25/09/2023

DATA RETRIEVAL

ITC / DEPARTMENT OF GEO-INFORMATION PROCESSING

Presented by Lucas De Oto - September 2023

ITC

UNIVERSITY OF TWENTE.

1

OVERVIEW

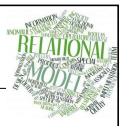
- First principles of data retrieval:
  - Tuple selection
  - Attribute projection
  - Tuple selection & attribute projection
- SQL basics
- Simple method of query definition
- Querying more than 1 relation at the time
- Simple JSP query
- Types of JSP query
- Spatial queries

ITC UNIVERSITY OF TWENTE.

2

TERMINOLOGY

DATA EXTRACTION, QUERY AND TRANSACTION



Data extraction

- The process of acquiring precisely the data that is required/wanted/needed from a data set/database

Query

- Precisely formulated request to extract data from the database.
- Data extraction from the database without altering it

Transaction

- Precisely formulated request to make changes to data in the database.

ITC UNIVERSITY OF TWENTE.

3

QUERYING

FUNDAMENTALS

To able to **properly query a database** we need to:

1. **Thoroughly know the data** set and what particularly will act as the input.
2. Precisely **understand what needs to be produced** as output
3. **'Speak' the DB query language**

ITC UNIVERSITY OF TWENTE.

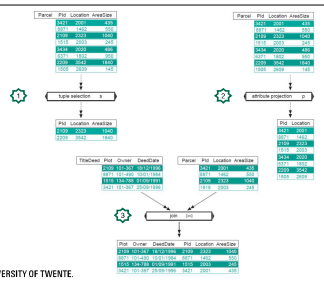
4

25/09/2023

25/09/2023

QUERYING

BASIC OPERATORS



ITC UNIVERSITY OF TWENTE.

5

QUERYING

BASIC OPERATORS

The three query operators have **some common traits**:

1. All of them require **input** and produce **output**
2. Both input and output are **relations**. This guarantees that the output of one query can be the input of another query, which makes it possible to build more and more complex queries.

The three operators can be combined to define queries of higher complexity

ITC UNIVERSITY OF TWENTE.

6

TERMINOLOGY

TUPLE SELECTION

Tuple selection

- **Retrieval of tuples** specified by a given condition.
- Works like a filter: it allows tuples that meet the selection condition to pass and disallows tuples that do not meet the condition.

Relation

	attribute 1	attribute 2	attribute 3
tuple 1			
tuple 2			
tuple 3			
tuple 4			
tuple 5			
tuple 6			
tuple 7			
tuple 8			
tuple 9			

Output of the tuple selection

	attribute 1	attribute 2	attribute 3
tuple 1			
tuple 4			
tuple 5			
tuple 6			

- ✓ Some tuples in the input are selected, others are left out.
- ✓ We get a subset of the input data

ITC UNIVERSITY OF TWENTE.

7

TERMINOLOGY

ATTRIBUTE PROJECTION

Attribute projection

- **Retrieval of indicated attributes** from all tuples in the relation.
- We are selecting a subset of the attributes present in the input

Relation

	attribute 1	attribute 2	attribute 3
tuple 1			
tuple 2			
tuple 3			
tuple 4			
tuple 5			
tuple 6			
tuple 7			
tuple 8			
tuple 9			

Output of the attribute projection

	attribute 1	attribute 3
tuple 1		
tuple 2		
tuple 3		
tuple 4		
tuple 5		
tuple 6		
tuple 7		
tuple 8		
tuple 9		

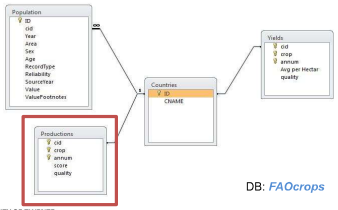
- ✓ We are 'projecting on' attributes 1 and 3.

ITC UNIVERSITY OF TWENTE.

8

QUERYING THE DATABASE
HOW TO..?

In order to understand basic queries, we will go through some examples



DB: *FAOcrops*

UNIVERSITY OF TWENTE.

9

QUERYING THE DATABASE
HOW TO..?

We want to know:

Q1 All the information related to worldwide crop production in 2009

Q2 Which crops were grown in which countries (the country code) and which year?

Q3 Which crops were grown in which countries (the country code) in 2009?

For each of these we will only **use 1 relation!**

UNIVERSITY OF TWENTE.

10

QUERYING THE DATABASE
HOW TO..?

In order to define a query we first need to have a **clear idea of the problem** that we are tackling.

