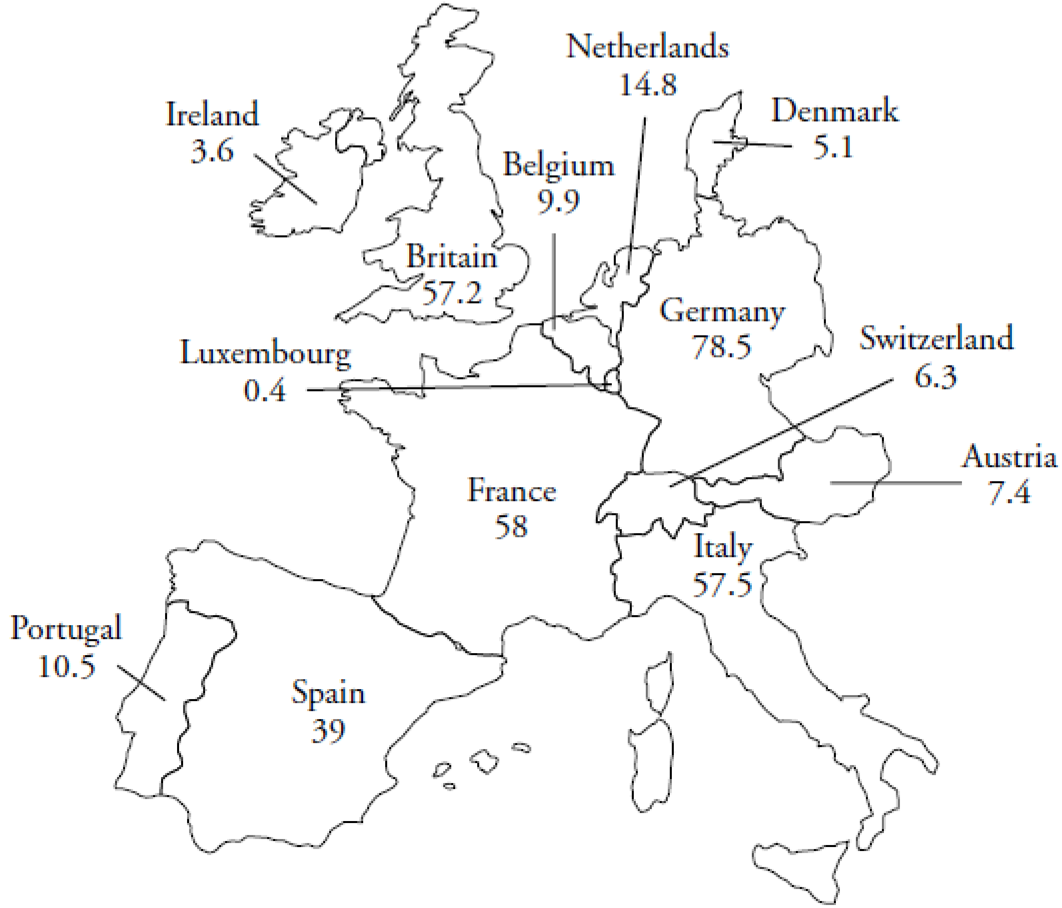
- Definition is quite involved, depends of whether a is a (set of) line(s) or a (set of) surface(s)



Spatial Objects

Country

name
population

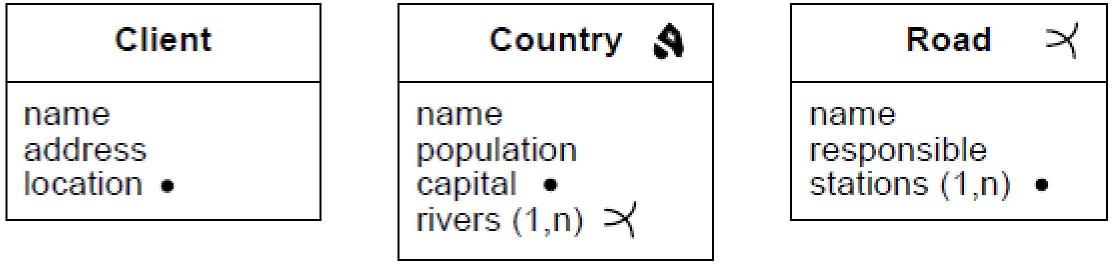


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Attributes



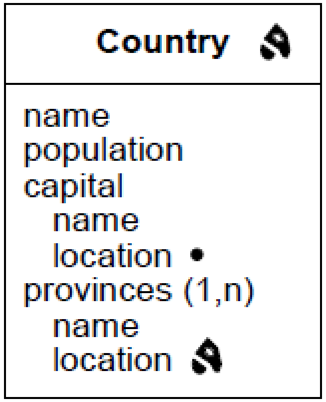
- Both non-spatial and spatial object types can have spatial attributes
- Domain of spatial attribute: a spatial type (point, line, surface, . . .)
- Spatial attributes can be multi-valued
- A spatial attribute of a spatial object type **may** induce a topological constraint
  - The capital of a country is located within the geometry of its country
- This is not necessarily the case
  - A given country will keep **either** the full geometry of the rivers flowing through it **or** these geometries will be projected to the section flowing through the country
  - This depends on application semantics
- The conceptual schema must explicitly state these topological constraints

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Complex Attributes



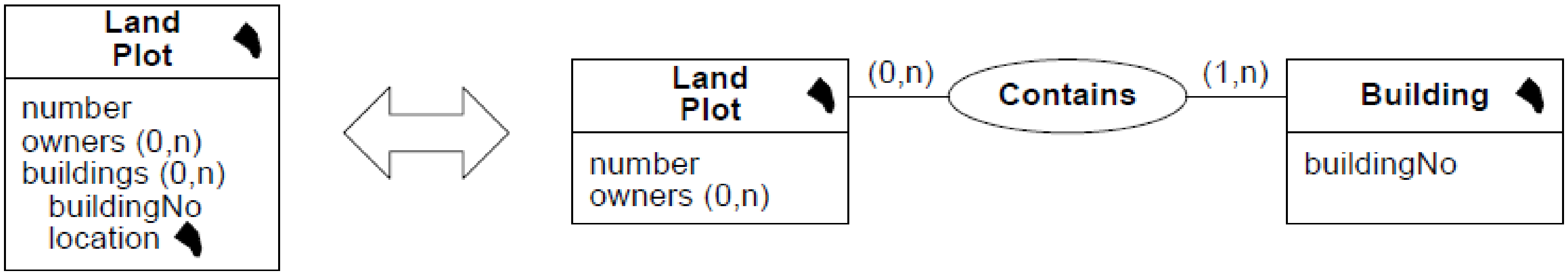
- Spatial attributes can be a component of a complex and/or multivalued attribute
- It is usual to keep both thematic (alphanumeric) and location data for attributes, e.g., capital
- This will allow to print both the name and the location of capitals/rivers/roads/... in a map
- However, in real maps the toponyms (names of objects appearing in a map) have also a location
  - There are precise cartographic rules for placing them, this is a semi-automatic process

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Objects vs. Spatial Attributes



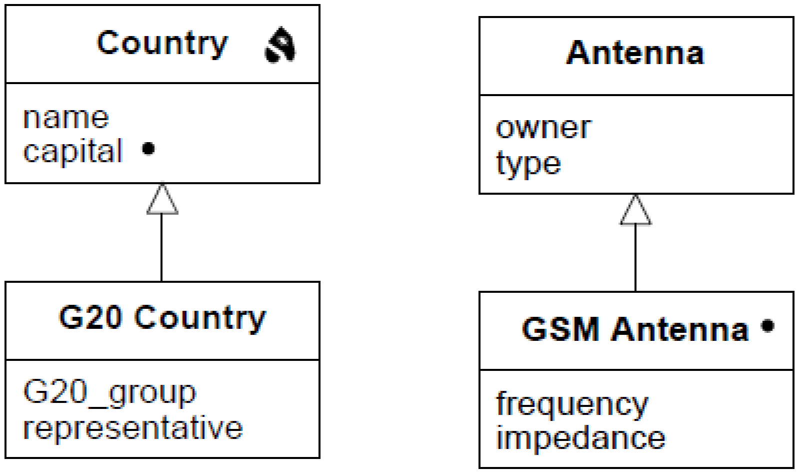
- Representing a concept as a spatial object or as a spatial attribute depends on the application
- Determined by the relative importance of the concept
- Has implications in the way of accessing the instances of the concept (e.g. the buildings)
  - As spatial objects: the application can access a building one by one
  - As spatial attributes: the access to a building must be made through the land plot containing it

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

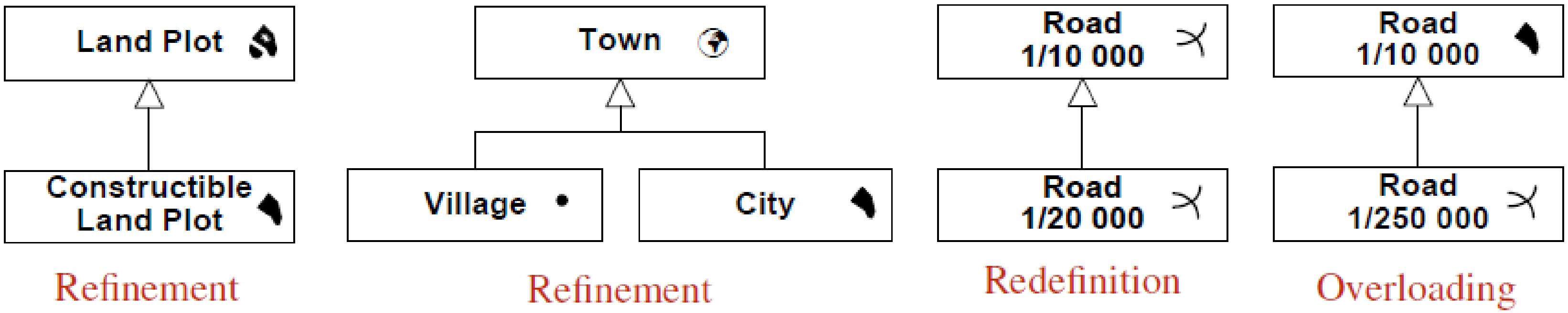
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Generalization: Inheriting Spatiality



- Spatiality is inherited through generalization
  - As any other feature: attributes, methods, relationships, integrity constraints, etc.
- Based on the well-known **susbtitutability principle** in OOP
- For simple inheritance it is not necessary to restate the geometry in the subtype (but see later)
- As usual, spatiality can be added to a subtype
  - Only instances of the subtype will have associated spatiality

Generalization: Refining/Redefining Spatiality



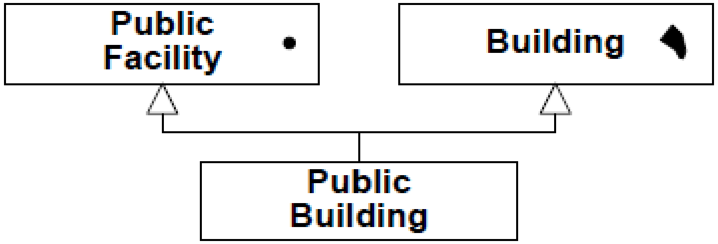
- Arise when a spatiality is restated in a subtype
- **Refinement**: restricts the inherited property, value remains the same in the supertype and the subtype
- In redefinition and overloading an instance of the subtype has both a locally defined spatiality and the inherited one
- **Redefinition**: Keeps substitutability wrt typing, allows dynamic binding
- **Overloading**: Relaxes substitutability, inhibiting dynamic binding

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

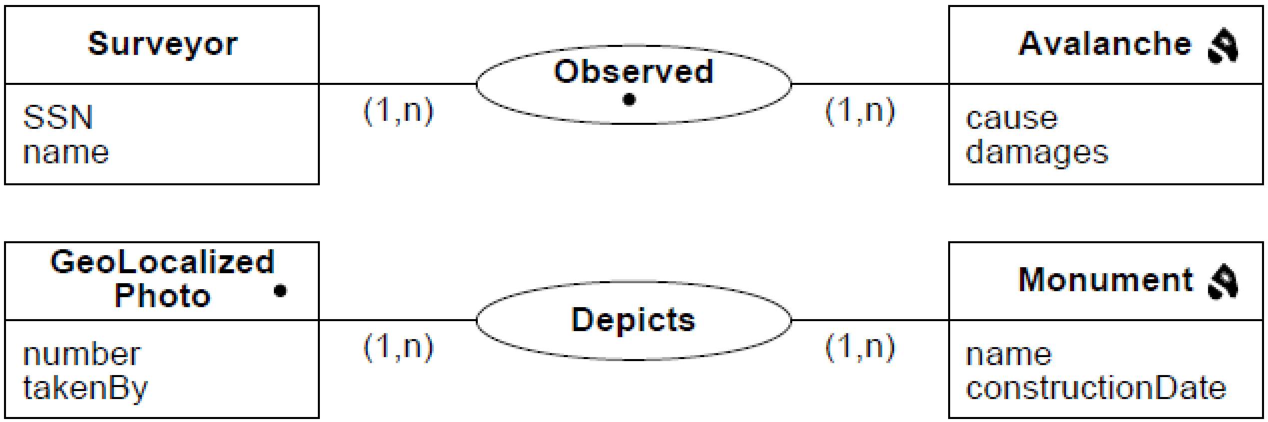
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Generalization: Multiple Inheritance



- Spatiality is inherited from several supertypes
  - Similar situation may arise with any kind of attribute
- There is ambiguity when referring to the spatiality of the subtype
- Several policies have been proposed for solving this issue in the OO community
- Most general policy
  - All inherited properties are available in the subtype, user must disambiguate in queries
  - `PublicFacility.geometry` vs `Building.geometry` for an instance of `PublicBuilding`

Spatial/Non-Spatial Relationships



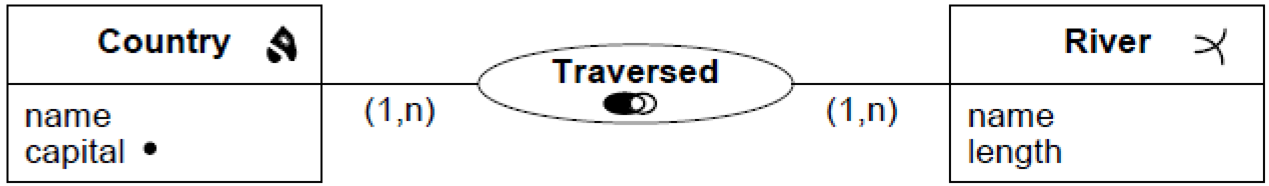
- Spatiality can also be defined for relationship types
- Spatiality of relationship types is orthogonal to the fact that linked object types are spatial
- If a spatial relationship type relates spatial type(s), spatial constraints **may** restrict the geometries

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Topological Relationships



- Specified on a relationship type that links at least two spatial types
- Constrain the spatiality of the instances of the related types
- Many topological constraints can be defined using the 9IM
  - 5 between complex points
  - 8 between simple regions
  - 18 between simple regions with holes
  - 33 between complex regions
  - 43 between a complex line and a complex region
  - . . .
- Conceptual model depicts only the most general ones

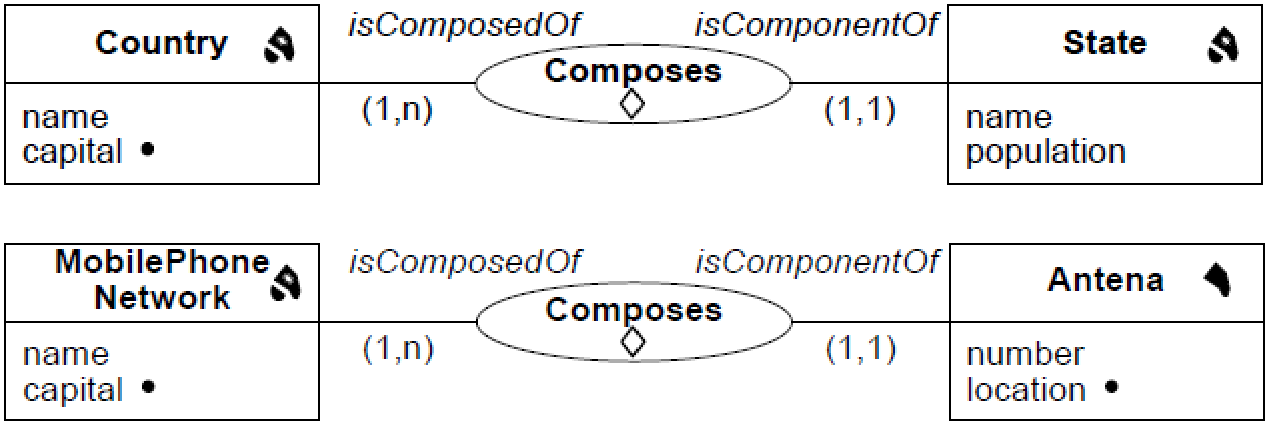
Topological Relationship	Icon
TopoGeneric	
TopoDisjoint	
TopoOverlap	
TopoWithin	
TopoTouch	
TopoCross	
TopoEqual	

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Aggregation



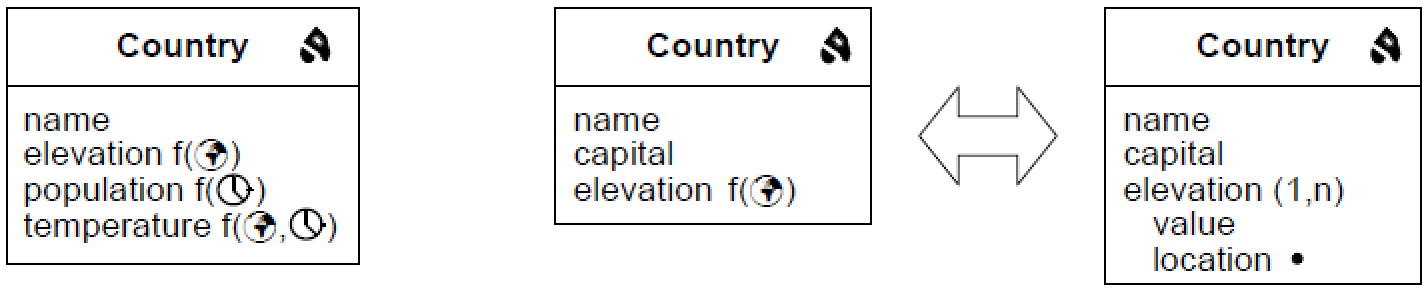
- Traditional aggregation relationship can link spatial types
- Usually, aggregation has exclusive semantics (stated by cardinalities in the component role)
- Usually, the spatiality of the aggregation is **partitioned** into the spatiality of the components
- It is not the case for the second example, where the spatiality of `Antena` corresponds to its coverage
  - The same location can be covered by several antennas
  - Spatiality of the aggregation is the **spatial union** of the spatiality of the antennas

