[Please select]
[Please select]

EUROPEAN CENTRAL BANK

EUROSYSTEM

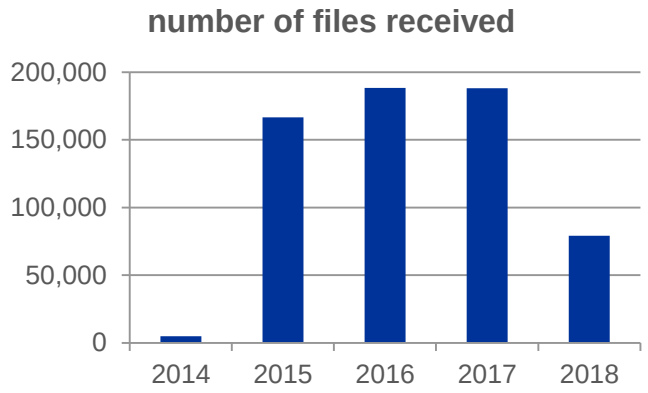
Matthias Pécot
ESCB/IO expert
Statistical Applications Section

Analysing supervisory data using Hadoop

Eurofiling XBRL Week 2018

SUBA: some numbers

- Number of files received by SUBA:
  - 166,700 files in 2015
  - 188,300 files in 2016
  - 188,000 files in 2017



- remitted by 5200 entities. 2018 is ongoing.

- Right now, there are more than 800 million data points in SUBA fact table (named OBSERVATIONS).

- When extracted in .csv file format, the bare table is more then 150GB.

SUBA: considerations

- Data is stored in an Oracle database, optimized for transactional usage.
- Interactive querying is not really possible. Moreover, this is a production database, so not suitable either.
- To be human readable, the OBSERVATIONS table has to be joined with many other tables containing labels/infos on entities, modules, data points, cells, tables…

DISC

- Since recently, at the ECB, DISC project offers access to a Cloudera Hadoop cluster. Right now are available:
  - *Hdfs*
  - *Hive*
  - *Impala*
  - *Pig*
  - *Oozie*
  - *Spark (announced)*

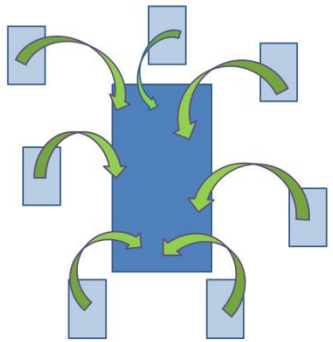
POC SUBA data on Hadoop

- Proof Of Concept: ongoing, not in production at the moment

- The goal is to:
  - *enable interactive querying on SUBA data*
  - *provide easy data visualization*
  - *assess possibilities and performance on DISC*
  - *collect best practices / useful tips*
  - *answer the question: how to best represent SUBA data in DISC?*

- Difficulties:
  - *Impala performs poorly on multi-join queries*
  - *SUBA data model is quite complex (similar to the DPM, more than 100 tables) and requires sometimes complicated queries.*

POC SUBA data on Hadoop



- Solution: denormalize data!
- By inserting into the fact table the data related to its foreign keys.
- In this way, we take advantage from data locality. No more joins: when accessing a fact, the relevant infos on entities, files, … are stored on the same line of the table.
- Data is stored into a Parquet file, using Hive.
- Impala is used to query it.

POC SUBA data on Hadoop: denormalized table

entity_id	string
entity_attributes_struct	struct<waivcompind_sec:string, finrep_solo:string, ulssmparentcou:string, ulssmparent_lei:string, ...>
value_txt	string
value_decimal	float
value_boolean	string
value_date	string
tid_received_modules	int
received_modules_attributes_struct	struct<vr_status:string, acceptance_status:string, dpt_status:string, dpt_comments:int, ...>
variable_id	string
precision	int
unit	string
dsd_id	string
tid_members	int
reported_period	string
reception_date	string
data_point_id	string
is_shaded	string
cells_array	array<struct<table_id:string, table_name:string, tid_cell:double, ...>>
taxonomy_code	string
taxonomy_name	string
taxonomy_basepath	string
taxonomy_dpm_template	string
taxonomy_path	string
taxonomy_last_version	string
taxonomy_item_name	string
taxonomy_item_from_date	string
taxonomy_item_to_date	string

Impala is fast on single table queries

- verifying the unicity of the primary key in SUBA fact table (800 millions lines)

```
select
   count(*), entity_id, variable_id, tid_received_modules, reported_period
from OBSERVATIONS
group by entity_id, variable_id, tid_received_modules, reported_period
having count(*)>1
```

returns 0 line in under 4mn with Impala (16mn with Hive, does not return in Oracle after 1 hour).