But once we do, we still need to correctly **design our query**:

- Which questions to pose?
- In which order should we pose these questions?
- What to do if the problem to tackle is more complex?

How to do a query formulation in a **methodical way?**

UNIVERSITY OF TWENTE.

11

QUERYING THE DATABASE
SIMPLE METHOD FOR QUERY DEFINITION

Steps:

1. **INPUT RELATION**
   - On which relation is the query based?
   - What will be the tuple variable?
2. **SELECTION CONDITION**
   - What is the condition that the selected tuple for the output must fulfill?
3. **OUTPUT RELATION**
   - What are the attributes in the output?

UNIVERSITY OF TWENTE.

12

QUERYING THE DATABASE
HOW TO..?

- In **SQL**:

```
SELECT output tuple definition (step 3)
FROM input tuple declaration (step 1)
WHERE selection condition (step 2)
```

UNIVERSITY OF TWENTE.

13

QUERYING THE DATABASE
TUPLE SELECTION EXAMPLE

Retrieve all the information related to worldwide crop production in 2009

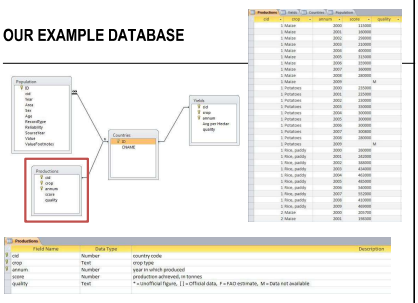
- In FAOcrops:
  - What table do we need?
  - What is the selection condition?
  - What are the attributes that we require in the output?

Q1

UNIVERSITY OF TWENTE.

14

OUR EXAMPLE DATABASE



UNIVERSITY OF TWENTE.

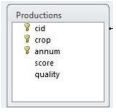
15

QUERYING THE DATABASE
TUPLE SELECTION EXAMPLE

Retrieve all the information related to worldwide crop production in 2009

Q1

- In FAOcrops:
  - What table do we need?
    - Productions
  - What is the selection condition?
    - Annum = 2009
  - What are the attributes that we require in the output?
    - All attributes from Productions



Relation instance when performing query:
➢ **5** attributes and **6930** tuples

UNIVERSITY OF TWENTE.

16

QUERYING THE DATABASE
TUPLE SELECTION EXAMPLE

Retrieve all the information related to worldwide crop production in 2009

Q1

- In FAOcrops.mdb:
  - What table do we need?
    - Productions
  - What is the selection condition?
    - Year 2009 (Annum = 2009)
  - What are the attributes that we require in the output?
    - All attributes of Productions
- In SQL:

```
SELECT *
FROM Productions
WHERE annum = 2009
```

select pilih coloumn apa, kalau from table apa kalau where conditionnya (baris apa)

UNIVERSITY OF TWENTE.

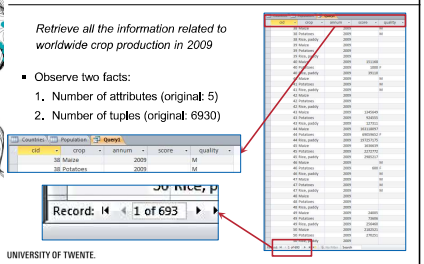
17

QUERYING THE DATABASE

Q1

Retrieve all the information related to worldwide crop production in 2009

- Observe two facts:
  1. Number of attributes (original: 5)
  2. Number of tuples (original: 6930)



UNIVERSITY OF TWENTE.

18

QUERYING THE DATABASE
ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (country code) and which year?

- In FAOcrops:
  - What table do we need?
  - What is the selection condition?
  - What are the attributes that we require in the output?

Q2

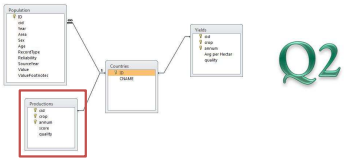
UNIVERSITY OF TWENTE.

19

QUERYING THE DATABASE
ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (country code) and which year?

Q2



UNIVERSITY OF TWENTE.

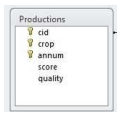
20

QUERYING THE DATABASE
ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (the country code) and which year?

Q2

- In FAOcrops:
  - What table do we need?
    - Productions
  - What is the selection condition?
    - none
  - What are the attributes that we require in the output?
    - crop, cid, year(annum)



Relation instance when performing query:
- 5 attributes and 6930 tuples

UNIVERSITY OF TWENTE.