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

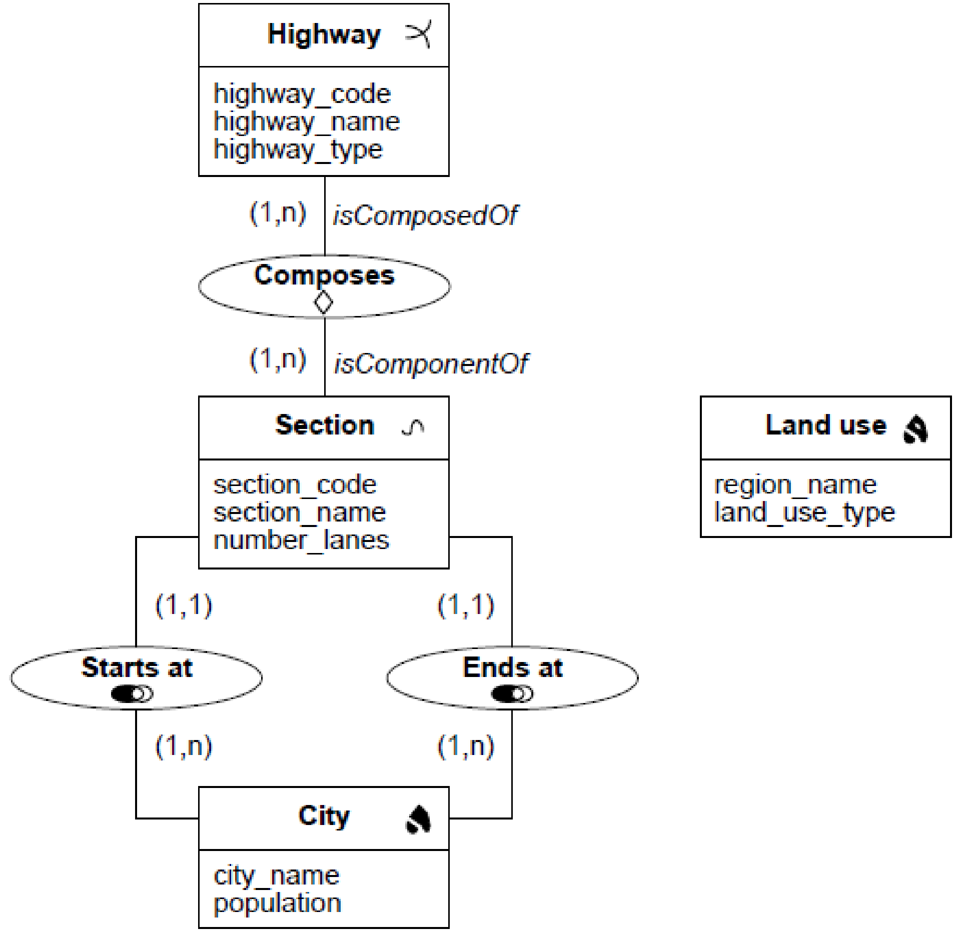
Space- and Time-Varying Attributes



- Also referred to as continuous fields
- Allow to represent phenonemon that change in space and/or in time
  - `elevation`: to each point in space is associated a real number
  - `population`: to each point in time is associated an integer number
  - `temperature`: to each point in space is associated a real number, this value evolves over time
- At the **conceptual level**, can be represented as continuous function
  - Operators for manipulating fields can be defined at this level
- At the **logical level** can be implemented in several ways
  - **Raster**: Discretize the space into regular cells, assign a value to each cell
  - **TIN**: Keep values at particular locations, interpolation used for calculating value at any point

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Conceptual Schema: Example



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Topics

- Introduction
- Georeferences and Coordinate Systems
- Conceptual Modeling for Spatial Databases
- **Logical Modeling for Spatial Databases**
- SQL/MM
- Representative Systems
- Summary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Representation Models

- Representation of infinite point sets of the Euclidean space in a computer
- Two alternative representations
- **Object-based models** (Vector)
  - Describes the spatial extent of relevant objects with a set of points
  - Uses points, lines, and surfaces for describing spatiality
  - Choice of geometric types is arbitrary, varies across systems
- **Field-based models** (Raster)
  - Each point in space is associated with one/several attribute values, defined as continuous functions
  - Examples: altitude, temperature, precipitation, polution, etc.

Belgium Map: Vector

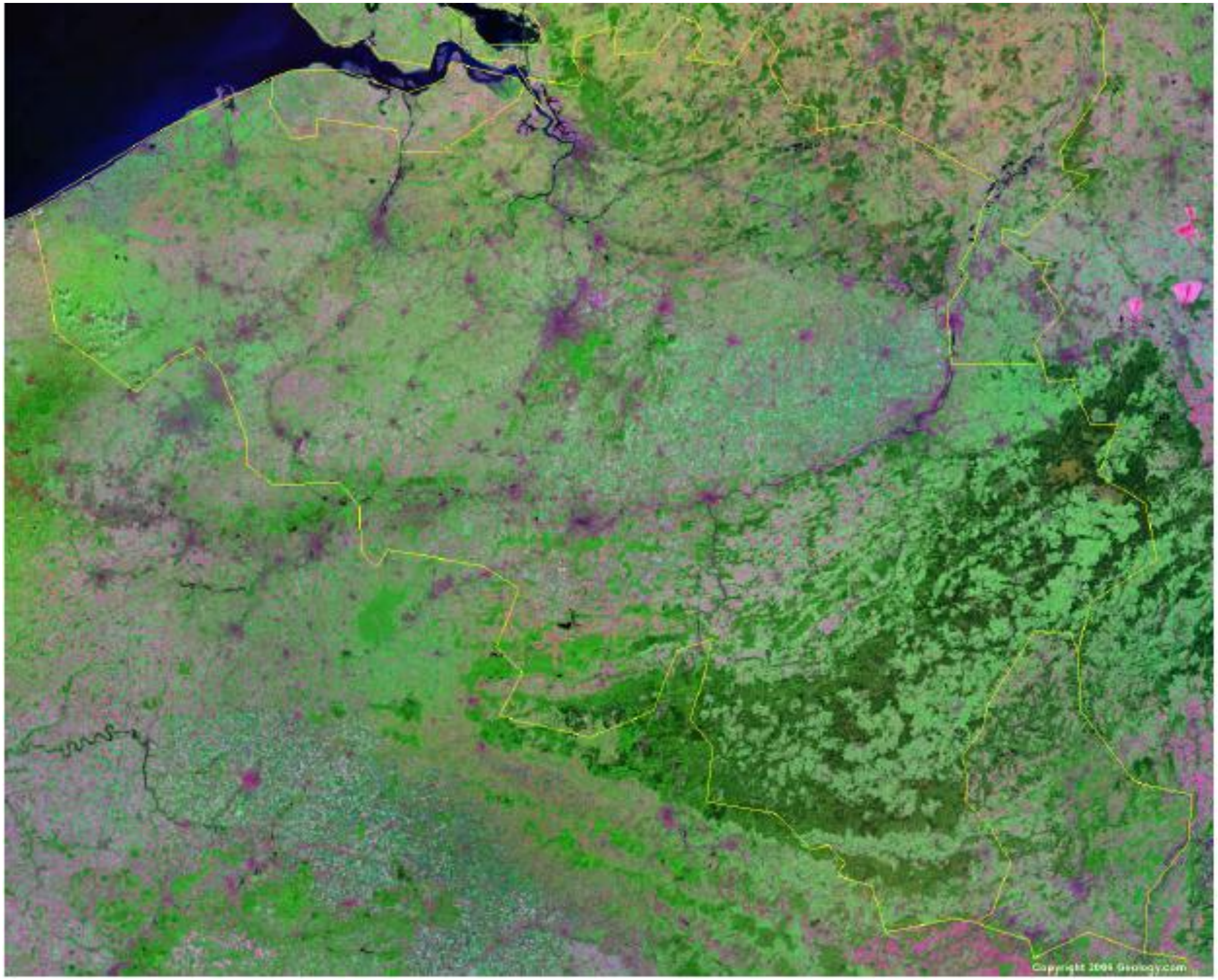
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Belgium Map: Raster

Belgium Map: Satellite

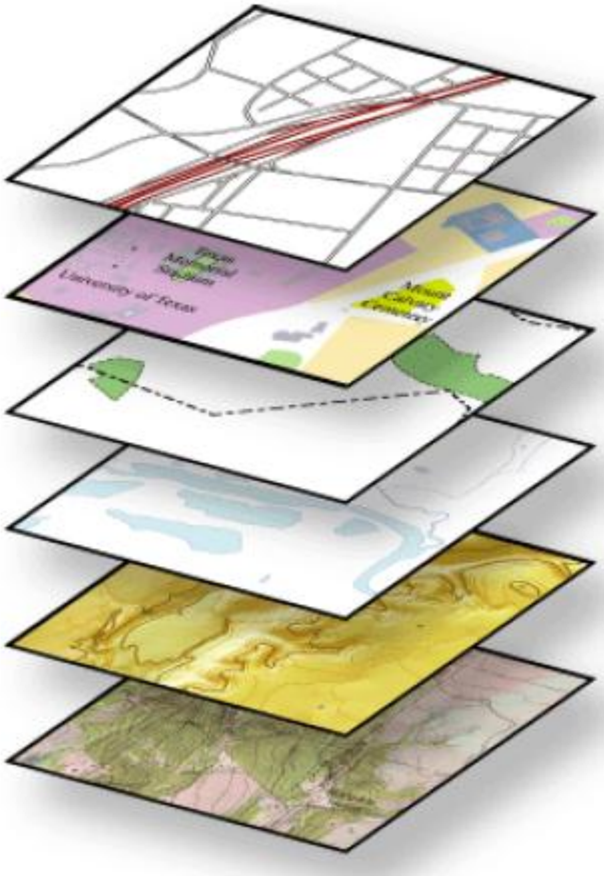


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

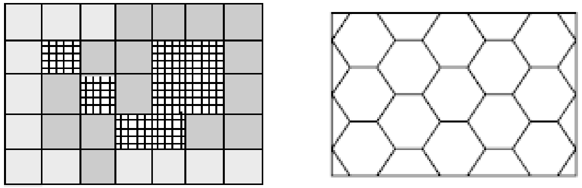
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oraganizing Spatial Data: Layers

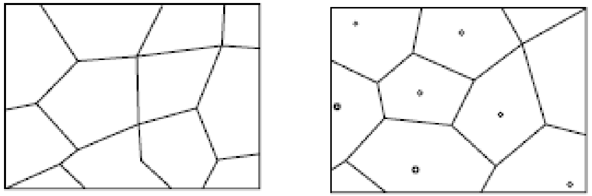


Raster Model: Tesselation

- Decomposition of the plane into polygonal units
- May be **regular** or **irregular**, depending on whether the polygonal units are of equal size

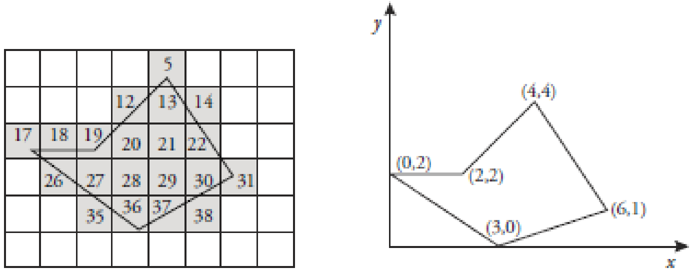


Regular



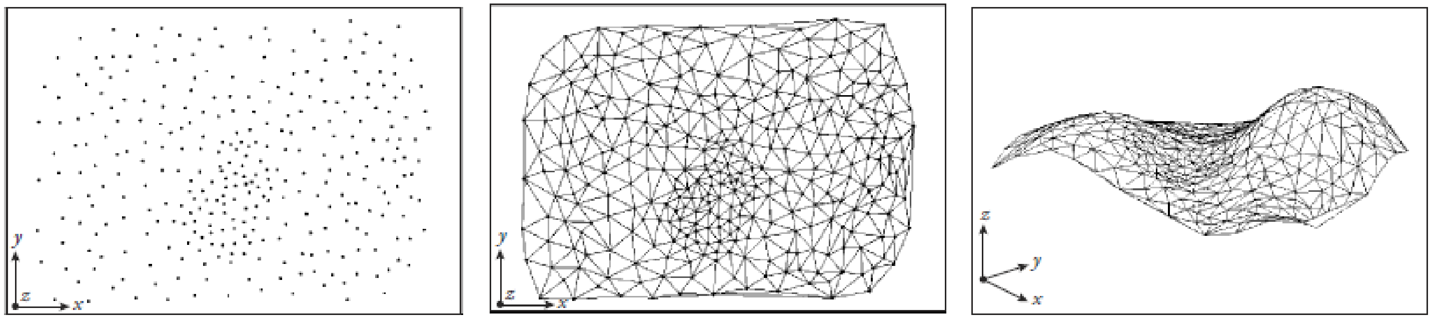
Irregular

- Regular tesselation used for remote sensing data (e.g., satellite images)
- Irregular tesselation used for zoning in social, demographic or economic data
- A spatial object is represented by the smallest subset of pixels that contains it



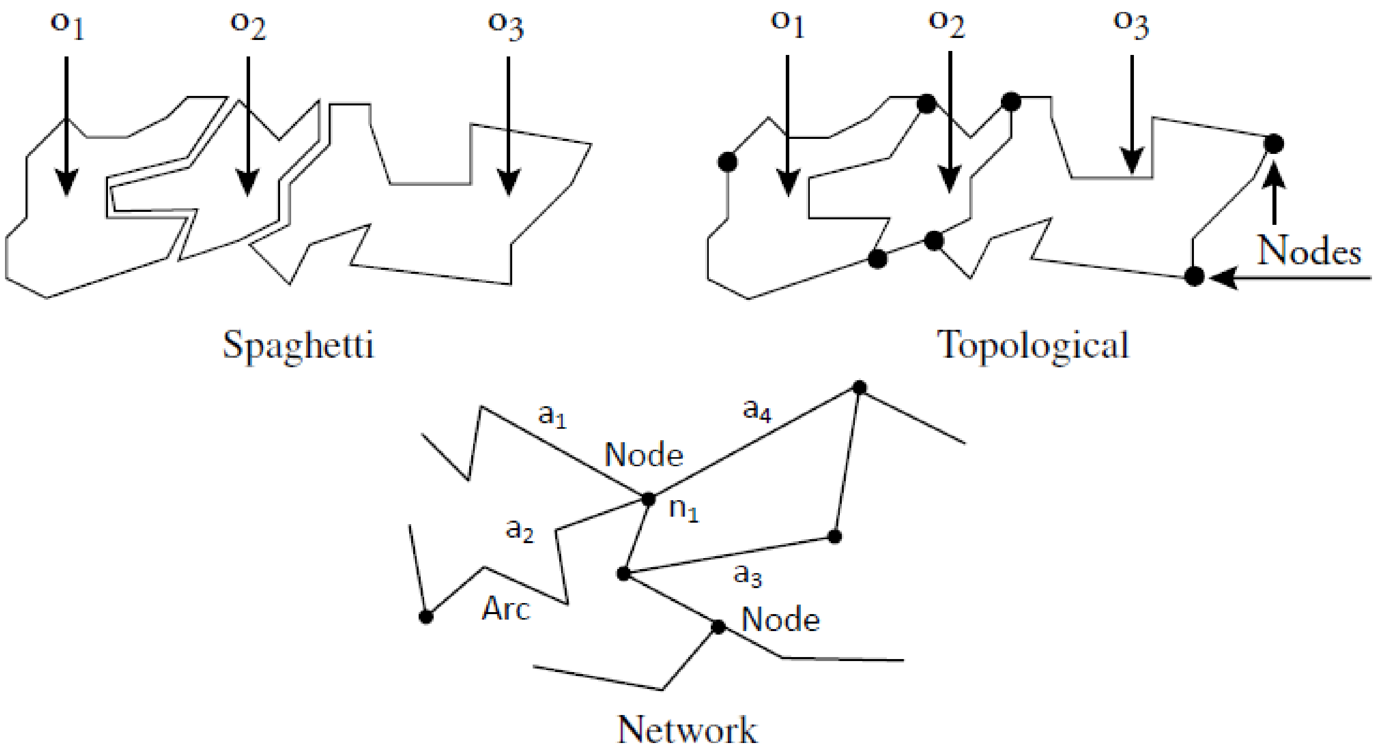
Digital Elevation Models (DEMs)

- Provide a digital (finite) representation of an abstract modeling of space
- DEMs are useful to represent natural phenomemon that is a continuous function of the 2D space
  - temperature, presure, moisture, or slope
- Based on a finite collection of sample values, values at other points are obtained by **interpolation**
- **Triangulated Irregular Networks** (TINs) are based on a triangular partition of the 2D space



- No assumption is made on the distribution and location of the vertices of the triangles
- Elevation value is recorded at each vertex
- Value at any other point P inferred by linear interpolation of the 3 vertices of the triangle containing P

Representing the Geometry of a Collection of Objects



- Three comonly used representations: Spaghetti, Network, Topological
- Mainly differ in the expression of topological relationships among the component objects

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spaghetti Model

- Geometry of any object described independently of other objects
- No topology is stored in the model, all topological relationships must be computed on demand
- Implies representation redundancy
  - E.g., boundary of adjacent regions represented twice
- Enables heterogeneous representations mixing points, polylines and regions without restrictions
  - E.g., polylines may intersect without the intersection points stored explicitly in the database
- Advantages
  - Simplicity
  - Provides the end use with easy input of new objects into the collection
- Drawbacks
  - Lack of explicity information about topological relationships among spatial objects
  - No sharing of information $\Rightarrow$ redundancy, problem with large data sets, inconsistency

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Network Model

- Destined for network (graph)-based applications
  - transportation services, utility management (electricity, telephone, . . .)
- Topological relationships among points and polylines are stored
- **Nodes**: Distinguished point that connects a list of arcs
- **Arcs**: Polyline that starts at a node and ends at a node
- Nodes allow efficent line connectivity tests a network computations (e.g., shortest paths)
- Two types of points: **regular points** and **nodes**
- Depending on the implementation, the network is planar or nonplanar
- **Planar network**: each edge intersection is recorded as a node, even if does not correspond to a real-world entity
- **Nonplanar network**: edges may cross withoug producing an intersection
  - Examples include ground transportation with tunnels and passes

Topological Model

- Similar to the network model, except that the network is plannar
- Induces a planar subdivision into adjacent polygons, some of which may not correspond to actual geographic objects
- **Node**: represented by a point and the (possibly empty) list of arcs starting/ending at it
  - **Isolated point**: identifies location of point features such as towers, point of interest, . . .
- **Arc**: features its ending points, list of vertices and two polygons having the arc as common boundary
- **Polygon**: represented by a list of arcs, each arc being shared with a neighbor polygons
- **Region**: represented by one or more adjacent polygons
- No redundacy: each point/line is stored only once
- **Advantages**: Efficient computation of topological queries, update consistency
- **Drawbacks**: Some database objects have no semantics in real-world, complexity of the structure may slow down some operations