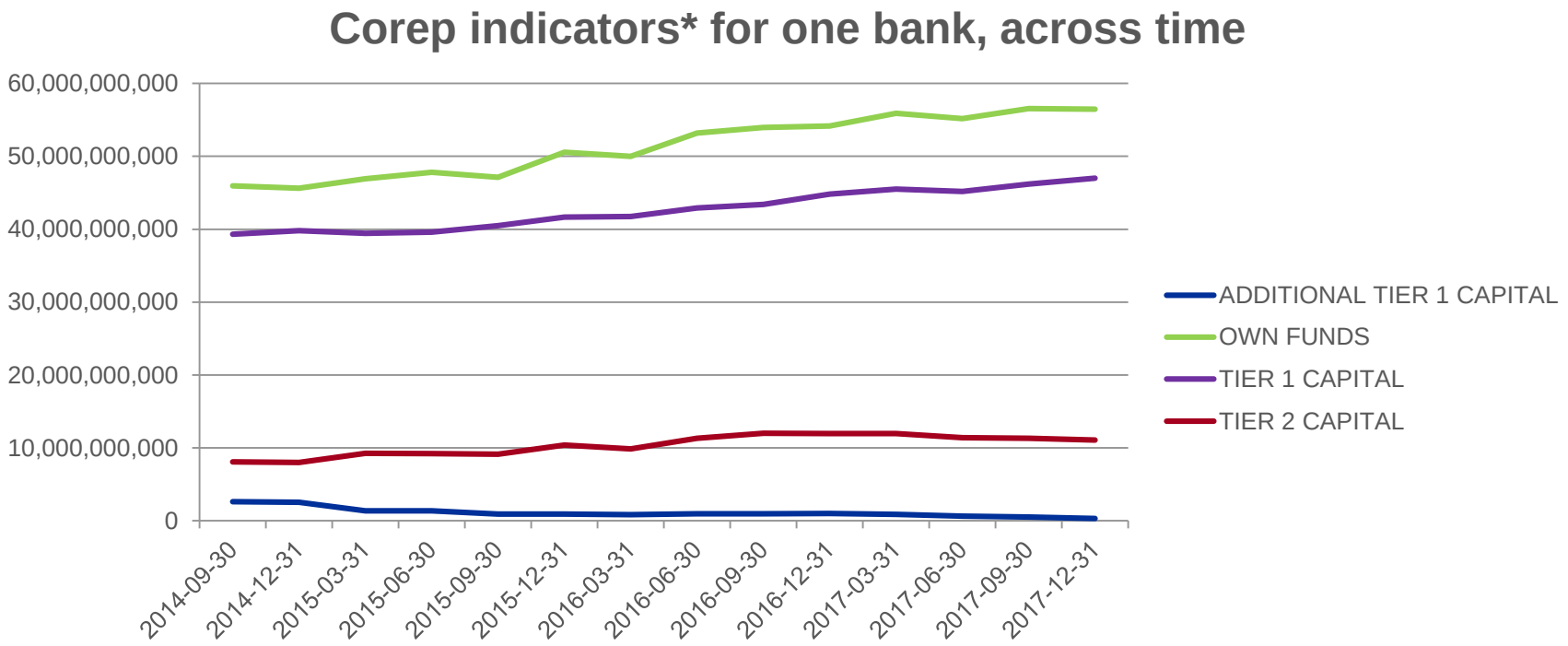
- counting the number of distinct entities:

```
select count(*), count(distinct entity_id) from OBSERVATIONS
```

return results in 7s with Impala (11mn with Hive , does not return in Oracle after 1 hour)