21

QUERYING THE DATABASE
ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (the country code) and which year?

Q2

- In FAOcrops.mdb:
  - What table do we need?
    - Productions
  - What is the selection condition?
    - none
  - What is the output tuple expression?
    - crop, cid, year(annum)
- In SQL:

```
SELECT crop, cid, annum
FROM Productions
```

UNIVERSITY OF TWENTE.

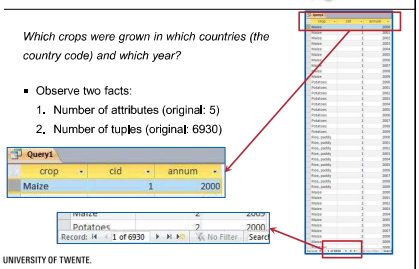
22

QUERYING THE DATABASE

Q2

Which crops were grown in which countries (the country code) and which year?

- Observe two facts:
  1. Number of attributes (original: 5)
  2. Number of tuples (original: 6930)



UNIVERSITY OF TWENTE.

23

QUERYING THE DATABASE
TUPLE SELECTION & ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (the country code) in 2009?

- In FAOcrops.mdb?:
  - What table do we need?
  - What is the selection condition?
  - What are the attributes that we require in the output?

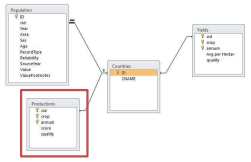
Q3

UNIVERSITY OF TWENTE.

24

QUERYING THE DATABASE
TUPLE SELECTION & ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (the country code) in 2009?



Q3

UNIVERSITY OF TWENTE.

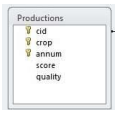
25

QUERYING THE DATABASE
TUPLE SELECTION & ATTRIBUTE PROJECTION EXAMPLE

Which crops were grown in which countries (the country code) in 2009?

Q3

- In FAOcrops.mdb?
  - What table do we need?
    - Productions
  - What is the selection condition?
    - annum = 2009
  - What are the attributes that we require in the output?
    - crop, cid



Relation instance when performing query:
➢ **5** attributes and **6930** tuples

UNIVERSITY OF TWENTE.

26

QUERYING THE DATABASE
TUPLE SELECTION EXAMPLE

Which crops were grown in which countries (the country code) in 2009?

Q3

- In FAOcrops.mdb:
  - What is the table we need?
    - Productions
  - What is the selection condition?
    - Year 2009 (Annum = 2009)
  - What are the attributes we require in the output?
    - crop, cid
- In SQL:

```
SELECT crop, cid      3
FROM Productions      1
WHERE annum = 2009    2
```

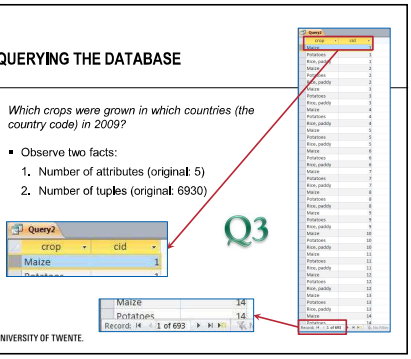
UNIVERSITY OF TWENTE.

27

QUERYING THE DATABASE

Which crops were grown in which countries (the country code) in 2009?

- Observe two facts:
  1. Number of attributes (original: 5)
  2. Number of tuples (original: 6930)



Q3

UNIVERSITY OF TWENTE.

28

QUERYING THE DATABASE
CLOSER LOOK AT **OUTPUT TUPLE DEFINITION**

Re-naming (*aliasing*) is sometimes handy/useful.

```
SELECT p.crop, p.cid
FROM Productions AS p
WHERE p.annum = 2009
```

For attributes only:

```
SELECT attribute1 AS Year,
       attribute2 AS Population
FROM… WHERE…
```

UNIVERSITY OF TWENTE.

29

QUERYING THE DATABASE
CLOSER LOOK AT **SELECTION CONDITION**

- Selection condition determines which of the input tuples are used to generate the output of the query.
- In SQL, the selection condition is represented by the `WHERE` clause:

```
SELECT p.crop, p.cid
FROM Productions AS p
WHERE p.annum = 2009
```

- Selection conditions are formulated as a **logical expression**.
- Logical expressions can be formed by **atomic** or **non-atomic** logical formulas.

UNIVERSITY OF TWENTE.