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

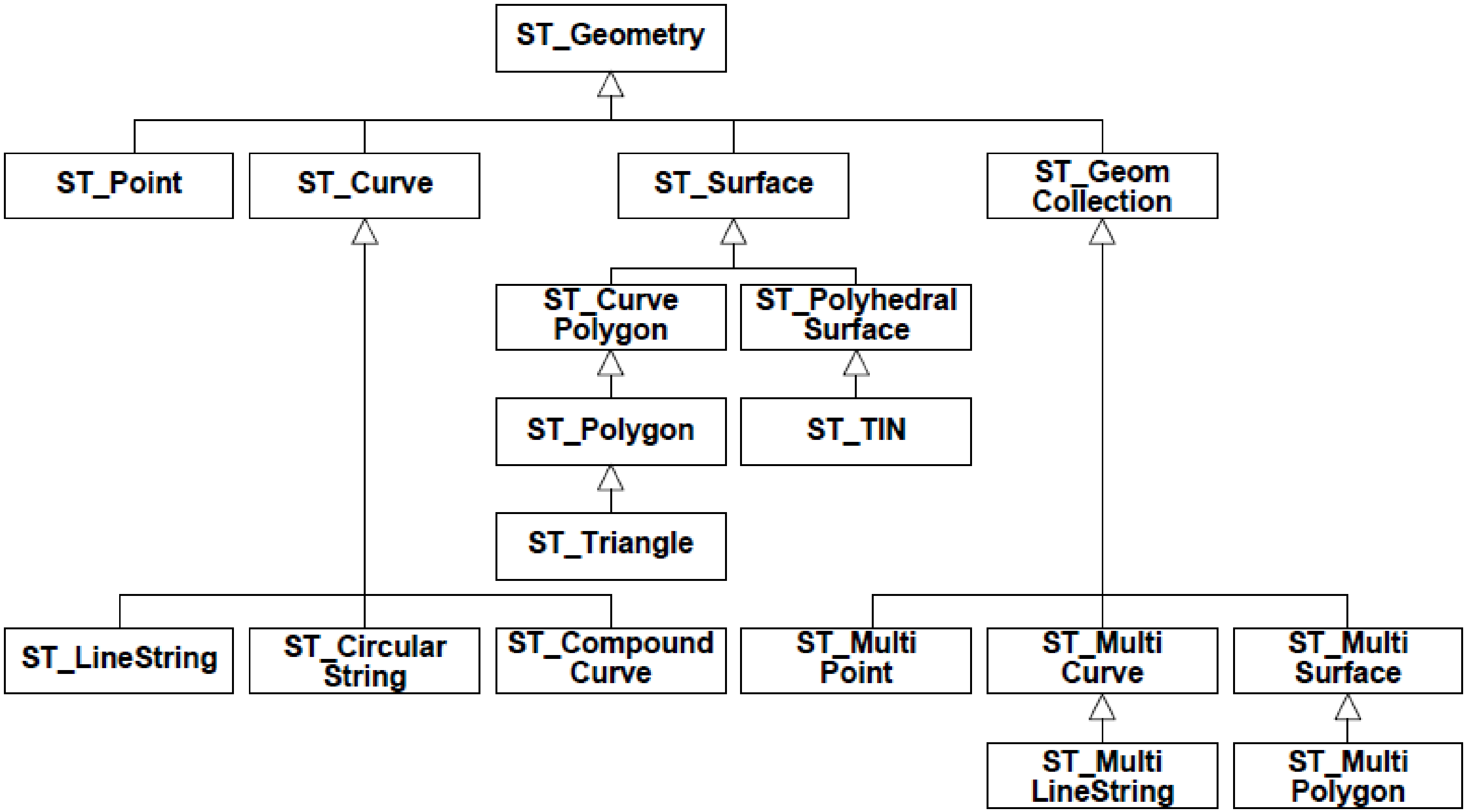
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Topics

- Introduction
- Georeferences and Coordinate Systems
- Conceptual Modeling for Spatial Databases
- Logical Modeling for Spatial Databases
- ➥ **SQL/MM**
- Representative Systems
- Summary

SQL/MM Spatial: Geometry Type Hierarchy



- ST_Geometry, ST_Curve, and ST_Surface are not instantiable types

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

ST_Geometry

- Represent 0D, 1D, and 2D geometries that exist in 2D ($\mathbb{R}^2$), 3D ($\mathbb{R}^3$) or 4D coordinate space ($\mathbb{R}^4$)
- Geometries in $\mathbb{R}^2$ have points with (x, y) coordinate values
- Geometries in $\mathbb{R}^3$ have points with either (x, y, z) or (x, y, m) coordinate values
- Geometries in $\mathbb{R}^4$ have points with (x, y, z, m) coordinate values
- The z coordinate of a point typically represent altitude
- The m coordinate of a point representing arbitrary measurement: key to supporting linear networking applications such as street routing, transportation, pipeline, . . .
- Geometry values are topologically closed (they include their boundary)
- All locations in a geometry are in the same spatial reference system (SRS)
- Geometric calculations are done in the SRS of the first geometry in the parameter list of a routine
- If a routine returns a geometry or measurement (e.g., length or area), the value is in the SRS of the first geometry in the parameter list

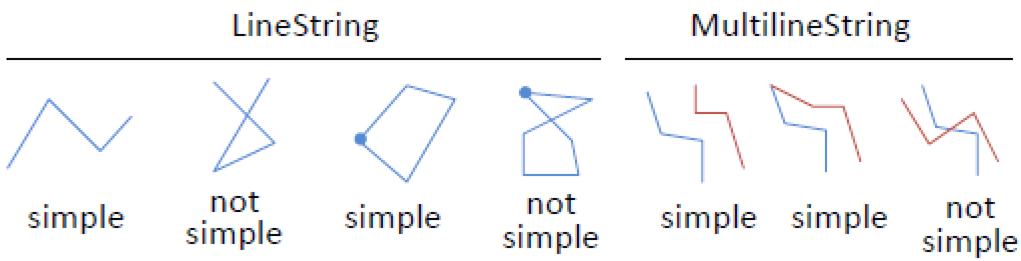
Methods on ST_Geometry[a]: Metadata (1)

- ST_Dimension: returns the dimension of a geometry
- ST_CoordDim: returns the coordinate dimension of a geometry
- ST_GeometryType: returns the type of the geometry as a CHARACTER VARYING value
- ST_SRID: observes and mutates the spatial reference system identifier of a geometry
- ST_Transform: returns the geometry in the specified spatial reference system
- ST_IsEmpty: tests if a geometry corresponds to the empty set

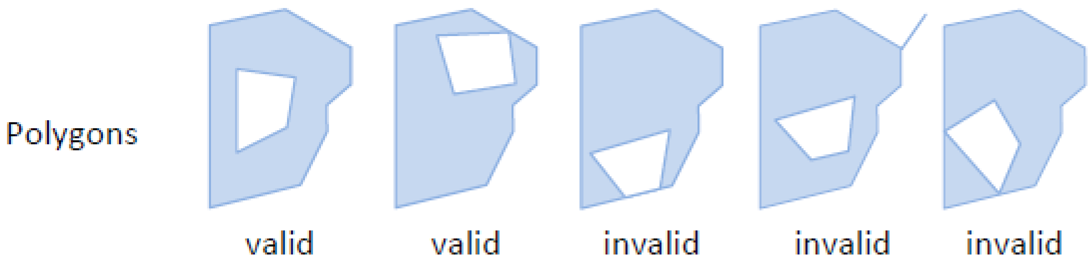
[a] 3D versions of some of these methods exists

Methods on ST_Geometry: Metadata (2)

- ST_IsSimple: tests if a geometry has no anomalous geometric points



- ST_IsValid: tests if a geometry is well formed

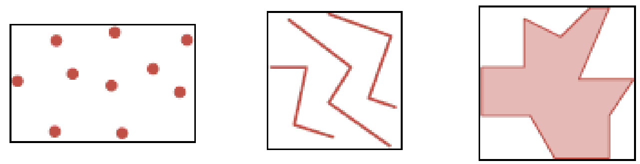


- ST_Is3D: tests whether a geometry has *z* coordinates
- ST_IsMeasured: tests whether a geometry has *m* coordinate values

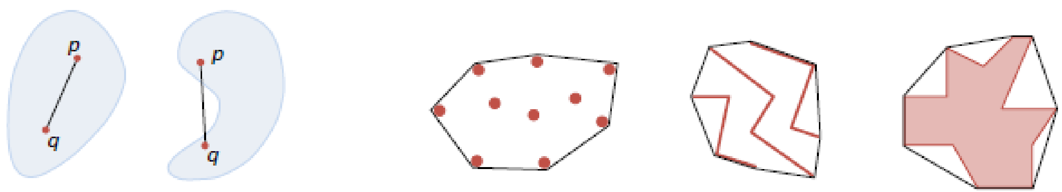
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Methods on ST_Geometry: Spatial Analysis (1)

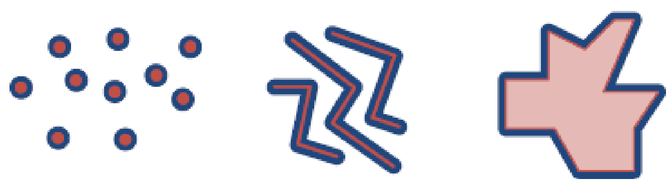
- ST_Boundary: returns the boundary of a geometry
- ST_Envelope: returns the bounding rectangle of a geometry



- ST_ConvexHull: returns the convex hull of a geometry



- ST_Buffer: returns the geometry that represents all points whose distance from any point of a geometry is less than or equal to a specified distance



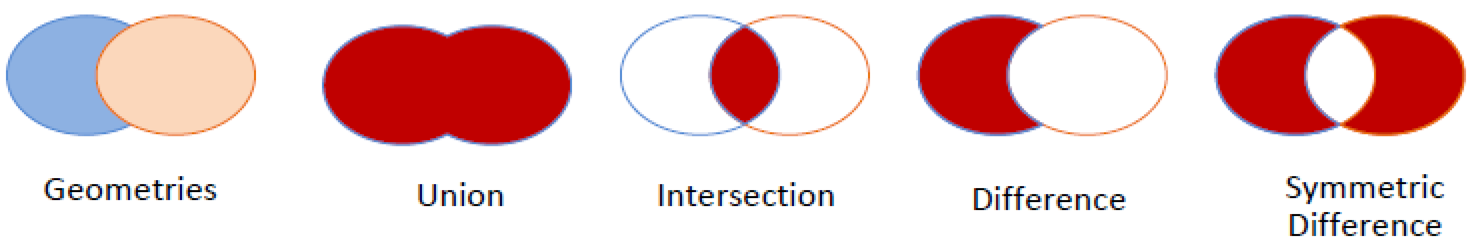
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Methods on ST_Geometry: Spatial Analysis (2)

- ST_Union: returns the geometry that represents the point set union of two geometries
- ST_Intersection: returns the geometry that represents the point set intersection of two geometries
- ST_Difference: returns the geometry that represents the point set difference of two geometries
- ST_SymDifference: returns the geometry that represents the point set symmetric difference of two geometries



- ST_Distance: returns the distance between two geometries

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Methods and Functions on ST_Geometry: Input/Output

- ST_WKTToSQL: returns the geometry for the specified well-known text representation
- ST_AsText: returns the well-known text representation for the specified geometry
- ST_WKBToSQL: returns the geometry for the specified well-known binary representation
- ST_AsBinary: returns the well-known binary representation for the specified geometry
- ST_GMLToSQL: returns the geometry for the specified GML representation
- ST_AsGML: returns the GML representation for the specified geometry
- ST_GeomFromText: returns a geometry, which is transformed from a CHARACTER LARGE OBJECT value that represents its well-known text representation
- ST_GeomFromWKB: returns a geometry, which is transformed from a BINARY LARGE OBJECT value that represents its well-known binary representation
- ST_GeomFromGML: returns a geometry, which is transformed from a CHARACTER LARGE OBJECT value that represents its GML representation

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Boundary, Interior, Exterior

- **Boundary** of a geometry: set of geometries of the next lower dimension
  - ST_Point or ST_MultiPoint value: empty set
  - ST_Curve: start and end ST_Point values if nonclosed, empty set if closed
  - ST_MultiCurve: ST_Point values that are in the boundaries of an odd number of its element ST_Curve values
  - ST_Polygon value: its set of linear rings
  - ST_MultiPolygon value: set of linear rings of its ST_Polygon values
  - Arbitrary collection of geometries whose interiors are disjoint: geometries drawn from the boundaries of the element geometries by application of the mod 2 union rule
  - The domain of geometries considered consists of those values that are topologically closed
- **Interior** of a geometry: points that are left when the boundary points are removed
- **Exterior** of a geometry: points not in the interior or boundary

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

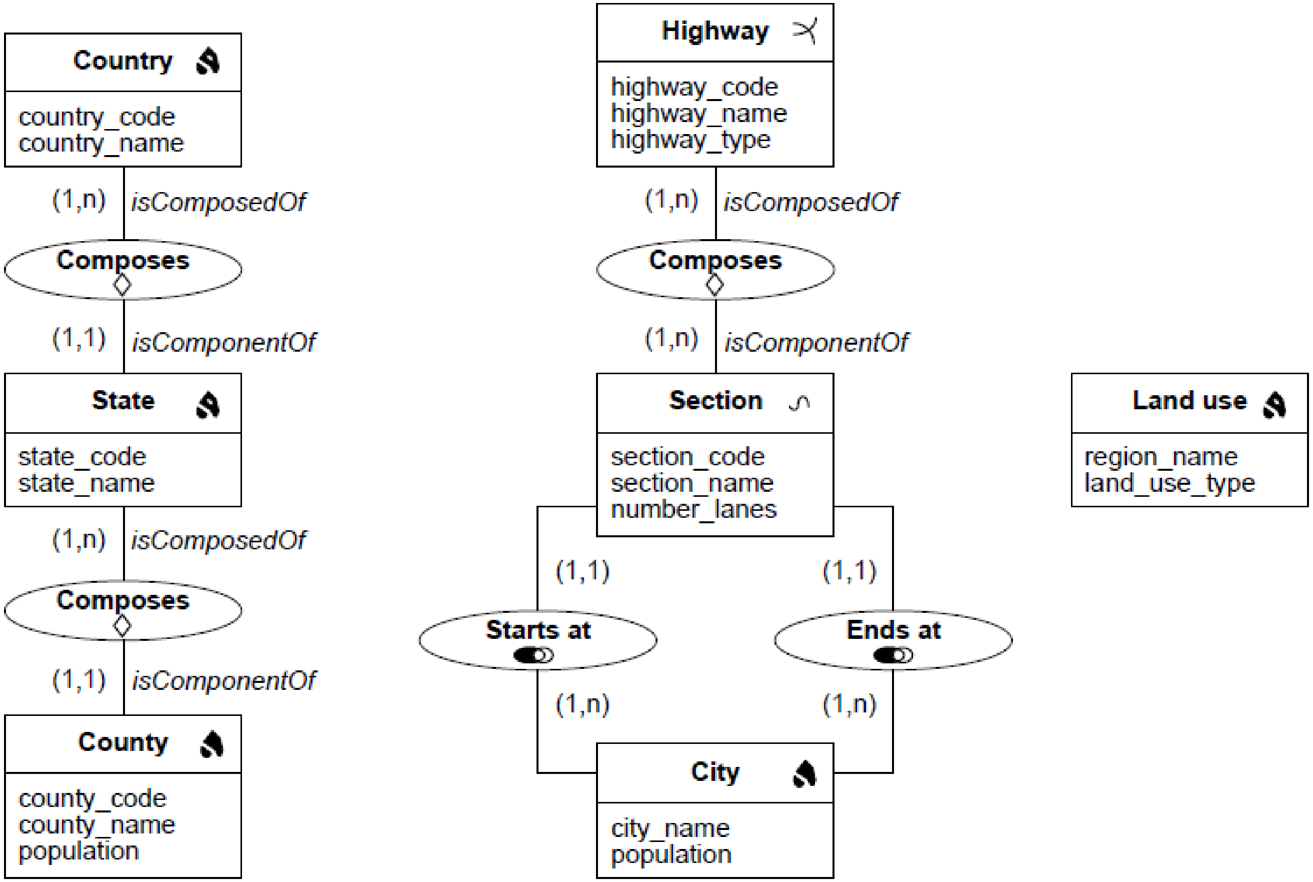
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Relationships

- `ST_Equals`: tests if a geometry is spatially equal to another geometry
- `ST_Disjoint`: tests if a geometry is spatially disjoint from another geometry
- `ST_Intersects`: tests if a geometry spatially intersects another geometry
- `ST_Touches`: tests if a geometry spatially touches another geometry
- `ST_Crosses`: tests if a geometry spatially crosses another geometry
- `ST_Within`: tests if a geometry is spatially within another geometry
- `ST_Contains`: tests if a geometry spatially contains another geometry
- `ST_Overlaps`: tests if a geometry spatially overlaps another geometry
- `ST_Relate`: tests if a geometry is spatially related to another geometry by testing for intersections between their interior, boundary and exterior as specified by the intersection matrix
  - `a.ST_Disjoint(b)` $\Leftrightarrow (I(a) \cap I(b) = \emptyset) \wedge (I(a) \cap B(b) = \emptyset) \wedge (B(a) \cap I(b) = \emptyset) \wedge (B(a) \cap B(b) = \emptyset) \Leftrightarrow$ `a.ST_Relate(b,'FF*FF****')`

Conceptual Schema: Example



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference Schemas (1)

```
Create Table Country
(country_code integer,
country_name varchar (30),
geometry ST_MultiPolygon,
Primary Key (country_code))

Create Table State
(state_code integer,
state_name varchar (30),
country_code integer,
geometry ST_MultiPolygon,
Primary Key (state_code),
Foreign Key (country_code) References Country)

Create Table County
(county_code integer
county_name varchar (30),
state_code integer,
population integer,
geometry ST_MultiPolygon,
Primary Key (county_code),
Foreign Key (state_code) References State)
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference Schemas (2)

```
/* Table Highway is NOT spatial */
Create Table Highway
(highway_code integer,
highway_name varchar (4),
highway_type varchar (2),
Primary Key (highway_code))

Create Table HighwaySection
(section_code integer,
section_number integer,
highway_code integer,
Primary Key (section_code,highway_code),
Foreign Key (section_code) References Section,
Foreign Key (highway_code) References Highway)
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference Schemas (3)

```
Create Table Section
(section_code integer,
section_name varchar (4),
number_lanes integer,
city_start varchar (30),
city_end varchar (30),
geometry ST_Line,
Primary Key (section_code),
Foreign Key (city_start) References City,
Foreign Key (city_end) References City)

Create Table City
(city_name varchar (30),
population integer,
geometry ST_MultiPolygon,
Primary Key (city_name))

Create Table LandUse
(region_name varchar (30),
land_use_type varchar (30),
geometry ST_Polygon,
Primary Key (region_name))
```

Reference Queries: Alphanumerical Criteria (1)

- Number of inhabitants in the county of San Francisco

```
select population
from County
where county_name = 'San Francisco'
```

- List of the counties of the State of California

```
select county_name
from County, State
from State.state_code = County.state_code
and state_name = 'California'
```

- Number of inhabitants in the US

```
select sum (c2.population)
from Country c1, State s, County c2
where c1.country_name = 'USA'
and c1.country_code = s.country_code
and s.state_code = c2.state_code
```

Reference Queries: Alphanumerical Criteria (2)