Displaying Corep indicators*

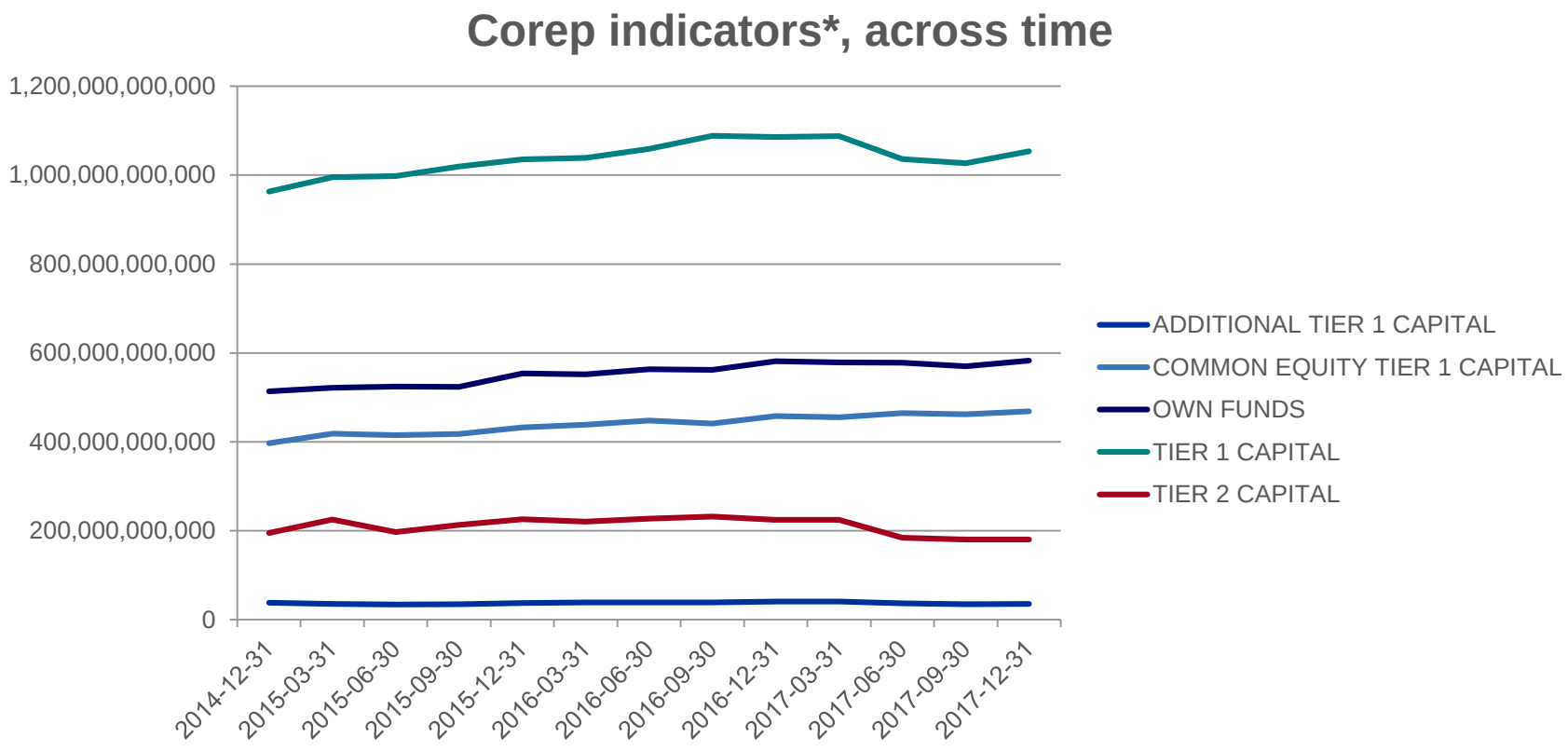
- It takes only 15 seconds to extract some facts for a precise entity. For example, here is the evolution of some capital indicators* from COREP (table C 01.00), for one bank:



**** Amounts and ratios have been randomized***

Displaying Corep indicators*

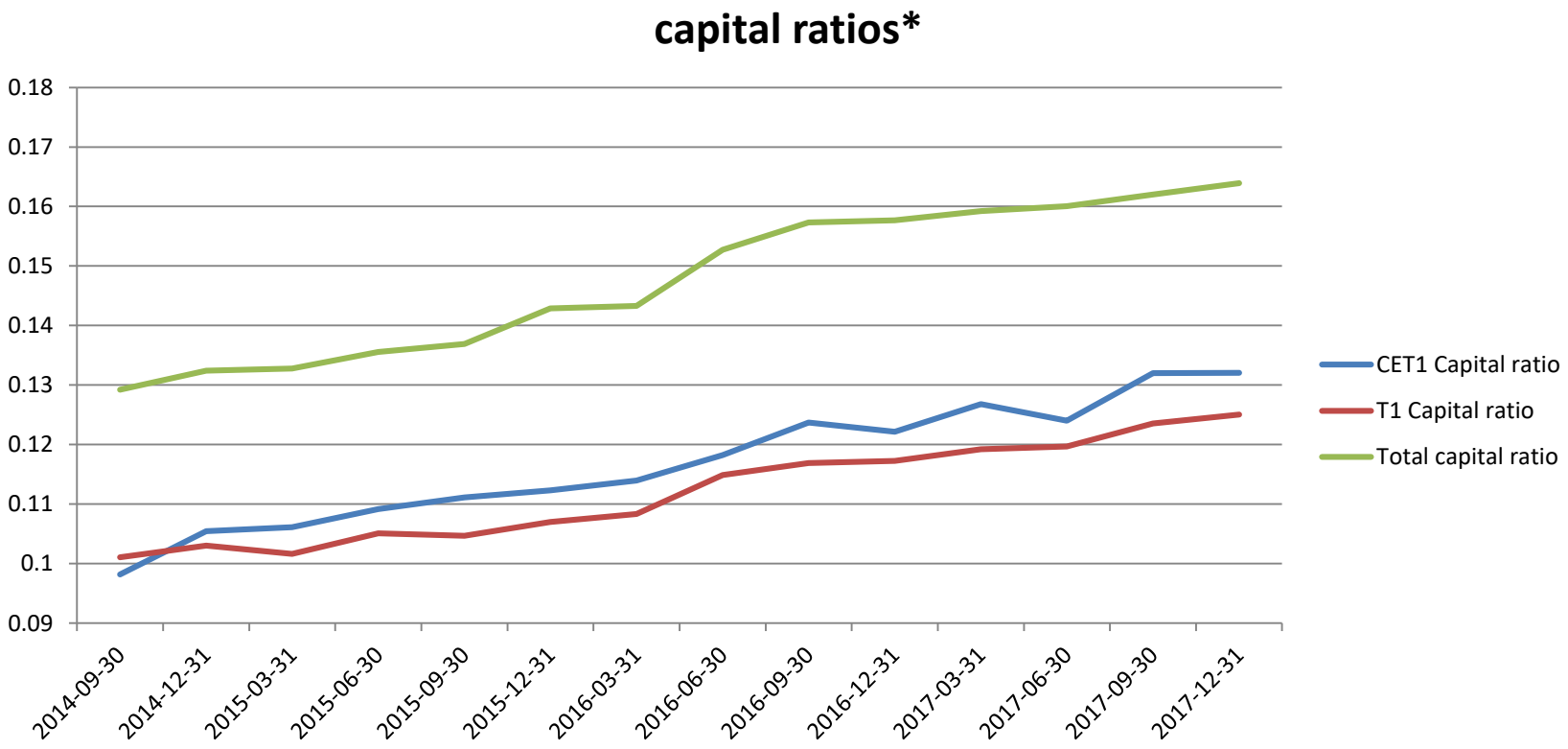
- Here is the same query, summing the indicators* for all entities of a given country. The query does not take any longer to compute:



**** Amounts and ratios have been randomized***

Displaying Corep indicators*

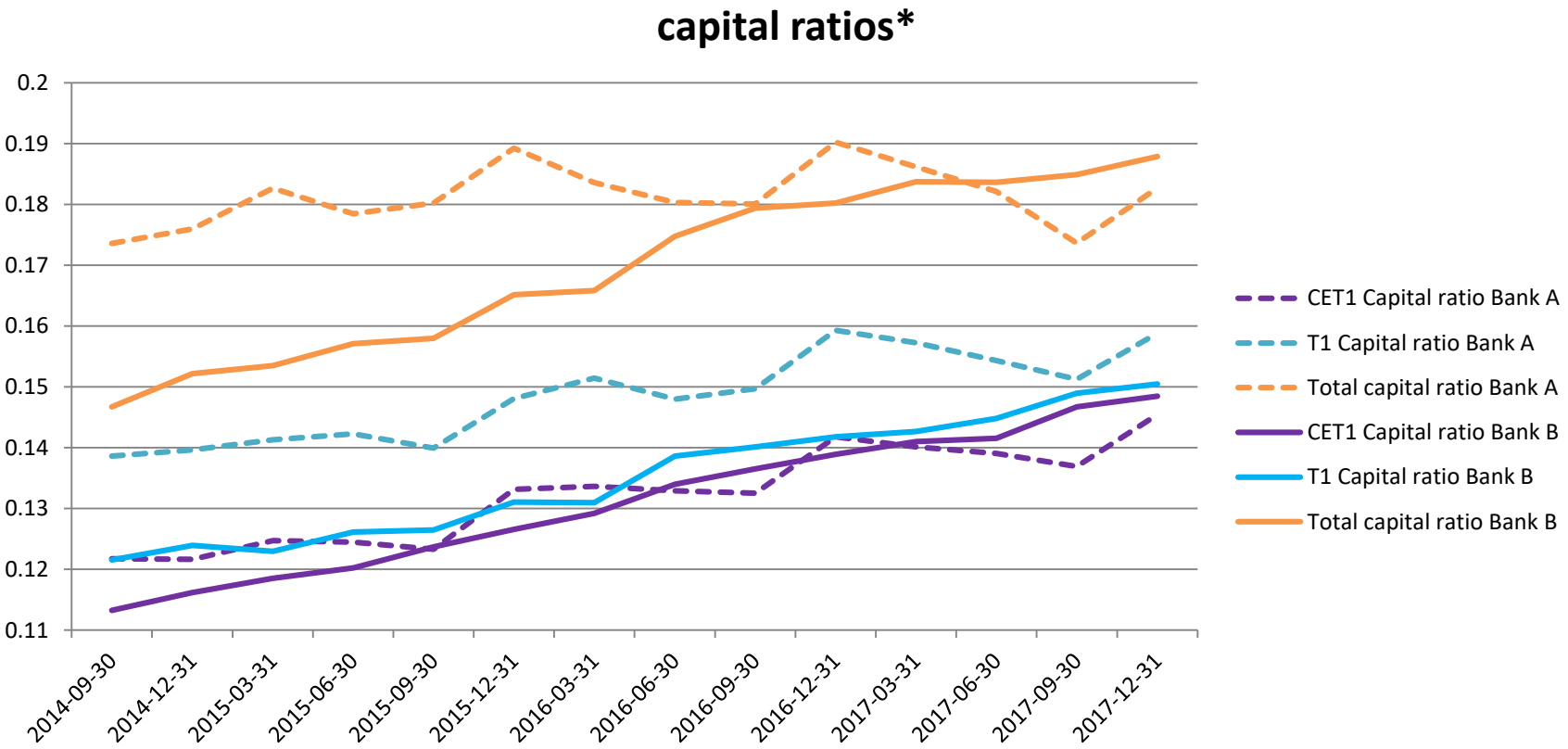
- Capital ratios* of a bank through time. The query returns in 20 seconds:



**** Amounts and ratios have been randomized***

Displaying Corep indicators*

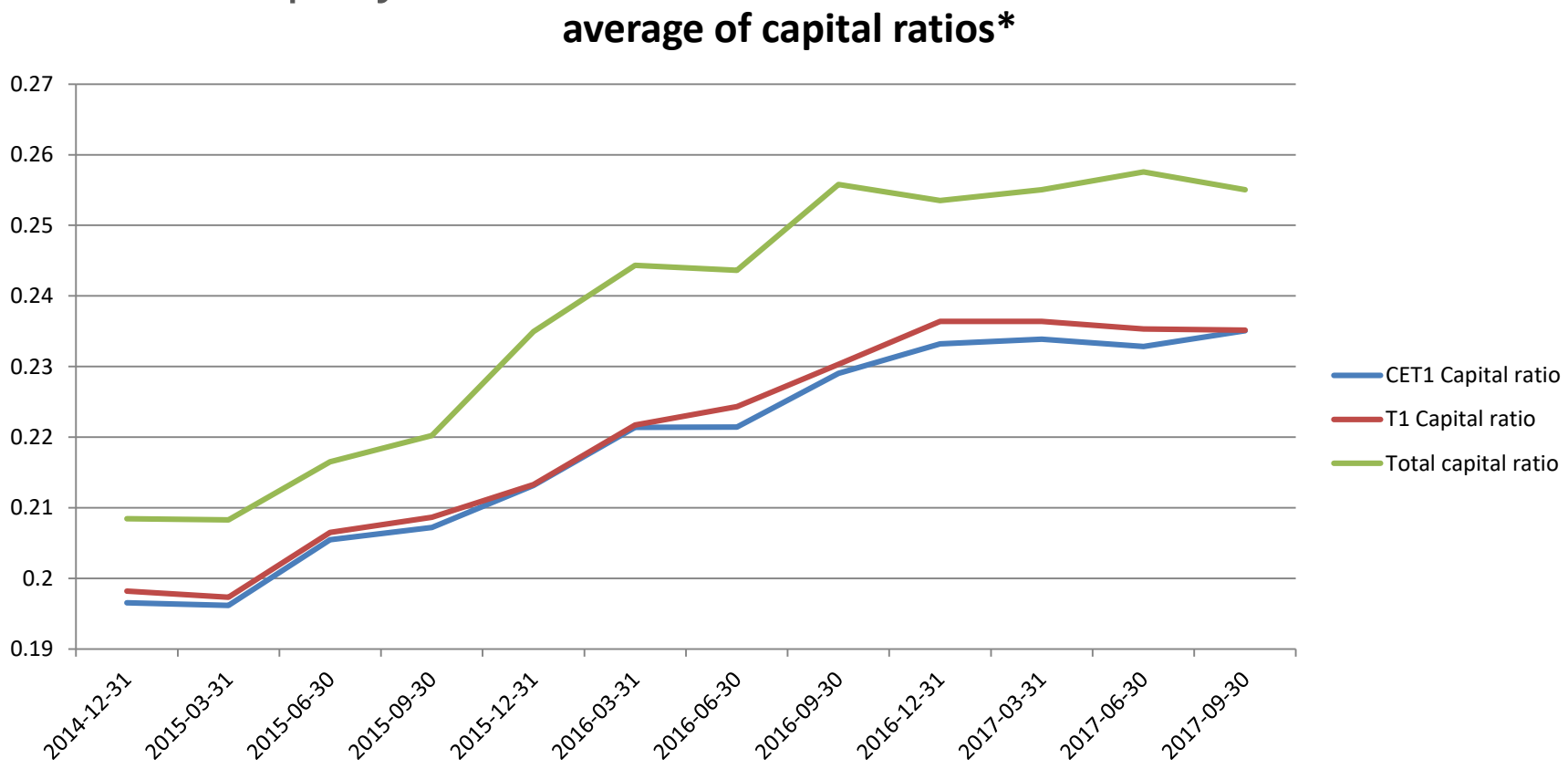
- Comparing capital ratios* of two banks through time. The query returns in 25 seconds:



**** Amounts and ratios have been randomized***

Displaying Corep indicators*

- Average (not weighted) of capital ratios* of 100 banks through time. The query returns in 20 seconds :



**** Amounts and ratios have been randomized***

Extracting data* to Excel

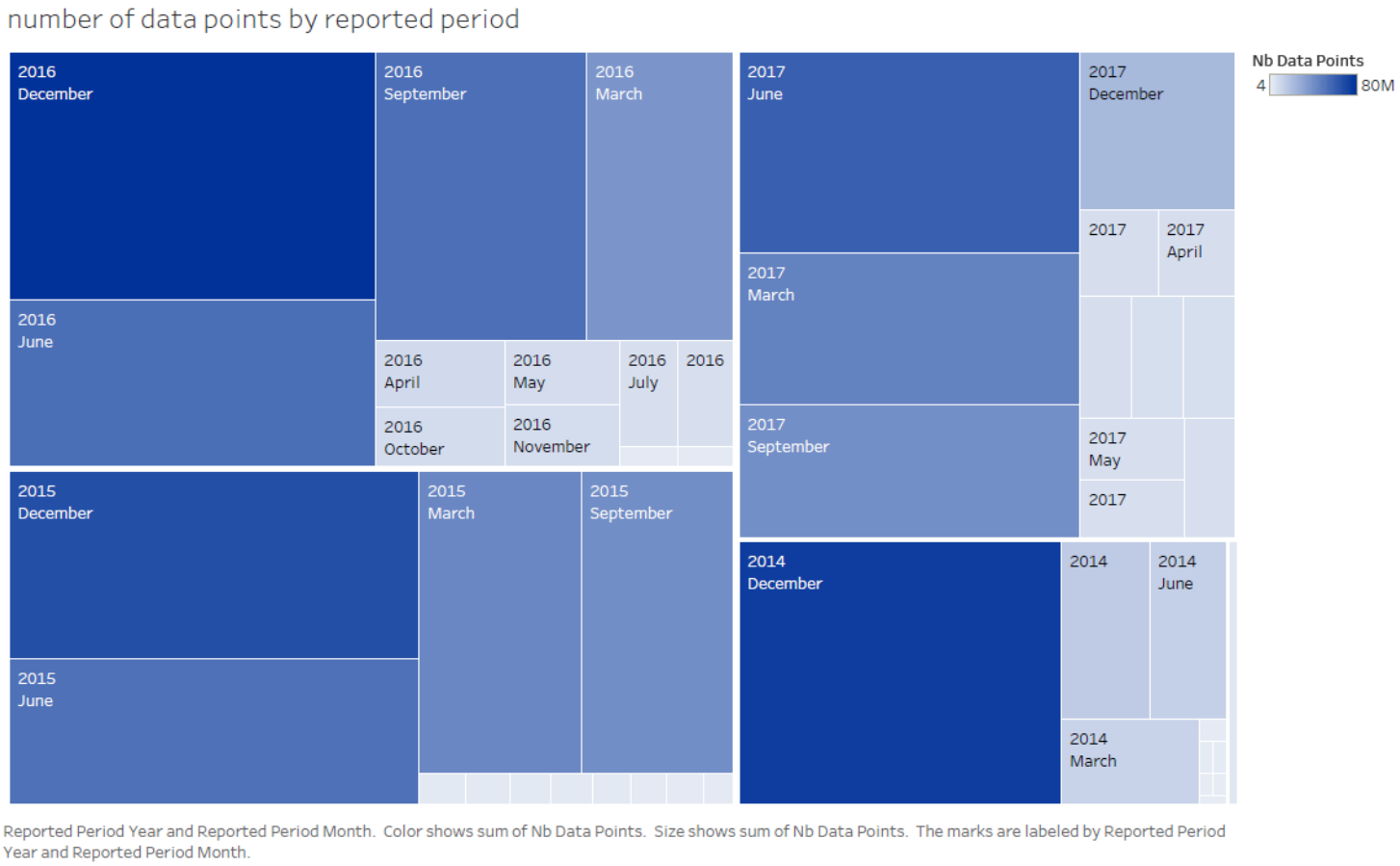
- It is easy to extract data*, with Excel PivotTable though ODBC:

Row Labels	010 OWN FUNDS	015 TIER 1 CAPITAL	530 ADDITIONAL TIER 1 CAPITAL	750 TIER 2 CAPITAL
010				
Amount				
2014-03-31	81,061,591,437	147,739,942,328	6,503,632,350	46,241,954,249
2014-06-30	85,124,736,159	151,707,668,137	8,191,692,960	46,242,382,563
2014-09-30	85,279,140,332	156,340,424,373	9,161,282,514	48,488,287,025
2014-12-31	86,276,756,062	157,927,567,868	9,507,254,135	48,323,459,758
2015-03-31	92,912,997,658	165,459,862,323	9,562,045,311	55,017,394,744
2015-06-30	89,235,391,240	165,610,303,254	8,758,373,717	50,328,546,953
2015-09-30	91,492,381,437	166,929,724,046	8,872,665,263	51,302,613,822
2015-12-31	96,068,156,009	174,374,207,830	8,650,467,556	52,578,603,845
2016-03-31	94,484,028,893	172,669,191,726	9,015,119,815	48,856,185,628
2016-06-30	93,335,237,144	175,554,786,133	8,836,670,228	48,032,198,525
2016-09-30	95,037,649,770	196,739,577,993	8,466,301,507	48,923,857,301
2016-12-31	99,203,197,745	204,525,357,188	8,810,902,131	49,348,603,442
2017-03-31	98,081,103,933	204,687,511,990	9,108,510,626	47,113,165,924
2017-06-30	95,023,698,672	162,342,078,200	7,514,159,906	27,697,499,384
2017-09-30	91,438,234,402	155,298,029,787	6,837,244,133	26,164,686,161
2017-12-31	95,786,421,548	163,531,233,175	6,772,193,390	25,266,883,120

**** Amounts and ratios have been randomized***

Data visualization

- Here are some charts created using Tableau, connecting to Impala though ODBC and querying the fact table:



Data visualization

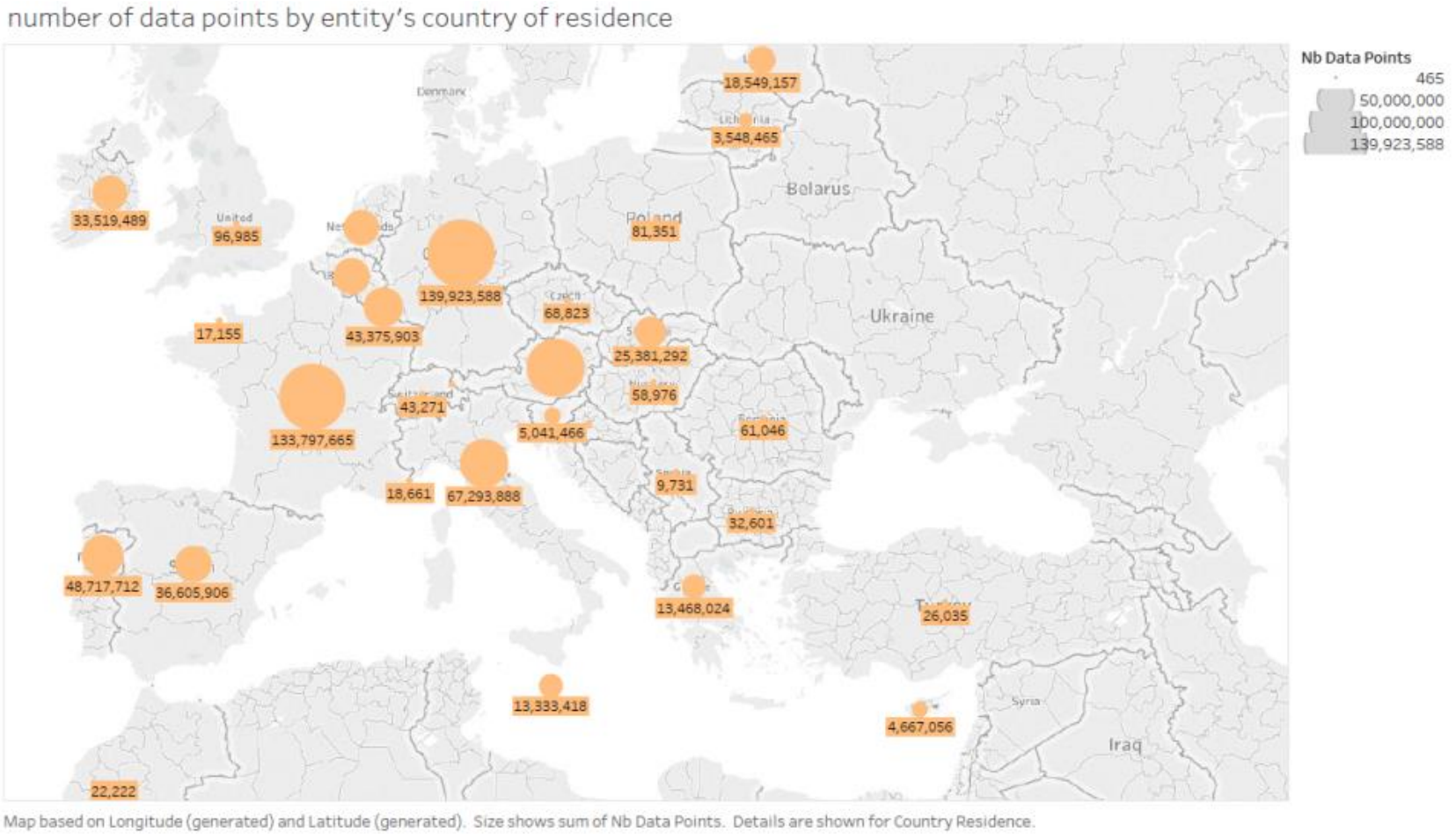
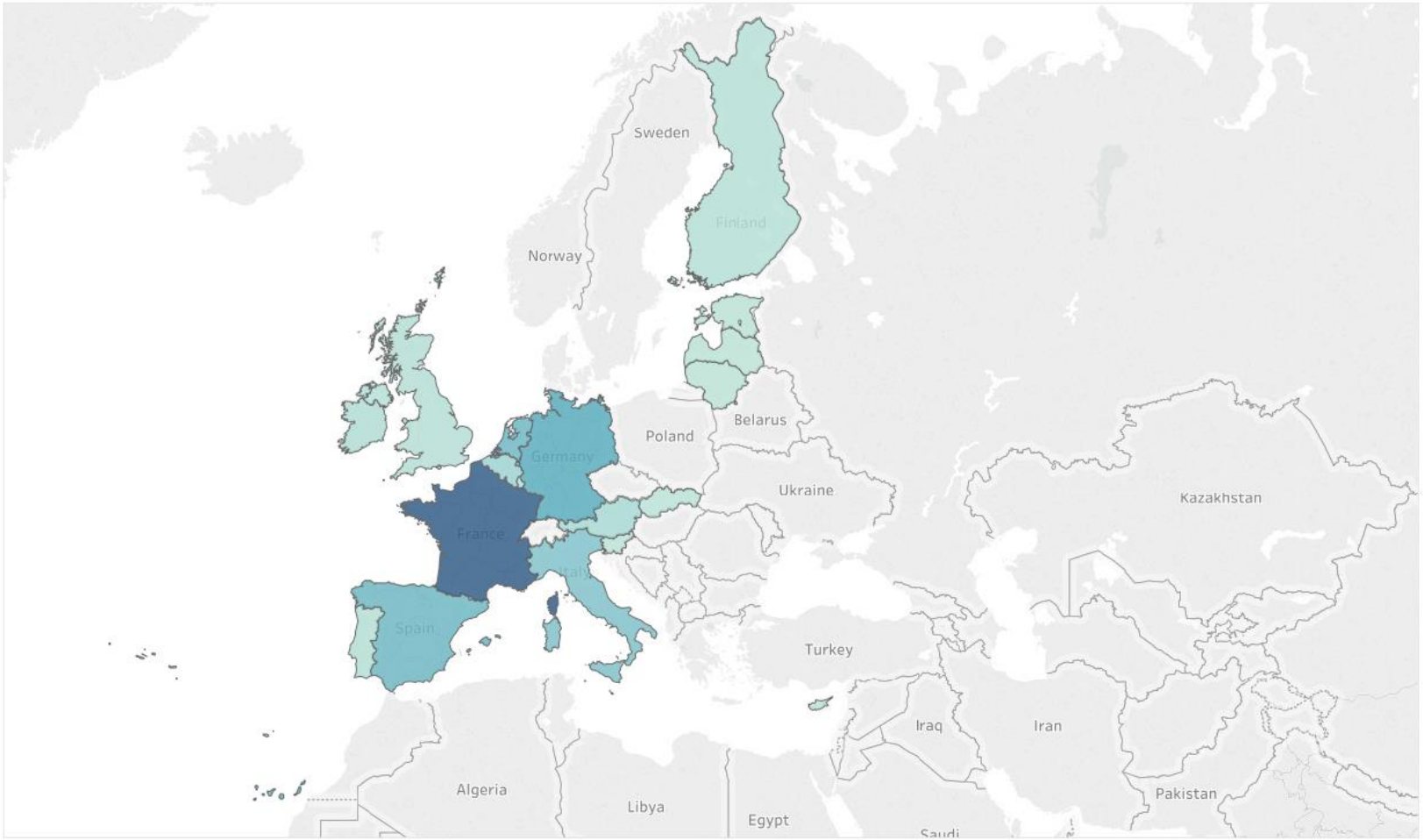


Tableau: sum of own funds (Corep C 01.00) by country in September 2017

- The query returns in 30 seconds in Tableau:



Conclusions

- Not possible/advisable to copy all tables from a model into Hadoop
- Only copy the fact table, enriched.
- Interactive querying is possible, using Impala on the denormalized table.
- Performances are very good when querying only one table.
- The process of denormalizing the fact table is quite intricate, because the underlying model is complex.
- The final fact table has one fact per line, it should be possible to use a more tabular format:
  - Easier to query
  - More difficult to construct