30

QUERYING THE DATABASE
CLOSER LOOK AT **SELECTION CONDITION**

- ***Atomic formulas*** (conditions) use a predicate symbol like: `=`, `>`, `<`, `≥`, `≤`, `<>`, `is null`, `is not null`
- Examples (of `WHERE` clause) in SQL:
  - `WHERE p.year = 2009`
  - `WHERE p.crop is not null`
  - `WHERE p.score > 1500000`

Logical expressions

UNIVERSITY OF TWENTE.

31

QUERYING THE DATABASE
CLOSER LOOK AT **SELECTION CONDITION**

- ***Non-atomic formulas*** (conditions) are built up from atomic ones, using connectives like: `AND`, `OR`, `NOT`, `()`
- Examples (of `WHERE` clause) in SQL:
  - `WHERE p.crop is not null AND p.score is not null`
  - `WHERE p.annum = 2009 OR p.annum = 2008`
  - `WHERE NOT p.annum = 2009`

UNIVERSITY OF TWENTE.

32

QUERYING THE DATABASE
CLOSER LOOK AT **SELECTION CONDITION**

- If we have the following:
  Logical expression 1 = **p**
  Logical expression 2 = **q**

(p AND q) is correct if and only if p is correct and q is correct
(p OR q) is correct when p is correct, or q is correct or both of them are correct.

Example Question: Select all the records for years 2009 and 2005

Answer : Selection condition would be :

~~Year = 2009 AND Year = 2005~~

Year = 2009 OR Year = 2005 ✓

Be back soon!

JSP QUERY
THE CONCEPT

- One of the query schemes that is used in JSP queries in SQL looks like:

```
SELECT…
FROM table1 AS t1, table2 AS t2
WHERE t1.attrib1=t2.attrib5 AND…
```

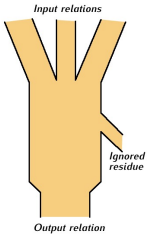
- Semantics of a JSP query is based on the **Cartesian product** of the relations involved in the query – e.g. `table1 × table2`
- After calculating the Cartesian product of relations involved in a query, **tuple selection** and **attribute projection** are performed.

JSP QUERY
THE CONCEPT

JOIN
- Two or more input relations are joined (using join condition)

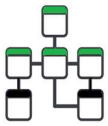
SELECT
- Joined tuples are filtered through tuple selection (using selection condition)

PROJECT
- Attributes are projected to the output relation (using attribute projection)



MULTI-RELATION QUERY
RATIONALE

- Single-input relation queries were discussed so far.
- Very often, to extract required data from a database, we need to combine data from **two** or **more** relations in one query.
  (*E.g. remember country names and country id were in two different tables in our FAOcrops database?*)
- In such queries, selection conditions depend on the **relationships between relations** that are involved in the query.



MULTI-RELATION QUERY
THE CONCEPT

- Multi-relation query is also called a 'join query' or a 'JSP query'
- The abbreviation JSP stands for:

JOIN
- Join condition

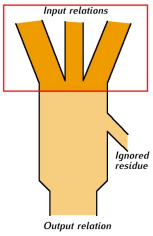
SELECT
- Tuple selection

PROJECT
- Attribute projection

JSP QUERY
THE CONCEPT

Phases of JSP query:

1. Cartesian product of input relations
   - ***Which relations are involved in a query?***



JSP QUERY
THE CONCEPT – CARTESIAN PRODUCT

- **Cartesian product** of `table1` and `table2` (`table1 ×table2`) is a relation obtained from `table1` and `table2` by concatenating ('gluing') any tuple of `table1` with any tuple of `table2`.
- Every resulting tuple has all the attributes of `table1` and all those of `table2`.

A	B
700	a
300	b
150	c

`Table1`

X	Y	Z
m	100	s
n	200	a

`Table2`

A	B	X	Y	Z
700	a	m	100	s
700	a	n	200	a
300	b	m	100	s
300	b	n	200	a
150	c	m	100	s
150	c	n	200	a

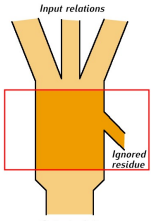
`Table1 × Table2`

25/09/2023

JSP QUERY
THE CONCEPT

Phases of JSP query:

1. Cartesian product of input relations
   - *Which relations are involved in a query?*
2. Tuple selection:
   - ***What is the join condition?***
   - ***What is the selection condition?***

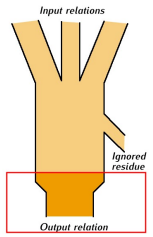


41

JSP QUERY
THE CONCEPT

Phases of JSP query:

1. Cartesian product of input relations
   - *Which relations are involved in a query?*
2. Tuple selection
   - *What is the join condition?*
   - *What is the selection condition?*
3. Output tuple creation
   - ***What attributes will the output relation have?***



42

JSP QUERY
SIMPLE EXAMPLE – from FAOcrops

In SQL:

```
SELECT c.Cname, p.crop, p.score, p.annum AS year
FROM Productions AS p, Countries AS c
WHERE p.cid=c.ID AND c.CNAME='France' AND p.annum=2009
```

And the output:

Query1

CNAME	crop	score	year
France	Maize	15299900	2009
France	Potatoes	7164200	2009
France	Rice, paddy	138100	2009

from `Countries` table — from `Productions` table

45

JSP QUERY
SIMPLE EXAMPLE – from FAOcrops

This would be the result if we have had selected all the attributes from the Cartesian product:

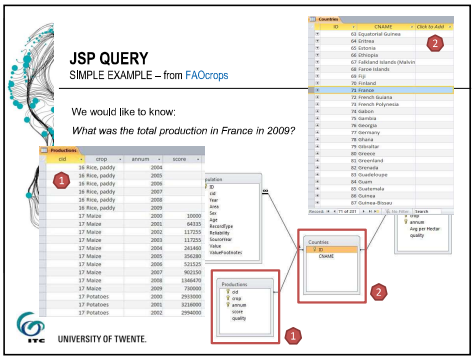
ID	CNAME	cid	crop	annum	score	quality
71	France	71	Maize	2009	15299900	
71	France	71	Potatoes	2009	7164200	
71	France	71	Rice, paddy	2009	138100	

In SQL:

```
SELECT *
FROM Productions AS p, Countries AS c
WHERE p.cid=c.ID AND c.CNAME='France' AND p.annum=2009
```


46

25/09/2023



JSP QUERY
SIMPLE EXAMPLE – from FAOcrops

We would like to know:

What was the total production in France in 2009?

43

JSP QUERY
SIMPLE EXAMPLE – from FAOcrops.mdb

In SQL:

```
SELECT c.CNAME, p.crop, p.score, p.annum AS Year
FROM Productions AS p, Countries AS c
WHERE p.cid=c.ID AND c.CNAME='France' AND p.annnum=2009
```

join condition

- Join condition is a special type of selection condition (content of the `WHERE` clause) – because `p.cid` and `c.ID` are from different relations
- Join condition is a subset of the Cartesian product of `Productions` × `Countries`

44

25/09/2023

OTHER JOINS
INNER JOIN

- The example shown earlier, is called an EQUI-JOIN – because of the "=" in the join condition (`p.cid=c.ID`). This type of JSP query is the most common.
- An alternative way of writing this query, makes use of explicit join statements in the `FROM` clause of the SQL query. This alternative is referred to as INNER JOIN and the query format is:

```
SELECT…
FROM table1 AS t1 INNER JOIN table2 AS t2
ON t1.attribute_A=t2.attribute_B
WHERE…
```

- Typically, the `attribute_A` is the foreign key of `table1` and `attribute_B` is a primary key of `table2` (or *vice versa*)

47

OTHER JOINS
INNER JOIN

Back to the example:

```
SELECT c.CNAME, p.crop, p.score, p.annum
FROM Productions AS p
     INNER JOIN Countries AS c ON p.cid=c.ID
WHERE c.CNAME='France' AND p.annum=2009
```

The output is:

Query1

Cname	crop	score	annum
France	Maize	15299900	2009
France	Potatoes	7164200	2009
France	Rice, paddy	138100	2009

48

OTHER JOINS
CLOSER LOOK AT THE JOIN CONDITION

- The comparison operator:
  - "=" is by far the most common comparison operator in join conditions, but
  - Any of the following operators can also be used:
    `<>`, `>`, `>=`, `<`, `<=`, `BETWEEN`, or `LIKE`
- Number of join conditions in more than two relations involved in a query – general rule:

N input relations → (*N-1*) join conditions:

UNIVERSITY OF TWENTE.

49

OTHER JOINS
CLOSER LOOK AT JOIN CONDITION

3 tables join in SQL:

```
SELECT…
FROM table1 AS t1, table2 AS t2, table3 AS t3
WHERE t1.attr1=t2.attr2 AND t2.attr2=t3.attr3
```

Or:

```
SELECT…
FROM (table1 AS t1 INNER JOIN table2 AS t2 ON
t1.attr1=t2.attr2) INNER JOIN table3 AS t3 ON
t2.attr2=t3.attr3
WHERE…
```

UNIVERSITY OF TWENTE.