- Number of lanes in the first section of Interstate 99

```
select s.number_lanes
from Highway h1, HighwaySection h2, Section s
where h1.highway_code = h2.highway_code
and h2.section_code = s.section_code
and h1.highway_name = 'I99'
and h2.section_number = 1
```

- Name of all sections that constitute Interstate 99

```
select s.section_name
from Highway h1, HighwaySection h2, Section s
where h1.highway_name = 'I99'
and h1.highway_code = h2.highway_code
and h2.section_code = s.section_code
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference queries: Spatial Criteria (1)

- Counties adjacent to the county of San Francisco in the same state

```
select c1.county_name
from County c1, County c2
where c2.county_name = 'San Francisco'
and c1.state_code = c2.state_code
and ST_Touches(c1.geometry, c2.geometry)
```

- Display of the State of California

```
select ST_Union(c.geometry)
from County c, State s
where s.state_code = c.state_code
and s.state_name = 'California'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference queries: Spatial Criteria (2)

- Counties larger than the largest county in California

```
select c1.county_name
from County c1
where ST_Area(c1.geometry) >
    (select max (ST_Area(c.geometry))
     from County c, State s
     where s.state_code = c.state_code
     and s.state_name = 'California')
```

- Length of Interstate 99

```
select sum (ST_Length(s.geometry))
from Highway h1, HighwaySection h2, Section s
where h1.highway_name = 'I99'
and h1.highway_code = h2.highway_code
and h2.section_code = s.section_code
```

- Shortest path between Santa Fe and Los Angeles
  - Requires the table function ST_ShortestDirPath to determine shortest path in directed graphs

Reference queries: Spatial Criteria (3)

- All highways going through the State of California

```
select distinct h1.highway_name
from State s1, Highway h1, HighwaySection h2, Section s2
where s1.state_name = 'California'
and h1.highway_code = h2.highway_code
and h2.section_code = s2.section_code
and ST_Overlaps(s2.geometry, s1.geometry)
```

- Display of all residential areas in the county of San Jose

```
select ST_Intersection(l.geometry, c.geometry)
from County c, LandUse l
where c.county_name = 'San Jose'
and l.land_use_type = 'residential area'
and ST_Overlaps(l.geometry, c.geometry)
```

- Overlay the map of administrative units and land use

```
select county_name, land_use_type, ST_Intersection(c.geometry, l.geometry)
from County c, LandUse l
where ST_Overlaps(c.geometry, l.geometry)
```

Reference Queries: Interactive queries (1)

- Description of the county pointed to on the screen

```
select county_name, population
from County
where ST_Contains(geometry, @point)
```

- Counties that intersect a given rectangle on the screen

```
select county_name
from County
where ST_Overlaps(geometry, @rectangle)
```

- Part of counties that are within a given rectangle on the screen

```
select ST_Intersection(geometry, @rectangle)
from County
where ST_Overlaps(geometry, @rectangle)
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Reference Queries: Interactive queries (2)

- Description of the section pointed to on the screen

```
select section_name, number_lanes
from Section
where ST_Contains(geometry, @point)
```

- Description of the highway of which a section is pointed to on the screen

```
select h1.highway_name, h1.highway_type
from Highway h1, HighwaySection h2, Section s
where h1.highway_code = h2.highway_code
and h2.section_code = s.section_code
and ST_Contains(s.geometry, @point)
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SQL/MM: Conclusion

- SQL/MM provides a standard way to declare and manipulate geometries
- The last version includes 3D and 4D types
- Several spatial data type organized in a hierarchy with associated methods
- These methods can be combined in SQL queries and programs with standard ones
- We only convered a small part of the standard
- For additional information refer to the document
- However, systems deviate, sometimes considerably from the standard

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Topics

- Introduction
- Georeferences and Coordinate Systems
- Conceptual Modeling for Spatial Databases
- Logical Modeling for Spatial Databases
- SQL/MM
- Representative Systems
  - **Oracle**
- Summary

Oracle Locator

- Included in all editions of the database
- All functions required for standard GIS tools
- All geometric objects
  - Points, lines, polygons
  - 2D, 3D, 4D
- Indexing: quadtrees and rtrees
- Geometric queries
- Proximity search
- Distance calculation
- Multiple projections
- Conversion of projections

Oracle Spatial

- Advanced functions: Option of Oracle Database Enterprise Edition
- = Locator + ...
  - Geometric transformations
  - Spatial aggregations
  - Dynamic segmentation
  - Measures
  - Network modeling
  - Topology
  - Raster
  - Geocoder
  - Spatial Data Mining
  - 3D Types (LIDAR, TINS)
  - Web Services (WFS, CSW, OpenLS)

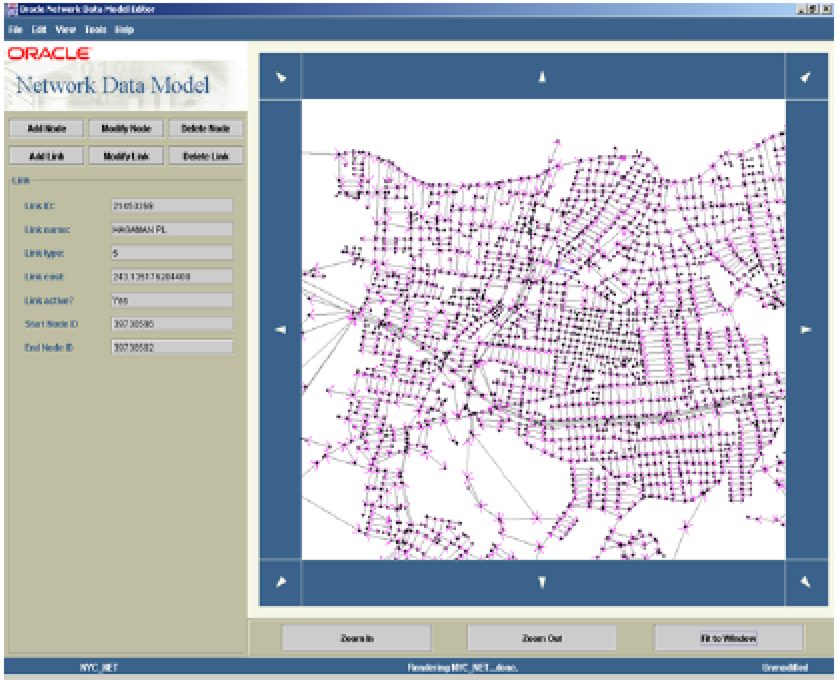
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle Network Model

- A data model for representing networks in the database
- Maintains connectivity
- Attributes at the link and node levels
- Used for network management
  - Transportation, logistics, utilities, location-based services, . . .
- Navigation engine for route calculation



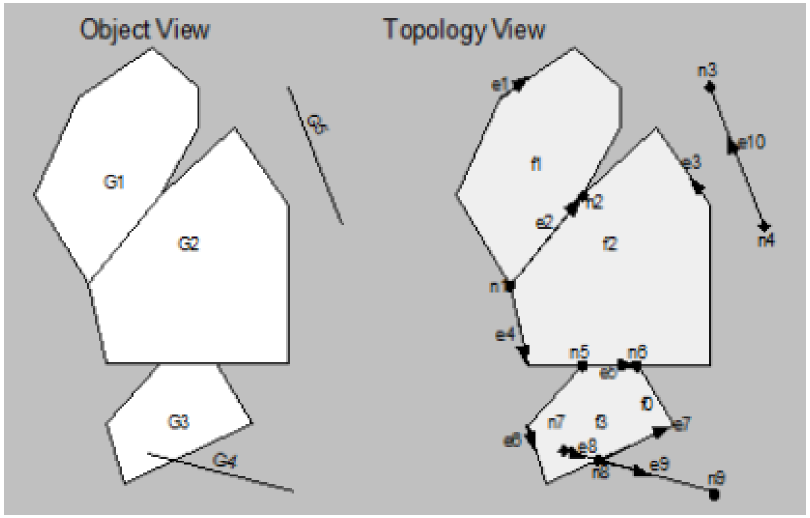
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle Topological Model

- Persistent storage of the topology
  - Nodes, arcs, faces
  - Topological relations
  - Allows advanced consistency checks
- Data model
  - Defining objects (features) through topological primitives
  - New type SDO_TOPO_GEOMETRY
  - Use traditional operators (SDO_RELATE ...)
- Co-existence with traditional spatial data
  - Possibility of combining SDO_GEOMETRY and SDO_TOPO_GEOMETRY



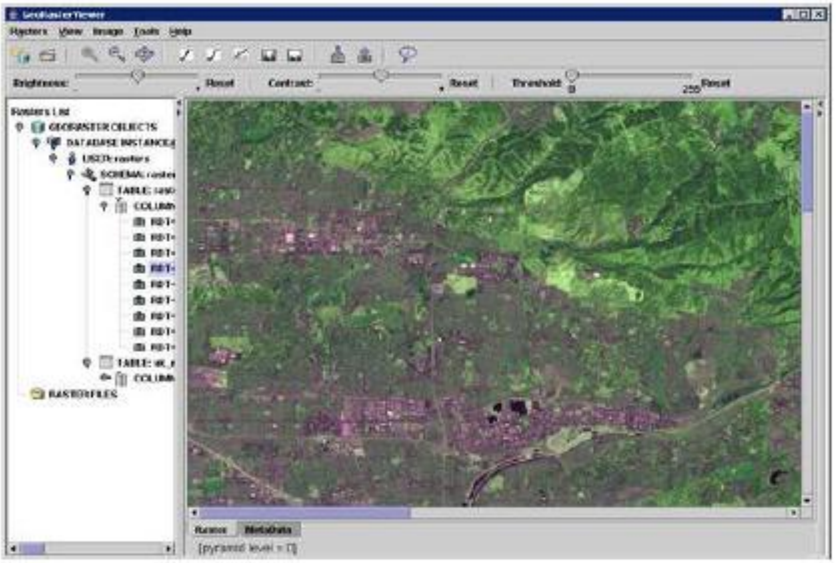
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

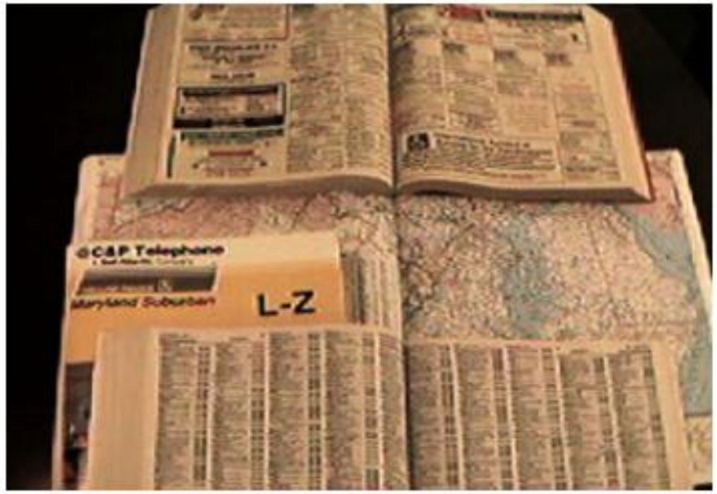
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle Geo Raster

- New data type SDO_GEORASTER
  - Satellite images, remote sensing data
  - Multi-band, multi-layer
  - Metadata in XML
  - Geo-referencing information
- Functionality
  - Open, general purpose raster data model
  - Storage and indexing of raster data
  - Querying and analyzing raster data
  - Delivering GeoRaster to external consumers: Publish as JPEG, GIFF images



Oracle Geocoding



- Generates latitude/longitude (points) from address
- International addressing standardization
- Formatted and unformatted addresses
- Tolerance parameters support fuzzy matching
- Record-level and batch processes
- Data available from NAVTEQ, Tele Atlas

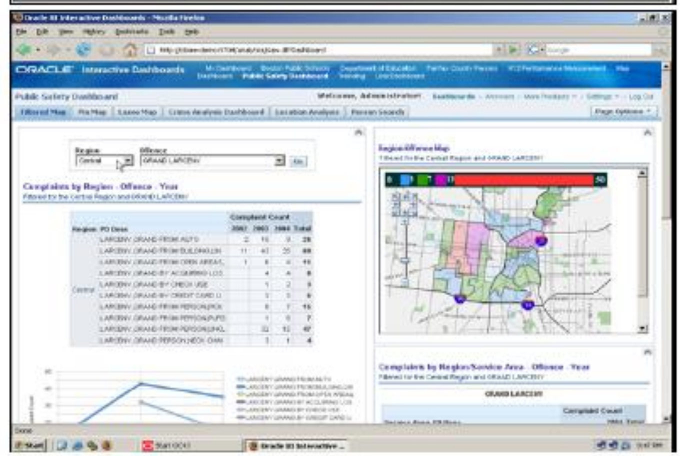
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle MapViewer

- Supplied with all versions of Oracle Application Server
- XML interfaces, Java and Javascript (Ajax)
- Tool for map definition
- Maps described in the database
  - Symbology, visibility, etc.
- Thematic maps
- Formats: PNG, GIF, JPEG, SVG
- OGC WMS compatibility
  - Both server and client



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle: Geometry Type

- Creation of spatial tables

```
CREATE TABLE Cells (
    Cell_id        NUMBER,
    Cell_name      VARCHAR2(32),
    Cell_type      NUMBER,
    Location       SDO_GEOMETRY,
    Covered_area   SDO_GEOMETRY);
```

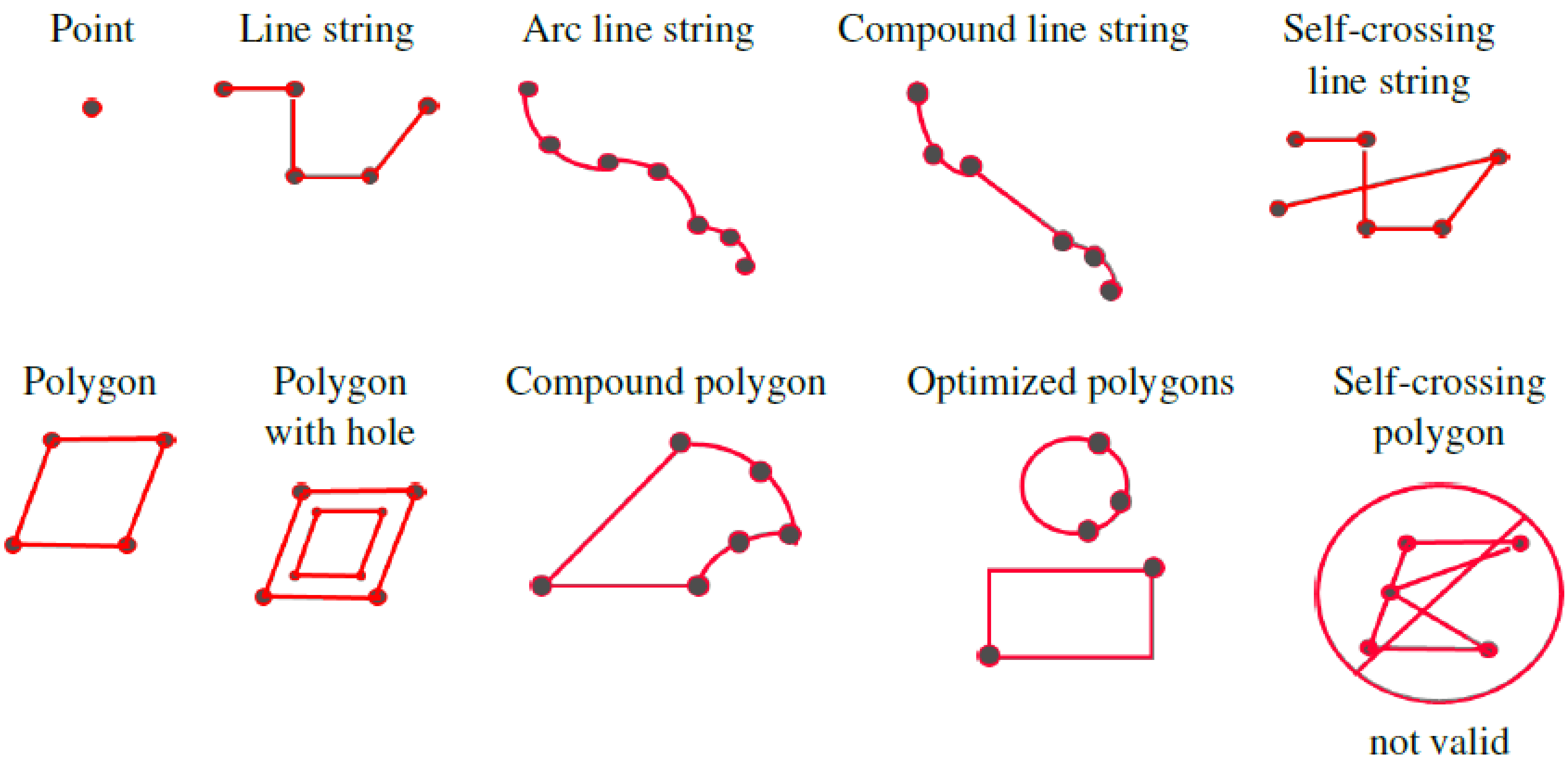
- Use the SDO_GEOMETRY type
- No limit to the number of geometrical columns per table
- The column can contain any type of geometry

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle: Geometrical Primitives



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

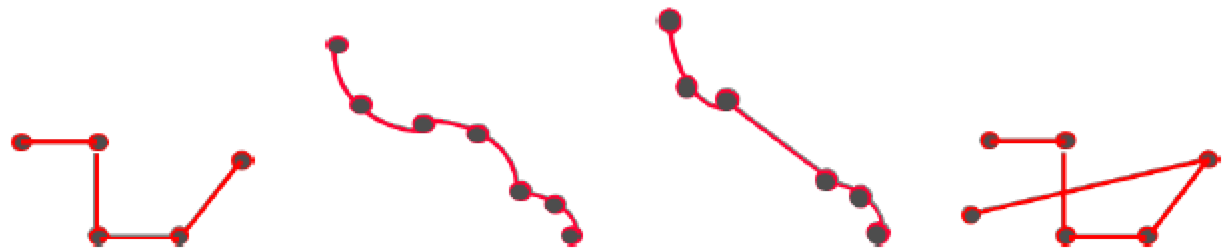
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Geometrical Primitives: Points

- Points (X1, Y1)
- Represent des point objects: buildings, clients, agencies, ...
- 2, 3, or 4 dimensions

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Geometrical Primitives: Lines



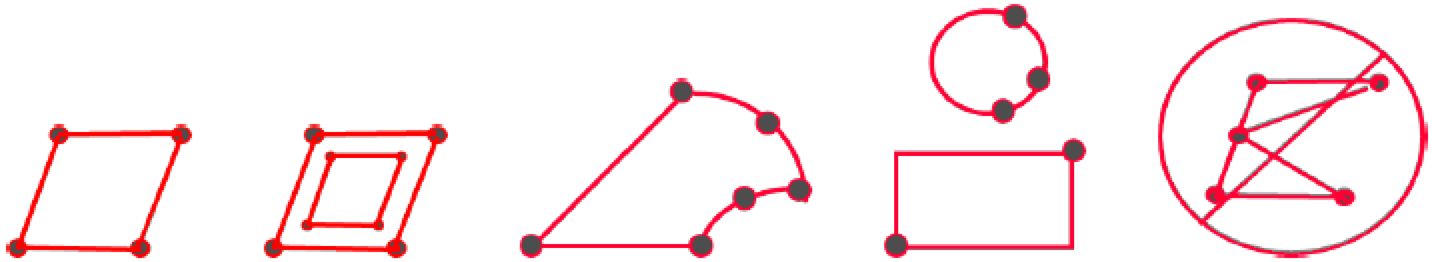
- Lines (X1, Y1, ... Xn, Yn)
- Represent linear objects such as roads, cables, rivers, etc.
- Formed of straight lines or arcs (or their combination)
- A closed line does not delineate surface
- Self-crossing lines allowed

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Geometrical Primitives: Polygones



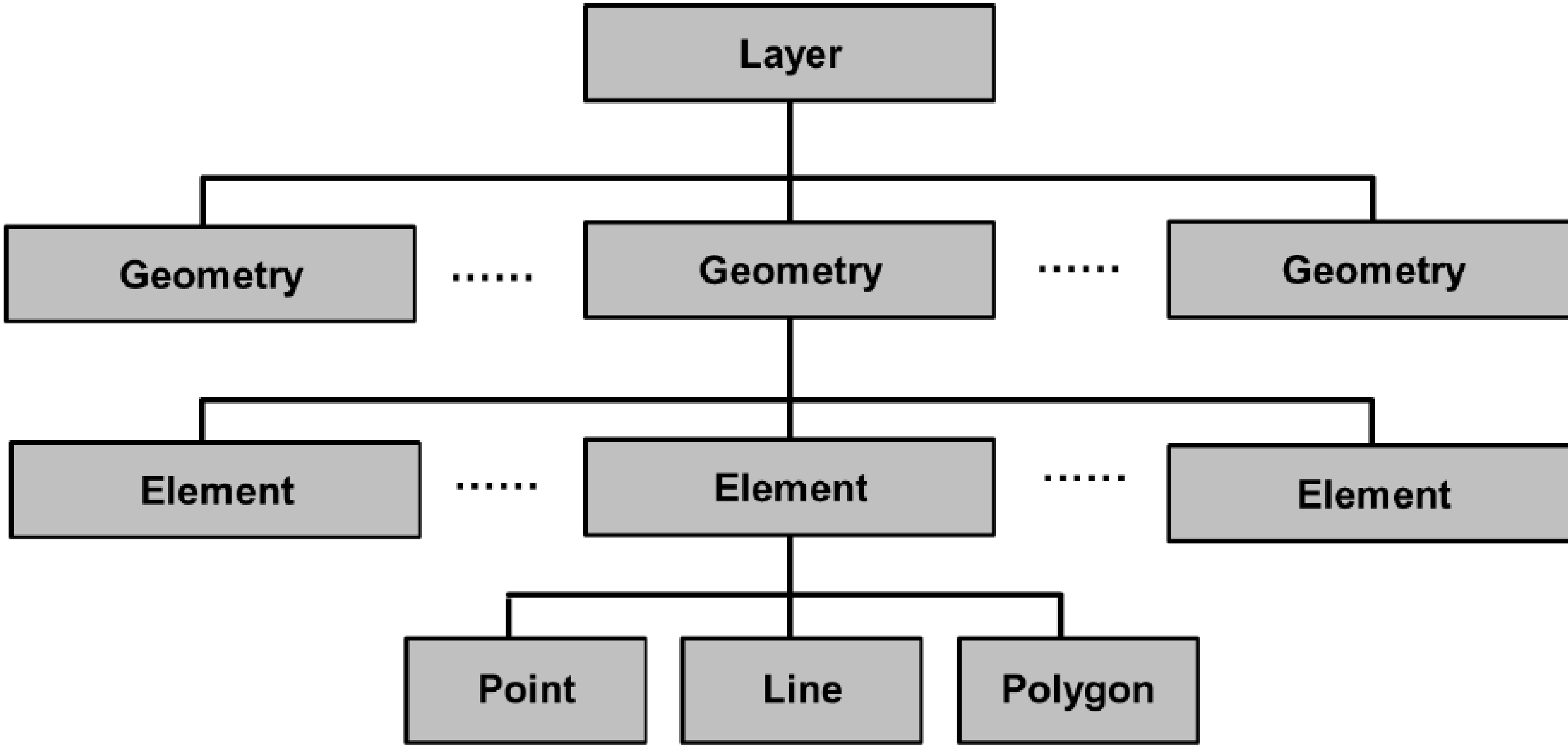
- Polygons (X1, Y1, ... Xn, Yn)
- Represent surface objects: fields, regions, postal codes, etc.
- The contour must be closed (last point = first point)
- The interior can contain one or more holes or voids
- The boundary cannot intersect
- Boundary formed by straight lines or arcs (or their combination)
- Also specific forms: rectangle, circle

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle: Structuring of Spatial Data



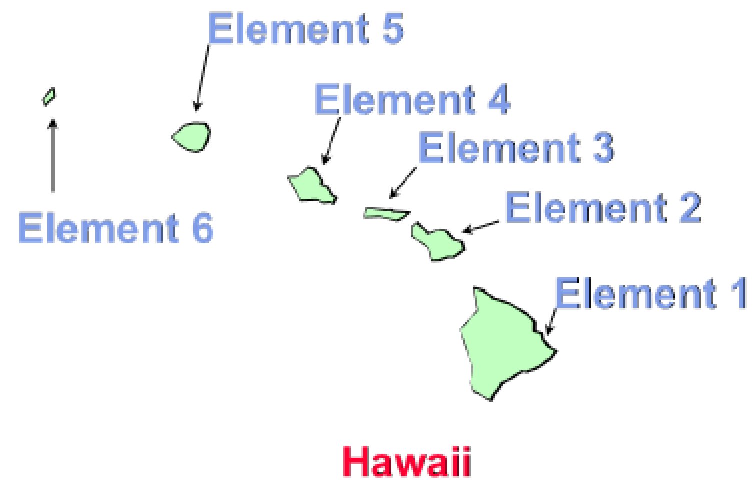
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Oracle: Element

- Basic component of geometric objects
- Element type:
  - point
  - line
  - polygon
- Formed of an ordered sequence of points

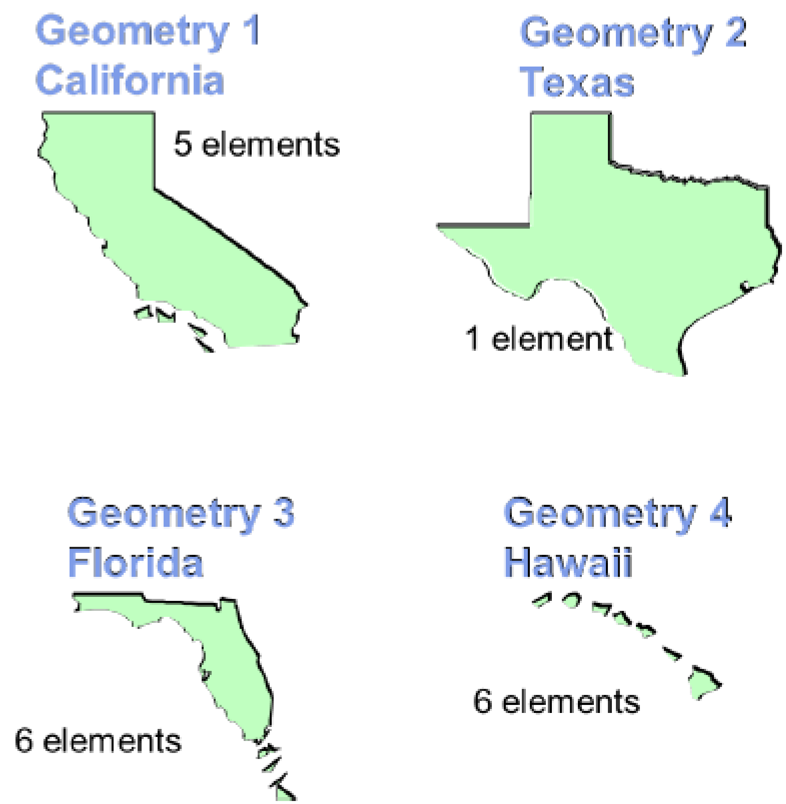


A single geometry composed of 6 elements

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

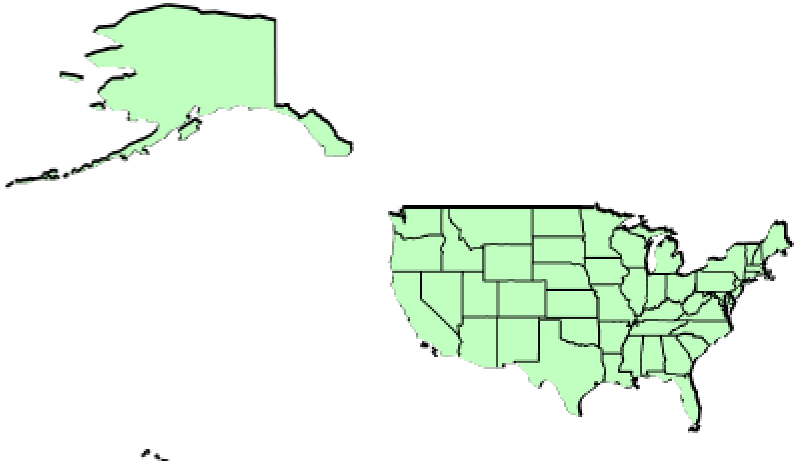
Oracle: Geometry

- Represent a spatial object
- Composed of an ordered list of elements
- May be homogeneous ou heterogeneous



Oracle: Layer

- Represent a geometrical column in a table
- In general, contain objects of the same nature (i.e.having the same attributes)
  - Client layer (points)
  - Street layer (lines)
  - State layer (polygons)



SDO_GEOMETRY Type

- Structure of the SDO_GEOMETRY object

```
SDO_GTYPE          NUMBER
SDO_SRID           NUMBER
SDO_POINT          SDO_POINT_TYPE
SDO_ELEM_INFO      SDO_ELEM_INFO_ARRAY
SDO_ORDINATES      SDO_ORDINATE_ARRAY
```

- Example of use

```
CREATE TABLE states (
    state       VARCHAR2(30),
    totpop      NUMBER(9),
    geom        SDO_GEOMETRY);
```

- SDO_POINT_TYPE

```
x        NUMBER
y        NUMBER
z        NUMBER
```

- SDO_ELEM_INFO_ARRAY and SDO_ORDINATE_ARRAY

```
VARRAY (1048576) OF NUMBER
```

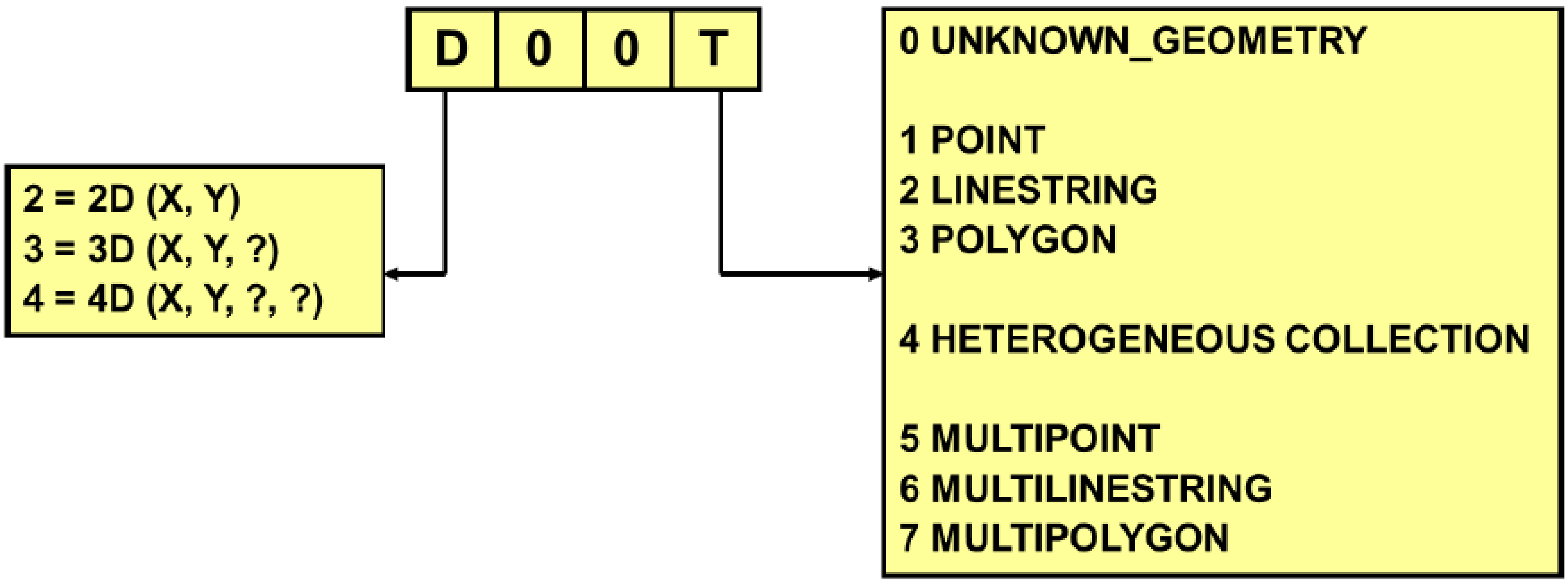
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_GTYPE

- Define the nature of the geometric shape contained in the object



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_SRID

- SRID = Spatial Reference system ID
- Specifies the coordinate system of the object
- List of possible values is found in the table `MDSYS.CS_SRS`
  - More than 1,000 different systems
- A common value: 8307
  - "Longitude / Latitude WGS84"
  - Used by the GPS system
  - Navteq and TeleAtlas data is WGS84
- All geometries of a layer must have the same SRID
- Layers may have different SRIDs
- Automatic conversion for spatial queries

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_POINT

- Object type SDO_POINT_TYPE

```
x         NUMBER
y         NUMBER
z         NUMBER
```

- Example of use

```
INSERT INTO TELEPHONE_POLES (col-1, ..., col-n, geom)
    VALUES (attribute-1, ..., attribute-n,
        SDO_GEOMETRY (
            2001, 8307,
            SDO_POINT_TYPE (-75.2,43.7,null),
            null, null)
    );
```

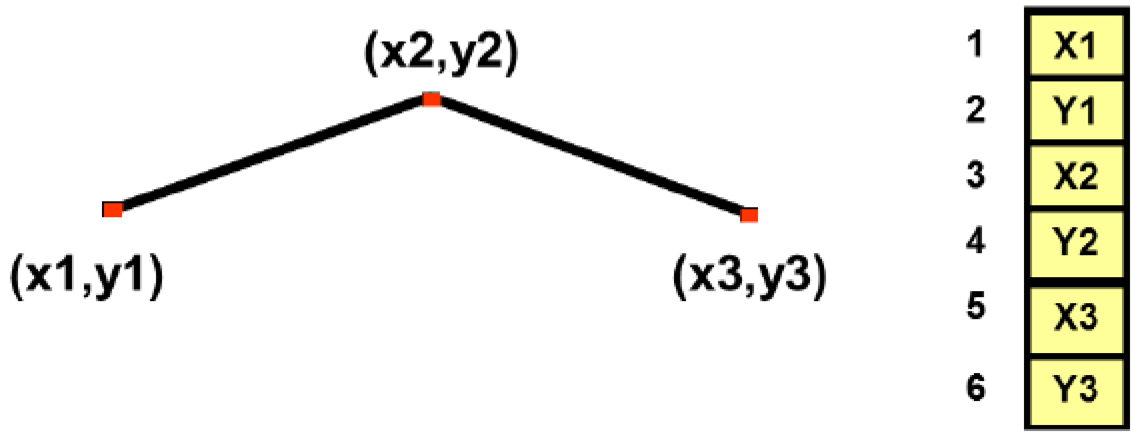
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_ORDINATES

- Object type `SDO_ORDINATE_ARRAY`

  `VARRAY (1048576) OF NUMBER`

- Store the coordinates of lines et polygons



- For example, in 2D two entries per point

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_ELEM_INFO

- Object type SDO_ELEM_INFO_ARRAY

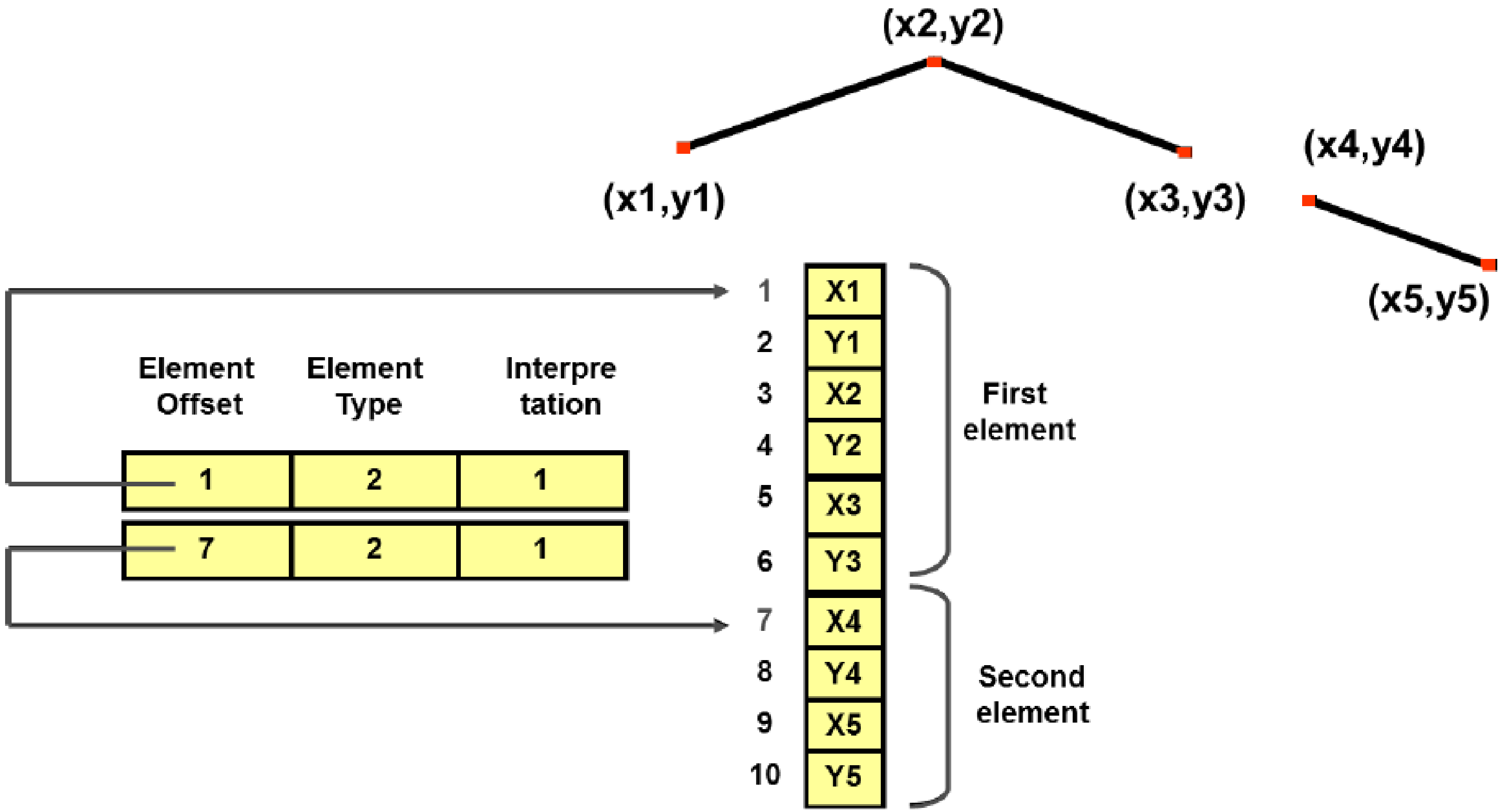
  VARRAY (1048576) OF NUMBER
- Specifies the nature of the elements
- Describes the various components of a complex object
- Three entries per element
  - **Ordinate offset**: Position of the first number for this element in the array SDO_ORDINATES
  - **Element type**: Type of the element
  - **Interpretation**: Straight line, arc, etc..