50

OTHER JOINS
OUTER JOINs

The INNER JOIN is a common operator in database querying, but it is not the only possible type of JOIN.

There are other options known as **OUTER JOINs**:

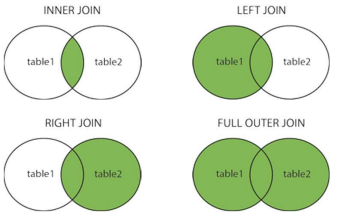
- **LEFT JOIN**
- **RIGHT JOIN**
- **FULL JOIN**

} OUTER JOINs

UNIVERSITY OF TWENTE.

51

DIFFERENT TYPES OF SQL JOINs



UNIVERSITY OF TWENTE.

https://www.w3schools.com/sql/sql_join.asp

52

SPATIAL QUERIES
BRIEF INTRODUCTION

- In a GIS or SDBMS:

Built in functions based on topological rules and standards.

Spatial Selections

```
SELECT f.*
FROM Factories AS f, Railway AS r
WHERE ST_DISTANCE (f.geom,r.geom) < 200
```

Spatial Join

```
SELECT r.name, r.length, s.name, s.area
FROM Rivers AS r, Regions AS S
WHERE ST_INTERSECTS(r.geom,s.geom)
```

UNIVERSITY OF TWENTE.

53

SPATIAL QUERIES
BASIC SPATIAL FUNCTIONS

Location

ST_X — Returns the X coordinate of a Point.
ST_Y — Returns the Y coordinate of a Point.

```
SELECT ST_Y(geom)
FROM vector.bi_settlements
WHERE gid='308'
```

Distance

ST_Distance — Returns the distance between two geometry or geography values.

```
SELECT ST_Distance((a.geom),(b.geom))
FROM vector.bi_settlements a, vector.bi_settlements b
WHERE a.gid='300' AND b.gid='302'
```

Length/Area

ST_Area — Returns the area of a polygonal geometry.

ST_Length — Returns the 2D length of a linear geometry.

UNIVERSITY OF TWENTE.

54

SPATIAL QUERIES
TOPOLOGY FUNCTIONS

ST_Contains(geometry A, geometry B): no points of B lie in the exterior of A, and at least one point of the interior of B lies in the interior of A.
ST_Crosses(geometry A, geometry B): Geometries have some, but not all, interior points in common.
ST_Disjoint(geometry A , geometry B): Geometries do not share any space together.
ST_DWithin(geometry A, geometry B, radius): geometries are within the specified distance (radius) of one another.
ST_Intersects(geometry A, geometry B): Geometries share a portion of space
ST_Overlaps(geometry A, geometry B): Geometries share space but are not completely contained by each other.
ST_Touches(geometry A, geometry B): Geometries have at least one point in common, but their interiors do not intersect.
ST_Within(geometry A , geometry B): Geometry A is completely inside geometry B

```
select a.name_com, a.geom, b.geom
from vector.bi_communes a join vector.bi_provinces b
on ST_Contains (b.geom, a.geom)
where b.name_prov = 'Gitega'
```

UNIVERSITY OF TWENTE.

55

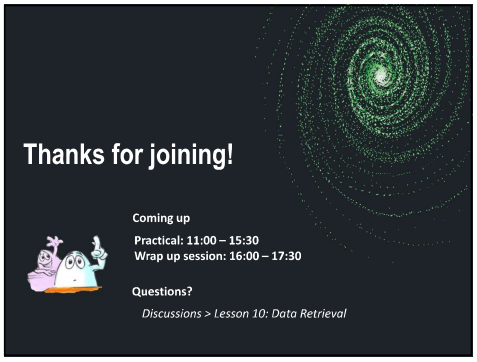
SPATIAL QUERIES
OVERLAY FUNCTIONS

ST_Difference — Returns a geometry representing the part of geometry A that does not intersect geometry B.
ST_Intersection — Returns a geometry representing the shared portion of geometries A and B.
ST_SymDifference — Returns a geometry representing the portions of geometries A and B that do not intersect.
ST_Union — Returns a geometry representing the point-set union of the input geometries.

```
select ST_Union (a.geom, b.geom)
from data.table_1 a, data.table_2 b
```

UNIVERSITY OF TWENTE.

56


Thanks for joining!
Coming up
Practical: 11:00 – 15:30
Wrap up session: 16:00 – 17:30
Questions?
Discussions > Lesson 10: Data Retrieval

57