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

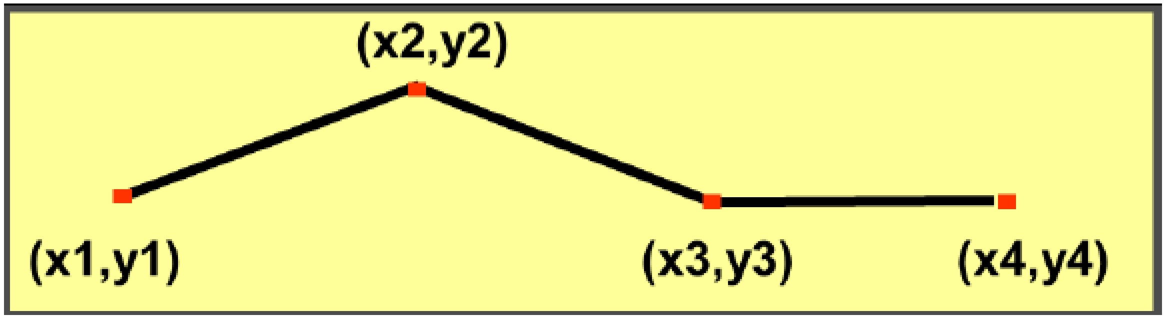
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_ELEM_INFO: Example

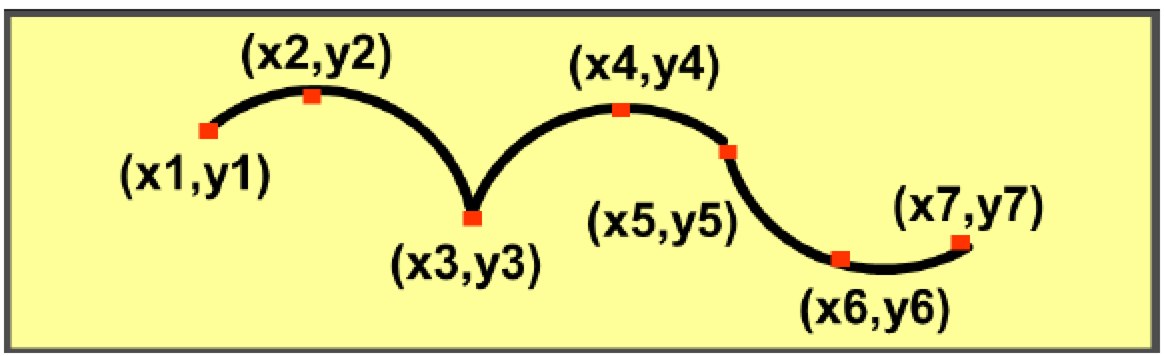


Line Examples

Ordinate offset	Element type	Interpretation
1	2	1

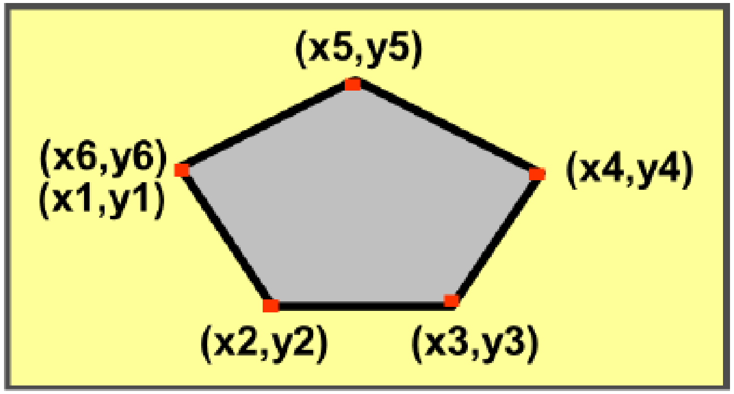


Ordinate offset	Element type	Interpretation
1	2	2

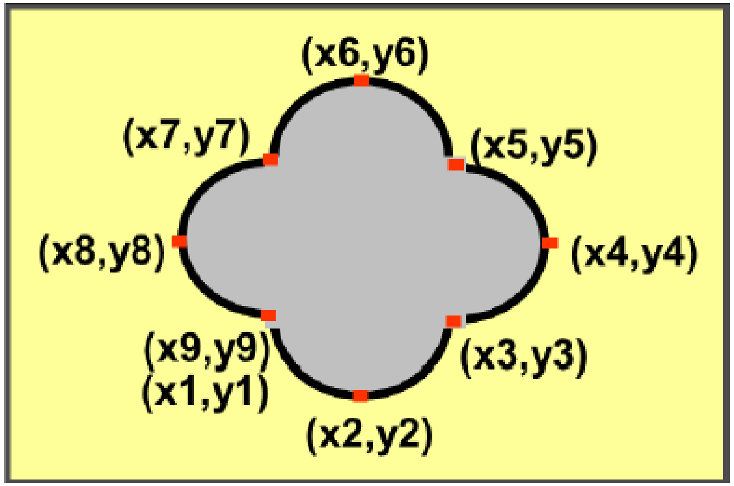


Polygon Examples (1)

Ordinate offset	Element type	Interpretation
1	1003	1



Ordinate offset	Element type	Interpretation
1	1003	2



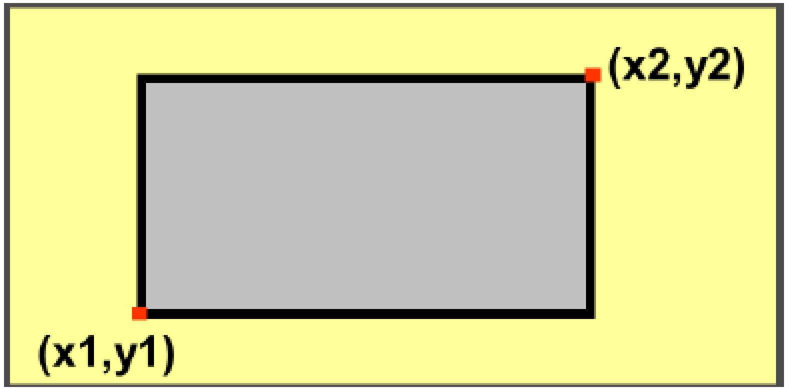
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

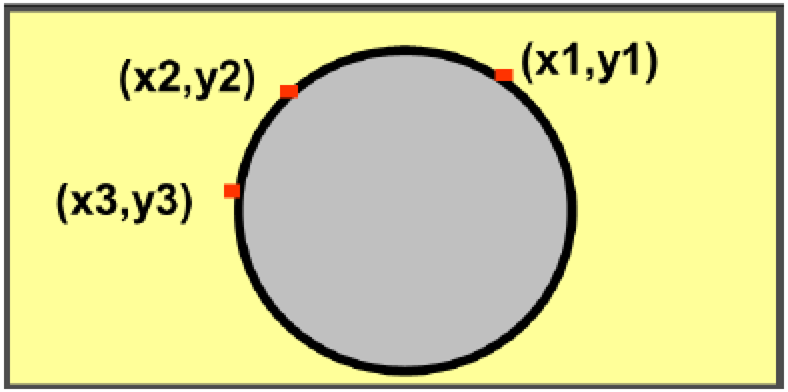
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Polygon Examples (2)

Ordinate offset	Element type	Interpretation
1	1003	3

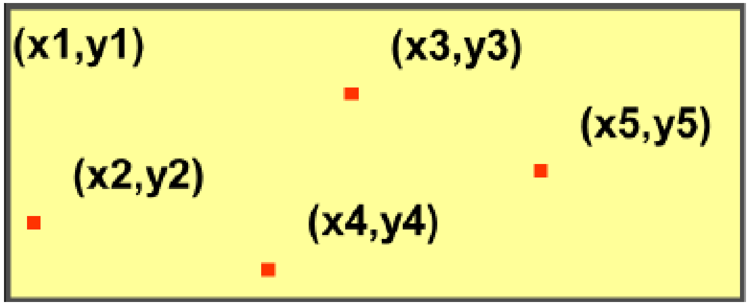


Ordinate offset	Element type	Interpretation
1	1003	4

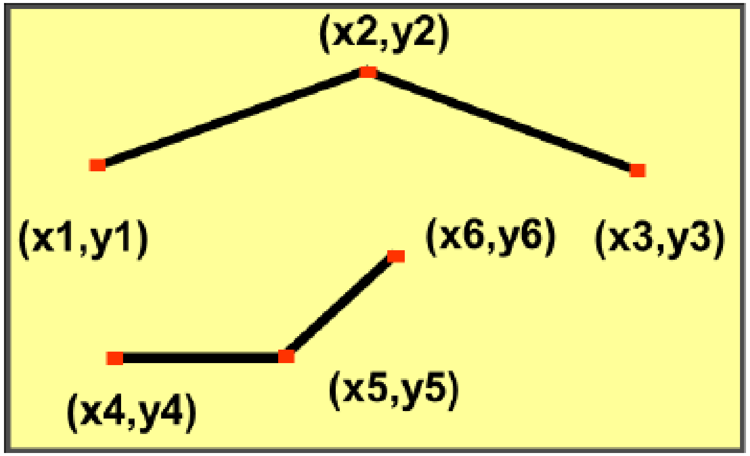


Multi-Point and Multi-Line Examples

Ordinate offset	Element type	Interpretation
1	1	5



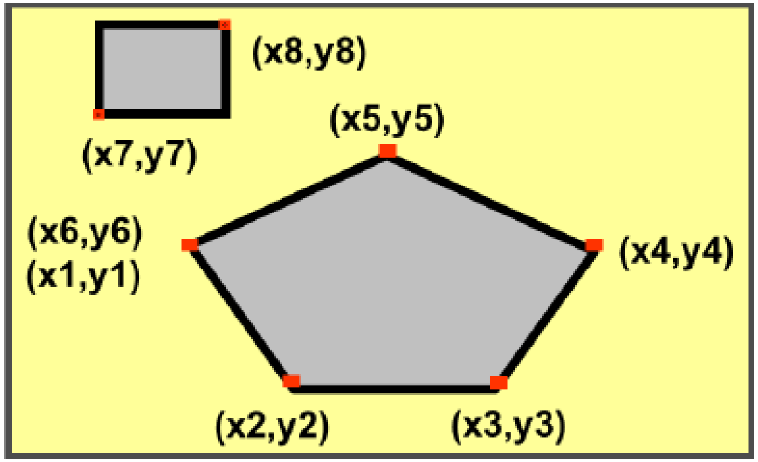
Ordinate offset	Element type	Interpretation
1	2	1
7	2	1



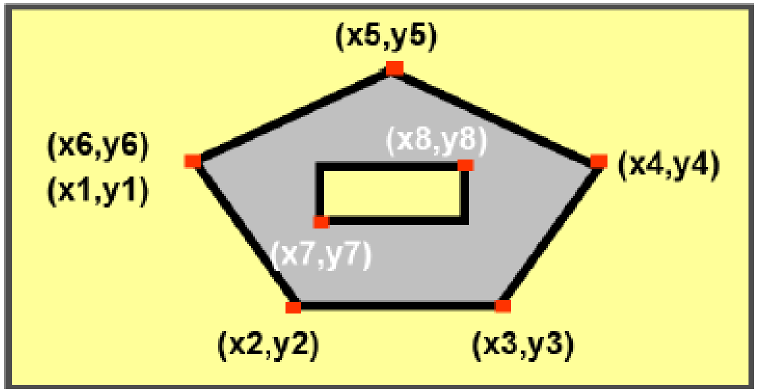
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Multi-Polygon and Polygon with Hole Examples

Ordinate offset	Element type	Interpretation
1	1003	1
13	1003	3



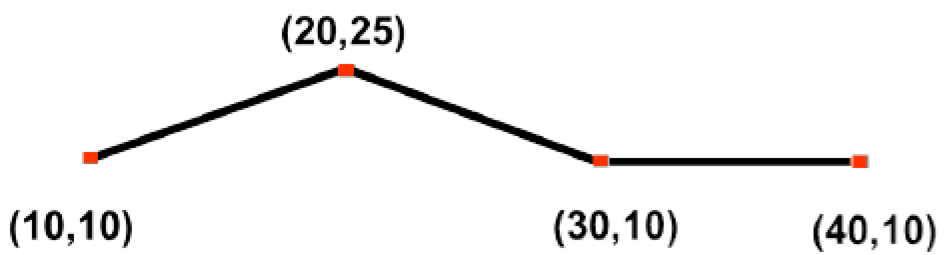
Ordinate offset	Element type	Interpretation
1	1003	1
13	2003	3



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Constructing a Line

```
INSERT INTO LINES (col-1, ..., col-n, geom) VALUES (
    attribute_1, ..., attribute_n,
        SDO_GEOMETRY (
            2002, 8307, null,
            SDO_ELEM_INFO_ARRAY (1,2,1),
            SDO_ORDINATE_ARRAY (
                10,10, 20,25, 30,10, 40,10))
      );
```



Metadata

- Defines the **boundaries** of a layer
  - Minimum and maximum coordinates for each dimension
- Sets the **tolerance** of a layer
  - Maximum distance between two points they are considered distinct
- Defines the **coordinate system** for a layer

```
INSERT INTO USER_SDO_GEOM_METADATA
    (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID) VALUES (
        'ROADS',
        'GEOMETRY',
        SDO_DIM_ARRAY (
            SDO_DIM_ELEMENT('Long', -180, 180, 0.5),
            SDO_DIM_ELEMENT('Lat',   -90,  90, 0.5)),
        8307 );
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Constructing Geometries

- Standard constructor

```
INSERT INTO TELEPHONE_POLES (col-1, ..., col-n, geom)
    VALUES (attribute-1, ..., attribute-n,
        SDO_GEOMETRY (
            2001, 8307,
            SDO_POINT_TYPE (-75.2,43.7,null),
            null, null)
    );
```

- Well-known Text (WKT) constructor

```
INSERT INTO TELEPHONE_POLES (col-1, ..., col-n, geom)
    VALUES (attribute-1, ..., attribute-n,
        SDO_GEOMETRY ('POINT (-75.2 43.7)',8307)
    );
```

- Well-known Binary (WKB) constructor

```
INSERT INTO TELEPHONE_POLES (col-1, ..., col-n, geom)
    VALUES (attribute-1, ..., attribute-n,
        SDO_GEOMETRY (:my_blob,8307)
    );
```

Geometry Extraction: WKT Format

```
SELECT c.geom.get_wkt()
FROM us_counties c
WHERE county = 'Denver';

POLYGON (
  (-105.052597 39.791199, -105.064606 39.789928, ...
   ... -105.024757 39.790947,-105.052597 39.791199),
  (-104.933578 39.698139, -104.936104 39.698299, ...
   ...  -104.9338 39.696701, -104.933578 39.698139))
```

- Do not forget to use an alias!
- Returns the geometry in a CLOB

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Geometry Extraction: WKB Format

```
SELECT county, c.geom.get_wkb()
INTO :my_blob
FROM us_counties c
WHERE state_abrv = 'NH';

CREATE TABLE us_counties_wkb AS
  SELECT county, state_abrv, c.geom.get_wkb() wkb_geom
  FROM us_counties c;
```

- Do not forget to use an alias!
- Returns the geometry in a BLOB

Geometry Extraction: GML Format

```
SELECT city, sdo_util.to_gmlgeometry(location)
FROM us_cities
WHERE state_abrv = 'CO';

<gml:Point srsName="SDO:8307
  xmlns:gml="http://www.opengis.net/gml">
  <gml:coordinates decimal="." cs="," ts=" ">
    -104.872655,39.768035
  </gml:coordinates>
</gml:Point>
```

Generation of XML documents: XMLForest (1)

```
SELECT xmlelement(
         "City",
          xmlattributes(
            'http://www.opengis.net/gml' as "xmlns:gml"),
          xmlforest(
            city as "Name",
            pop90 as "Population",
            xmltype( sdo_util.to_gmlgeometry(location) )
              as "gml:geometryProperty")
       ) AS theXMLElements
FROM us_cities
WHERE state_abrv = ('CO');
```

Generation of XML documents: XMLForest (2)

```
<City xmlns:gml="http://www.opengis.net/gml">
  <Name>Denver</Name>
  <Population>467610</Population>
  <gml:geometryProperty><gml:Point srsName="SDO:8307"
     xmlns:gml="http://www.opengis.net/gml">
     <gml:coordinates decimal="." cs="," ts=" ">
       -104.872655,39.768035 </gml:coordinates>
  </gml:Point></gml:geometryProperty>
</City>
...
<City xmlns:gml="http://www.opengis.net/gml">
  <Name>Lakewood</Name>
  <Population>126481</Population>
  <gml:geometryProperty><gml:Point srsName="SDO:8307"
     xmlns:gml="http://www.opengis.net/gml">
     <gml:coordinates decimal="." cs="," ts=" ">
       -105.113556,39.6952 </gml:coordinates>
  </gml:Point></gml:geometryProperty>
</City>
```

POLITECHNIKA POZNAŃSKA

Manipulation of Geometries in Java

- Java API provided in the kit `SDOAPI.JAR`
- Can be distributed with your applications
  - As the JDBC driver
- Class `JGeometry`
- `load()` and `store()` methods
- Many utility functions
  - Read and write GML
  - Read and write shape files

Reading Geometries

```
// Construct SQL query
String sqlQuery = "SELECT GEOM FROM US_COUNTIES"
// Execute query
Statement stmt = dbConnection.createStatement();
OracleResultSet rs = (OracleResultSet)stmt.executeQuery(sqlQuery);
// Fetch results
while (rs.next())
{
  // Extract JDBC object from record into structure
  STRUCT dbObject = (STRUCT) rs.getObject(1);
  // Import from structure into Geometry object
  JGeometry geom = JGeometry.load(dbObject);
}
```

Extracting Information from Geometries (1)

```
int gType =                     geom.getType();
int gSRID =                     geom.getSRID();
int gDimensions =               geom.getDimensions();
long gNumPoints =               geom.getNumPoints();
long gSize =                    geom.getSize();

boolean isPoint =               geom.isPoint();
boolean isCircle =              geom.isCircle();
boolean hasCircularArcs =       geom.hasCircularArcs();
boolean isGeodeticMBR =         geom.isGeodeticMBR();
boolean isLRSGeometry =         geom.isLRSGeometry();
boolean isMultiPoint =          geom.isMultiPoint();
boolean isRectangle =           geom.isRectangle();
// NON EXHAUSTIVE LIST !
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Extracting Information from Geometries (2)

```
// Point
double gPoint[]  =        geom.getPoint();
// Element info array
int gElemInfo[] =         geom.getElemInfo();
// Ordinates array
double gOrdinates[] =     geom.getOrdinatesArray();
// First and last point
double[] gFirstPoint =    geom.getFirstPoint();
double[] gLastPoint =     geom.getLastPoint();
// MBR
double[] gMBR =           geom.getMBR();
// Java Shape
Shape gShape =            geom.createShape();
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Constructing Geometries (1)

```
// Point
JGeometry geom = new JGeometry(10,5, 8307);
// Point (3D)
JGeometry geom = new JGeometry(10,5,3, 8307);
// Rectangle
JGeometry geom = new JGeometry(10,135, 20,140, 8307);
// Any geometry (compound linestring)
JGeometry geom = new JGeometry(
  2002,8307,
  new int[] {1,4,3, 1,2,1, 3,2,2, 7,2,1},
  new double[] {10,45, 20,45, 23,48, 20,51, 10,51});
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Constructing Geometries (2)

```
// Point
JGeometry geom = JGeometry.createPoint(
  new double[] {10,5}, 2, 8307);
// Linestring
JGeometry geom = JGeometry.createLinearLineString(
  new double[] {10,25, 20,30, 25,25, 30,30}, 2, 8307);
// Simple polygon
JGeometry geom = JGeometry.createLinearPolygon(
  new double[] {10,105, 15,105, 20,110, 10,110, 10,105}, 2, 8307);
// Polygon with voids
JGeometry geom = JGeometry.createLinearPolygon(
  new double[][] {{50,105, 55,105, 60,110, 50,110, 50,105},
    {52,106, 54,106, 54,108, 52,108, 52,106}}, 2, 8307);
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Constructing Geometries (3)

```
// Multi-point
JGeometry geom = JGeometry.createMultiPoint(
  new double[][] {{50,5}, {55,7}, {60,5}}, 2, 8307);
// Multi-linestring
JGeometry geom = JGeometry.createLinearMultiLineString(
  new double[][] {{50,15, 55,15}, {60,15, 65,15}}, 2, 8307);
// Circle (using 3 points on the circumference)
geom = JGeometry.createCircle(15,145, 10,150, 20,150, 8307);
// Circle (using a center point and radius)
geom = JGeometry.createCircle(10,150, 5, 8307);
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Writing Geometries

```
// Construct the SQL statement
String SqlStatement = "INSERT INTO SHAPES (ID, GEOM) VALUES (?,?)";
// Prepare the SQL statement
PreparedStatement stmt = dbConnection.prepareStatement(SqlStatement);
// Convert object into java STRUCT
STRUCT s = JGeometry.store (geom, dbConnection);
// Insert row in the database table
stmt.setInt (1, i);
stmt.setObject (2,s);
stmt.execute();
```

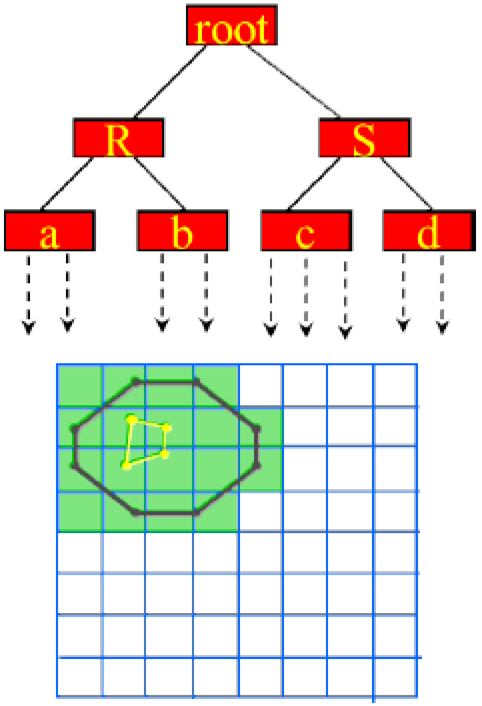
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

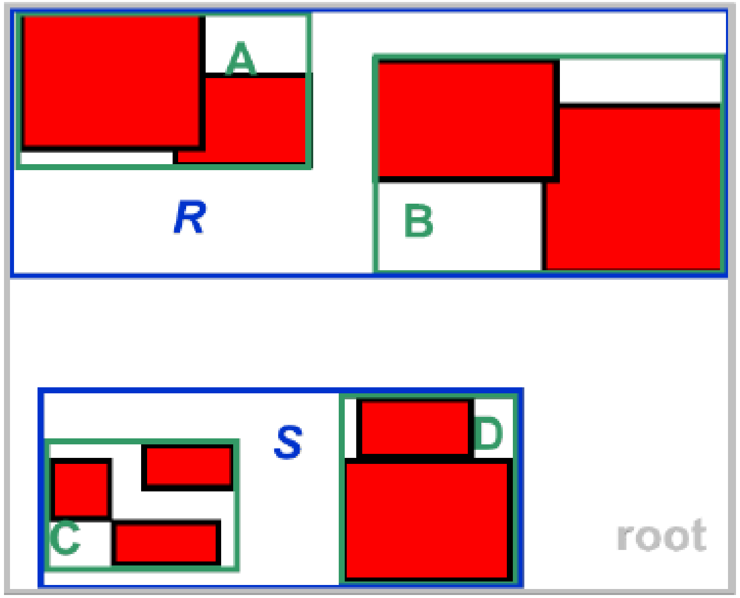
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

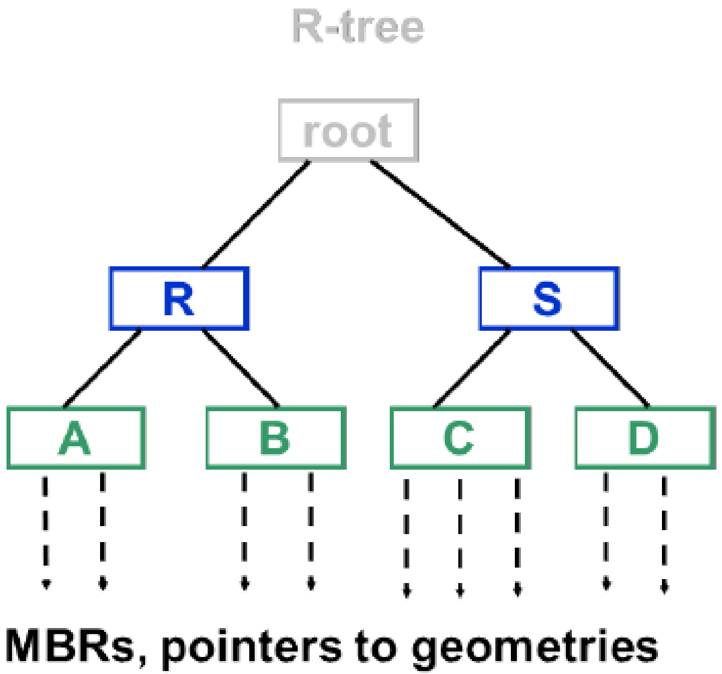
Spatial Indexes

- R-tree indexing
  - Tree rectangles (MBR)
  - Indexing in 2 or 3 dimensions
- Quad-tree indexing
  - Use of a regular grid
  - No longer documented from 10g
  - Do not use, unless exceptions ...
- A spatial index must exist before we can ask spatial queries on a table!



R-Tree Index





- Based on the Minimum Bounding Rectange (MBR) of objects

Creation and Deletion of an R-tree Index

- Index creation

```
create index CUSTOMERS_SIDX
   on CUSTOMERS (LOCATION)
 indextype is MDSYS.SPATIAL_INDEX;
```

- `CREATE INDEX` statement may have additional parameters
  - E.g., to specify the number of dimensions and where to store the index information
- Identifying the `SDO_INDEX_TABLE` that stores the spatial index on the customers table

```
SELECT SDO_INDEX_TABLE FROM USER_SDO_INDEX_INFO
WHERE TABLE_NAME = 'CUSTOMERS' AND COLUMN_NAME='LOCATION';
SDO_INDEX_TABLE
--------------------------------
MDRT_D81F$
```

- Index deletion

```
DROP INDEX <index_name>;
```

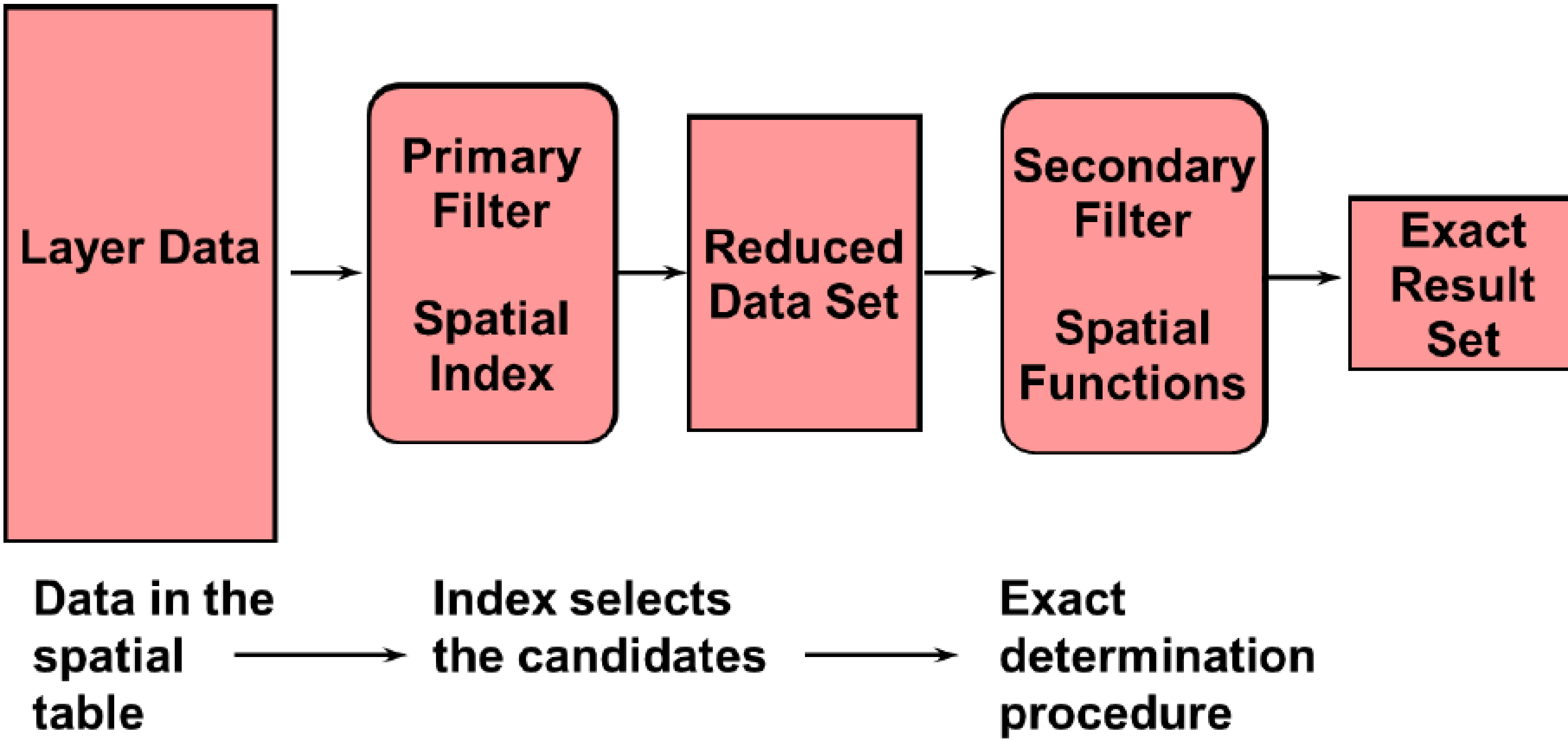
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

How to Find Information about Spatial Indexes

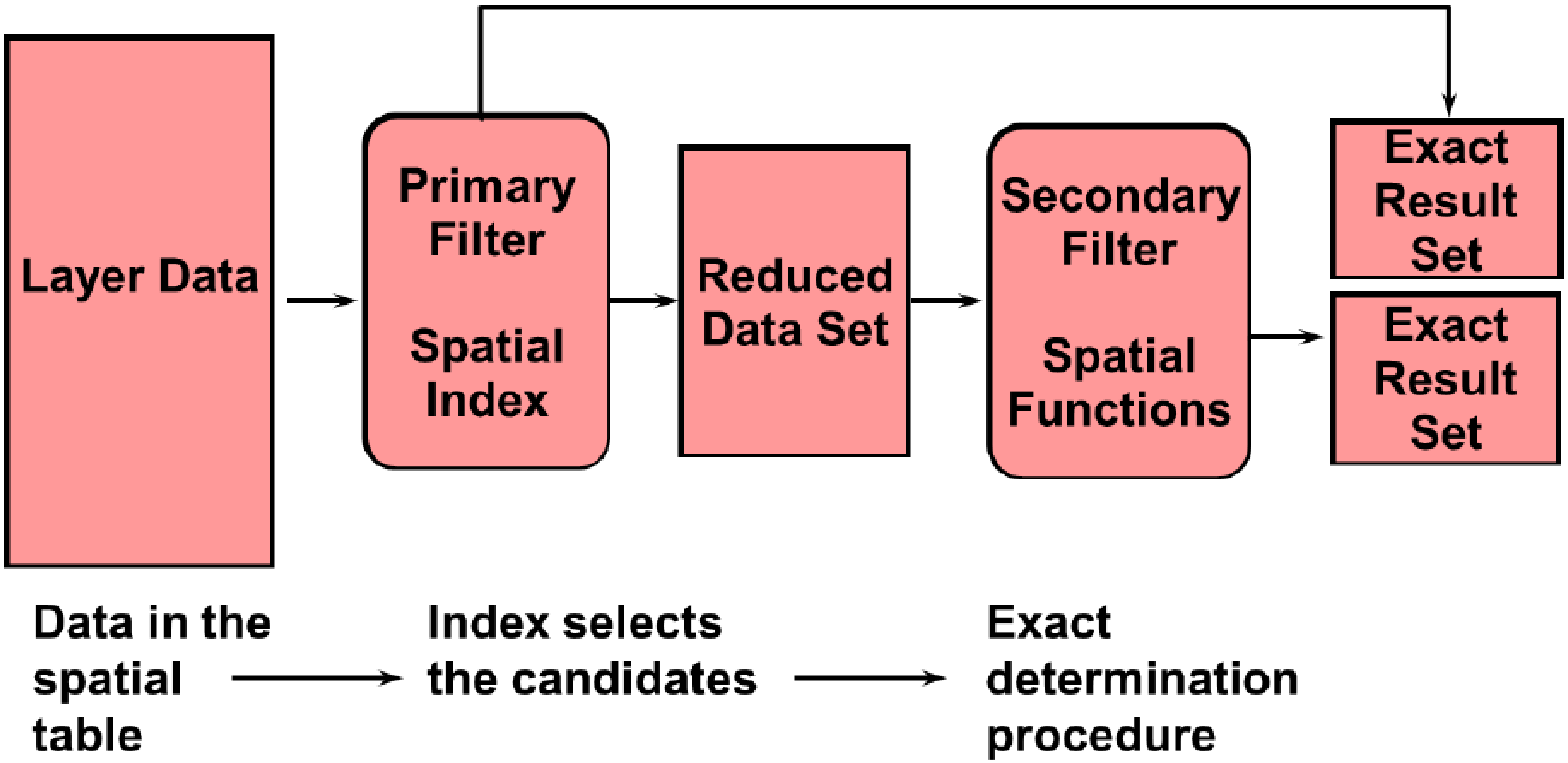
- USER_INDEXES
  - INDEX_TYPE = 'DOMAIN' and ITYP_NAME = 'SPATIAL_INDEX'
- USER_SDO_INDEX_INFO
  - Column SDO_INDEX_TABLE identifies the physical table containing the index proprement: table whose name is of the form MDRT_xxxx$
- USER_SDO_INDEX_METADATA
  - As above, but with more details: number of nodes, node size, height of the index tree, etc.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Query Execution Model



Optimized Query Execution Model



Writing Spatial Queries

- Contain a spatial predicate (WHERE clause)
  - Find the plots along a river
  - Find clients within 5 km of a warehouse
  - Find the nearest branch of a client
- Expressed through specific SQL operators
  - SDO_RELATE, SDO_INSIDE, SDO_TOUCH
  - SDO_WITHIN_DISTANCE
  - SDO_NN
- The spatial index MUST exist, otherwise

```
ORA-13226: interface not supported without a spatial index
ORA-06512: at "MDSYS.MD", line 1723
ORA-06512: at "MDSYS.MDERR", line 8
ORA-06512: at "MDSYS.SDO_3GL", line 387
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Topological Predicates

- Select objects by their topological relationship with another object
  - SDO_INSIDE
  - SDO_CONTAINS
  - SDO_COVERS
  - SDO_COVEREDBY
  - SDO_OVERLAPS
  - SDO_TOUCH
  - SDO_EQUAL
  - SDO_ANYINTERACT
- Example

```
WHERE SDO_INSIDE ( <geometry-1>, <geometry-2> ) = 'TRUE'
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Generic Topological Operator

- SDO_RELATE generic operator with a specific mask

```
WHERE SDO_RELATE (<geometry-1>, <geometry-2>, 'MASK=xxxx') = 'TRUE'
```

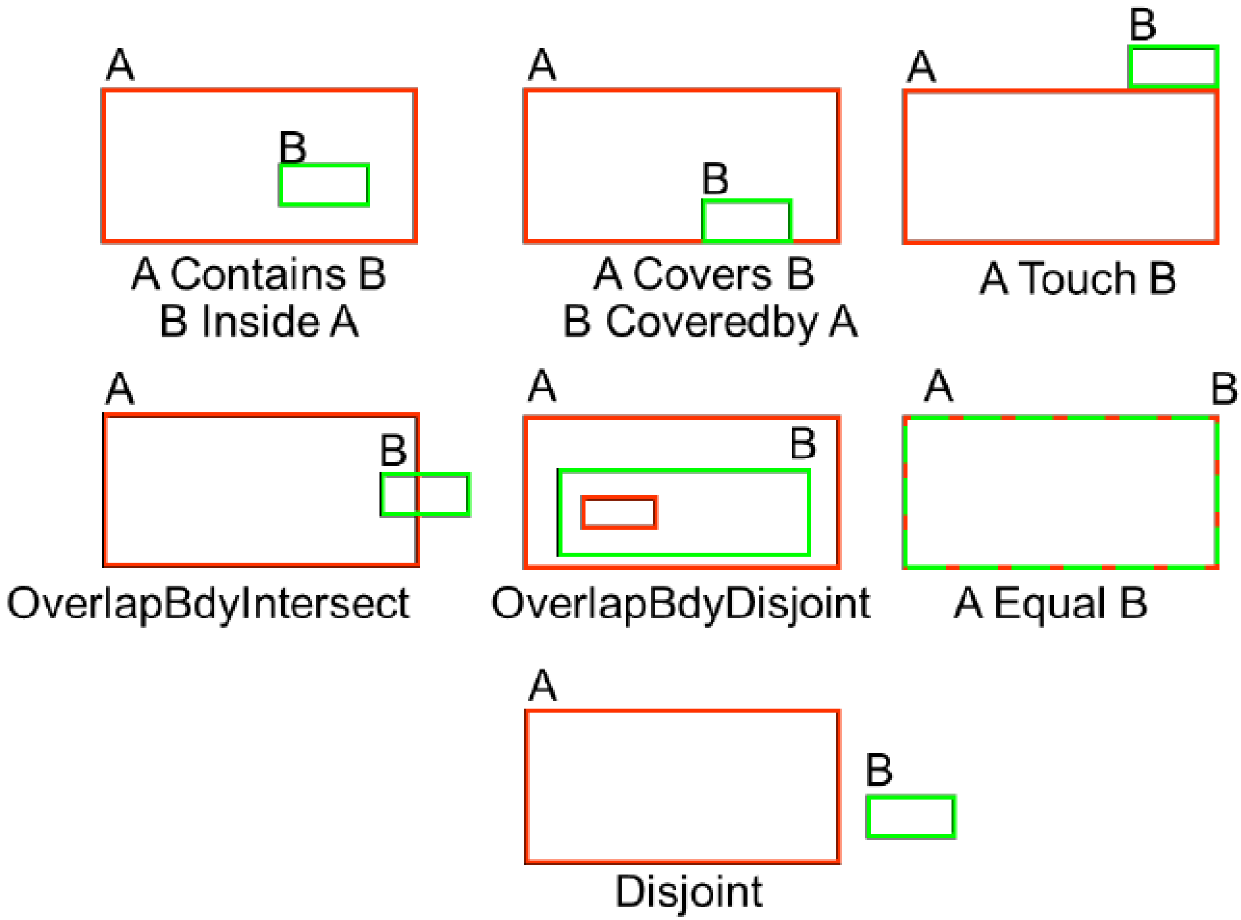
- Mask may be 'INSIDE', 'CONTAINS', 'TOUCH', etc.
- Or a combination: 'INSIDE+COVEREDBY'

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Topological Operators



Example Queries

- Which parks are entirely contained in the state of Wyoming?

```
SELECT p.name
  FROM us_parks p, us_states s
 WHERE s.state = 'Wyoming'
   AND SDO_INSIDE (p.geom, s.geom) = 'TRUE';
```

- Equivalent to

```
   AND SDO_RELATE(p.geom,s.geom,'MASK=INSIDE') = 'TRUE';
```

- Which states contain all or part of Yellowstone Park?

```
SELECT s.state
  FROM us_states s, us_parks p
 WHERE SDO_ANYINTERACT (s.geom, p.geom) = 'TRUE'
   AND p.name = 'Yellowstone NP';
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Example Queries

- In which competing jurisdictions is my client?

```
SELECT s.id, s.name
  FROM customers c, competitors_sales_regions s
 WHERE c.id = 5514 AND SDO_CONTAINS (s.geom, c.location) = 'TRUE';
```

- Find all counties around Passaic County (NJ)

```
SELECT c1.county, c1.state_abrv
FROM us_counties c1, us_counties c2
WHERE c2.state = 'New Jersey' AND c2.county = 'Passaic'
   AND SDO_TOUCH (c1.geom, c2.geom) = 'TRUE';
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Queries with a Constant Window

- Find all customers of type "Platinum" in a rectangular area

```
SELECT name, category
  FROM customers
 WHERE SDO_INSIDE (
         location,
         sdo_geometry (2003, 8307, null,
           sdo_elem_info_array (1,1003,3),
           sdo_ordinate_array (
             -122.413, 37.785,-122.403, 37.792))
       )='TRUE'
   AND customer_grade = 'PLATINUM';
```

- In which competitors sales territories is located a geographical point?

```
SELECT id, name
  FROM competitors_sales_regions
 WHERE SDO_CONTAINS (
         geom,
         SDO_GEOMETRY(2001, 8307,
           SDO_POINT_TYPE(-122.41762, 37.7675089, NULL),
           NULL, NULL)
       ) = 'TRUE';
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Queries Based on Distance

- Select objects according to distance from another object
- Operator SDO_WITHIN_DISTANCE

```
SDO_WITHIN_DISTANCE(
    <geometry-1>, <geometry-2>,
    'DISTANCE=distance UNIT=unit' ) = 'TRUE
```

- Distance can be expressed in any unit of measure
- If no unit is specified, the distance is expressed in the unit of the coordinate system (if projected)
- For longitude/latitude data, these are meters

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

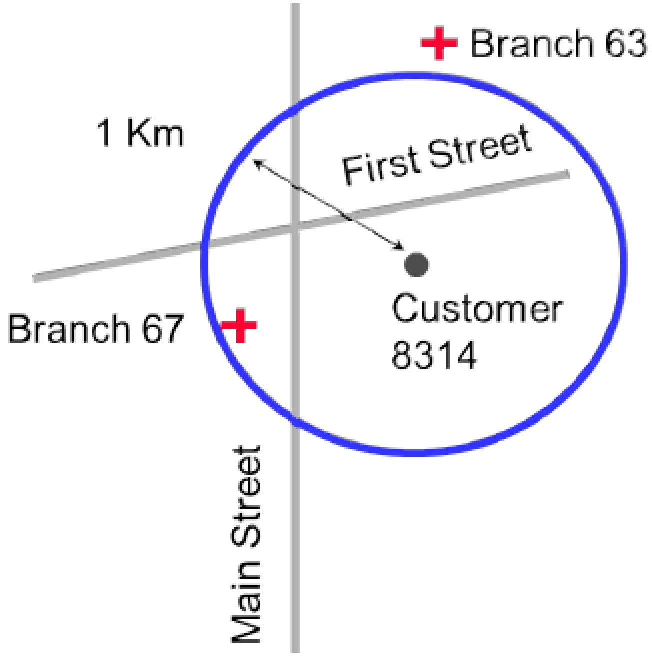
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Examples of Research on Distance (1)

- Which agencies are less than 1km from this client?

```
SELECT b.id, b.phone_number
FROM customers c, branches b
WHERE c.id = 8314
  AND SDO_WITHIN_DISTANCE(
        b.location, c.location,
        'distance=1 unit=km')
      = 'TRUE';
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Examples of Research on Distance (2)

- How many customers in each category are located within 1/4 mile of my office number 77?

```
SELECT customer_grade, COUNT(*)
  FROM branches b, customers c
 WHERE b.id=77
   AND SDO_WITHIN_DISTANCE (
         c.location, b.location,
         'DISTANCE=0.25 UNIT=MILE')='TRUE'
 GROUP BY customer_grade;
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Research Based on Proximity

- Selects the N closest objects of another object
- SDO_NN operator with the number of objects to be returned

```
WHERE SDO_NN (
    <geometry-1>, <geometry-2>,
    'SDO_NUM_RES=n') = 'TRUE
```

- ROWNUM can be used to limit results

```
WHERE SDO_NN (
    <geometry-1>, <geometry-2>) = 'TRUE
     AND ROWNUM <=n
```

- To use when additional filtering
- SDO_NN_DISTANCE() used to categorize answers by distance

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

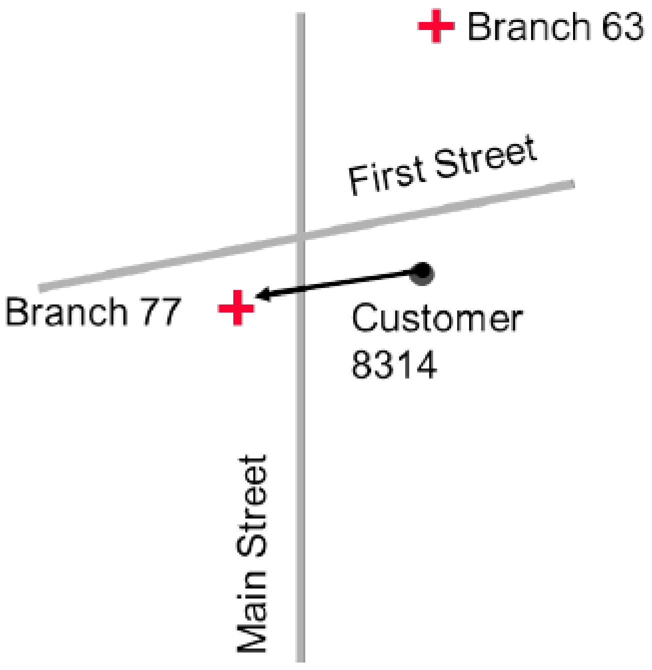
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Example of Proximity Search

- What is the nearest office to this client?

```
SELECT b.id, b.phone_number
  FROM customers c,
       branches b
 WHERE c.id = 8314
   AND SDO_NN(
        b.location, c.location,
        'sdo_num_res=1')
       = 'TRUE';
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Example of Proximity Search

- What are my five customers closest to this competitor?

```
SELECT c.id, c.name, c.customer_grade
  FROM competitors co, customers c
 WHERE co.id=1
   AND SDO_NN (
         c.location, co.location,
         'SDO_NUM_RES=5')='TRUE' ;

 809 LINCOLN SUITES                      GOLD
1044 MUSEUM OF THE THIRD DIMENSION       SILVER
1526 INTERNATIONAL FINANCE               SILVER
1538 MCKENNA AND CUNEO                   SILVER
8792 DESTINATION HOTEL & RESORTS         GOLD
```

- This only works if no other selection criterion is present!

Classifying Results by Distance

- What are my five customers closest to this competitor?

```
SELECT c.id, c.name, c.customer_grade,
       SDO_NN_DISTANCE(1) distance
  FROM competitors co, customers c
 WHERE co.id=1
   AND SDO_NN (
        c.location, co.location,
        'SDO_NUM_RES=5', 1)='TRUE'
 ORDER BY distance;

1538 MCKENNA AND CUNEO                SILVER          88
 809 LINCOLN SUITES                   GOLD            95
1044 MUSEUM OF THE THIRD DIMENSION    SILVER         139
8792 DESTINATION HOTEL & RESORTS      GOLD           145
1526 INTERNATIONAL FINANCE            SILVER         215
```

Additional Selection Criteria: First Attempt

- What are my five 'GOLD' clients closest to this competitor?

```
SELECT c.id, c.name, c.customer_grade
  FROM competitors co, customers c
 WHERE co.id=1
   AND SDO_NN (
            c.location, co.location,
            'SDO_NUM_RES=5')='TRUE'
   AND c.customer_grade = 'GOLD';

 809 LINCOLN SUITES                        GOLD
8792 DESTINATION HOTEL & RESORTS           GOLD
```

- Results may be incorrect!
- The 5 closest customers are read, and then those of type 'GOLD' are selected

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Additional Selection Criteria: Correct Query

- What are my five 'GOLD' clients closest to this competitor?

```
SELECT c.id, c.name, c.customer_grade
  FROM competitors co, customers c
 WHERE co.id=1
   AND SDO_NN (
        c.location, co.location)='TRUE'
   AND c.customer_grade = 'GOLD'
   AND ROWNUM <= 5;

 809 LINCOLN SUITES                         GOLD
8792 DESTINATION HOTEL & RESORTS            GOLD
 810 HOTEL LOMBARDY                         GOLD
7821 RENAISSANCE MAYFLOWER HOTEL            GOLD
6326 HOTEL LOMBARDY                         GOLD
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Joins: SDO_JOIN()

- To find correlations between two tables
  - Based on topology or distance
- Compares all objects in a table with all those of another table
- Requires an R-Tree index on each table
- Technically implemented as a function that returns a table

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SDO_JOIN Function

```
SDO_JOIN( table_name-1, column_name-1,
          table_name-2, column_name-2
         [,'parameters'] [, preserve_join_order])
RETURN SDO_ROWIDSET;

SQL> DESC sdo_rowidset;
 SDO_ROWIDSET TABLE OF MDSYS.SDO_ROWIDPAIR
 Name              Null?    Type
 ----------------- -------- ------------
 ROWID1                     VARCHAR2(24)
 ROWID2                     VARCHAR2(24)
```

SDO_JOIN Function Example: Topological Predicate

- Associate to each GOLD customer the sales territory in which it is located

```
SELECT s.id, c.id, c.name
  FROM customers c,
       sales_regions s,
       TABLE(SDO_JOIN(
             'customers', 'location',
             'sales_regions', 'geom',
             'mask=inside')) j
 WHERE j.rowid1 = c.rowid
   AND j.rowid2 = s.rowid
   AND c.customer_grade = 'GOLD'
 ORDER BY s.id, c.id;;
```

SDO_JOIN Function Example: Distance Predicate

- Find all gold customers who are less than 500 meters from one of our branches in San Francisco

```
SELECT DISTINCT c.id, c.name, b.id
  FROM customers c,
       branches b,
       TABLE(SDO_JOIN(
             'CUSTOMERS', 'LOCATION',
             'BRANCHES', 'LOCATION',
             'DISTANCE=500 UNIT=METER')) j
WHERE j.rowid1 = c.rowid
  AND j.rowid2 = b.rowid
  AND c.customer_grade = 'GOLD'
  AND b.city = 'SAN FRANCISCO';
```

Spatial Functions

	Unary Operations	Binary Operations
Numerical Result	SDO_AREA SDO_LENGTH	SDO_DISTANCE
Results in new object	SDO_CENTROID SDO_CONVEXHULL SDO_POINTONSURFACE SDO_BUFFER	SDO_DIFFERENCE SDO_INTERSECTION SDO_UNION SDO_XOR

- Objects must be in the same coordinate system!

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Calculations: Length, Area and Distance

- SDO_AREA(g): Calculates the area of a polygon
- SDO_LENGTH(g): Calculates the length of a line (or the perimeter of a polygon)
- SDO_DISTANCE(g1,g2): Calculates the distance between two objects
- The unit of measure of the result can be specified

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Calculations: Examples

- What is the total area of Yellowstone National Park?

```
SELECT sdo_geom.sdo_area(geom,0.005,'unit=sq_km')
  FROM us_parks
 WHERE name = 'Yellowstone NP';
```

- What is the length of the Mississippi river?

```
SELECT sdo_geom.sdo_length(geom,0.005,'unit=km')
  FROM us_rivers
 WHERE name = 'Mississippi';
```

- What is the distance between Los Angeles and San Francisco?

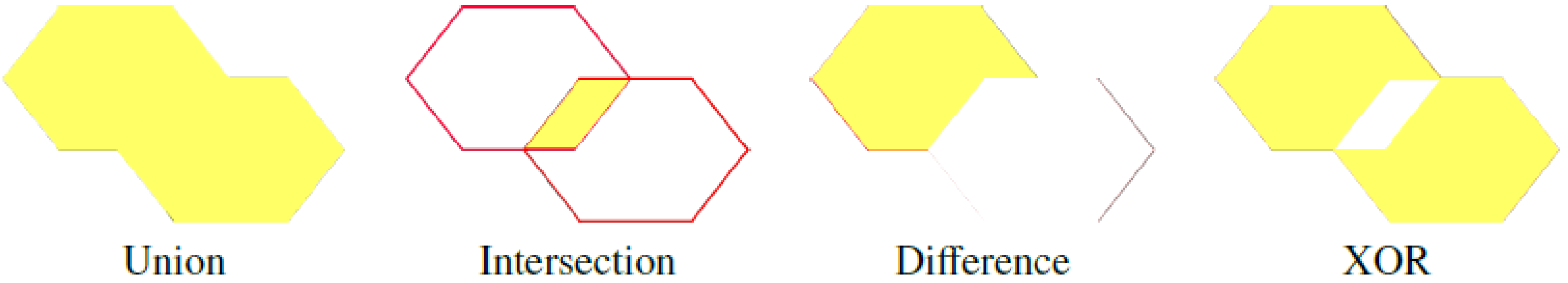
```
SELECT sdo_geom.sdo_distance(a.location, b.location, 0.005, 'unit=mile')
  FROM us_cities a, us_cities b
 WHERE a.city = 'Los Angeles'
   AND b.city = 'San Francisco';
```


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Generating Objects

- SDO_BUFFER(g,size): Generates a buffer size chosen
  - The dimension (size) can be negative for an internal buffer
- SDO_CENTROID(g): Calculates the center of gravity of a polygon
  - May be outside the polygon!
- SDO_CONVEXHULL(g): Generates the convex hull of the object (line or polygon)
- SDO_MBR(g): Generates the bulk of the rectangle object (line or polygon)

Combining Objects



- SDO_UNION(g1,g2): Produces an object that represents the geometric union of the two given objects
- SDO_INTERSECTION(g1,g2): Produces an object that represents the geometric intersection of the two given objects
- SDO_DIFFERENCE(g1,g2): Produces an object that represents the geometric difference of the two given objects
- SDO_XOR(g1,g2): Produces an object that represents the symmetric difference of the two given objects
  - Equivalent to the union minus the intersection

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Combining Objects: Example

- What is the area occupied by the Yellowstone Park in the state it occupies?

```
SELECT s.state, sdo_geom.sdo_area (
         sdo_geom.sdo_intersection (s.geom, p.geom, 0.5),
         0.5, 'unit=sq_km') area
  FROM us_states s, us_parks p
 WHERE SDO_ANYINTERACT (s.geom, p.geom) = 'TRUE'
   AND p.name = 'Yellowstone NP';

STATE                          AREA
-------------------------- ----------
Wyoming                    8100.75346
Montana                    640.295989
Idaho                      154.659879;
```

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Aggregation

- Aggregate functions (like SUM, COUNT, AVG ...)
- Operate on the set of objects
- SDO_AGGR_MBR: Returns the rectangle of space around a set of objects
- SDO_AGGR_UNION: Computes the union of a set of geometric objects
- SDO_AGGR_CENTROID: Calculates the centroid of a set of objects
- SDO_AGGR_CONVEXHULL: Calculates the convex hull around a set of objects

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Aggregation Examples

- Find the focal point of all our customers in Daly City

```
SELECT SDO_AGGR_CENTROID(SDOAGGRTYPE(location,0.5)) center
  FROM customers
 WHERE city = 'DALY CITY';
```

- Calculate the number of customers in each zip code, and calculate the focal point for these clients

```
SELECT COUNT(*), postal_code,
       SDO_AGGR_CENTROID(SDOAGGRTYPE(location,0.5)) center
  FROM customers
 GROUP BY postal_code;
```


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spatial Databases: Conclusions

- Location information is essential for an increasingly number of applications
- GIS were initially used for managing spatial data
- Since 1990s, major DBMS have been extended with spatial capabilities
  - Oracle Spatial was first introduced into version 8g
  - Open source implementations exist for many years
  - SQL Server introduced spatial support in version 2008
- Since the 1990s the Open Geo Spatial Consortium (OGC) has been working on defining standards
  - These standards are also international standards (ISO)
- Systems vary considerably on the support of standards

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego