



# VENDÉE GLOBE

EN SOLITAIRE | SANS ESCALE | SANS ASSISTANCE

**DÉPART  
LE 8.11.20**

 VENDÉE

 Sodeho

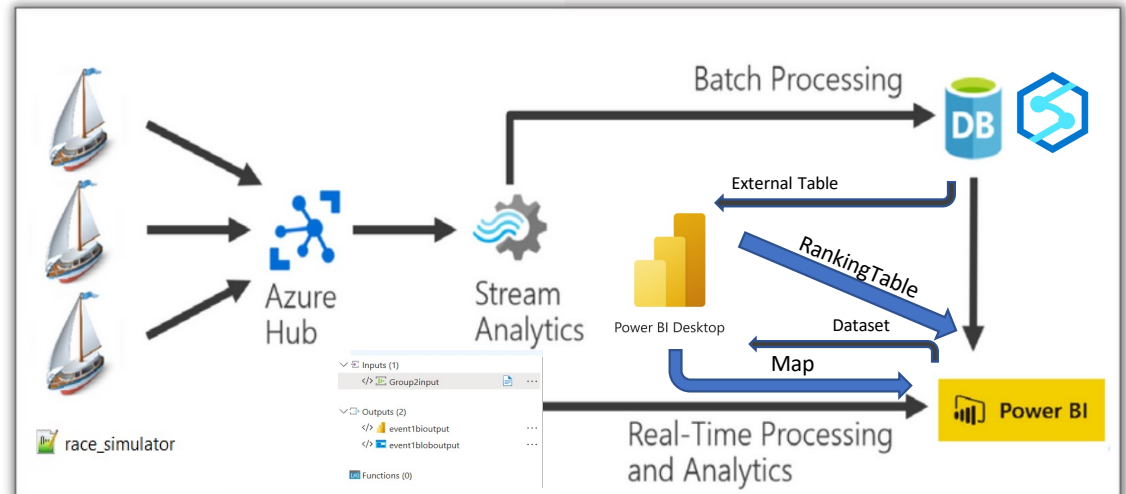
# Vendée Globe

## World Sailing Race 2020 – 2021

ROADMAP TO  
CREATING LAMBDA  
ARCHITECTURE  
STREAMMING DATA

### Business Case - The Challenge

Build a cloud-based Lambda Architecture to process the telemetry data from the sailing boats. Your architecture should run in Azure and contain a real-time path for processing sailing boat data in real-time, and a batch-processing path for collecting sailing boat data in batches and performing calculations on those batches.



# CHALLENGE 1: Removing Wrong Data

```
Boat: 0 Lat: 38.630708245161266 Long: -9.390403598572513 Headin
Boat: 1 Lat: 38.62126856059126 Long: -9.40120968639585 Heading:
Boat: 2 Lat: 38.628383103198026 Long: -9.385779110164174 Headin
Boat: 3 Lat: 38.6238944017545 Long: -9.395942918795413 Heading:
Boat: 4 Lat: 38.61726739948685 Long: -9.403844106686837 Heading
Boat: 5 Lat: 38.62534522430175 Long: -9.401123337978204 Heading
Boat: 6 Lat: 38.61795097913035 Long: -9.394103460033676 Heading
Boat: 7 Lat: 38.61432341311072 Long: -9.401796496356871 Heading
Boat: 8 Lat: 38.6293091666482 Long: -9.399453177516785 Heading:
Boat: 9 Lat: 38.61981723723841 Long: -9.391681451265836 Heading
Boat: 9 Lat: 38.61981723723841 Long: -9.391681451265836 Heading
Boat: 0 Lat: 38.628988065719234 Long: -9.392026780785999 Headin
Boat: 1 Lat: 38.620226938791504 Long: -9.402307977777548 Headin
Boat: 2 Lat: 38.62773602274056 Long: -9.38667495010637 Heading:
Boat: 3 Lat: -10000 Long: -10000 Heading: 216.2652525151492 Spe
Boat: 4 Lat: 38.61569530218923 Long: -9.405751229813495 Heading
Boat: 5 Lat: 38.623750354419215 Long: -9.402957485933316 Headin

Boat: 6 Lat: 38.616883439954144 Long: -9.396498752352212 Headin

Boat: 7 Lat: 38.613433868765526 Long: -9.403002295598952 Headin
Boat: 8 Lat: 38.628659308227164 Long: -9.400217037573247 Headin
Boat: 9 Lat: 38.61871320871621 Long: -9.393021521989374 Heading
Boat: 9 Lat: 38.61871320871621 Long: -9.393021521989374 Heading
Boat: 0 Lat: 38.627276904700345 Long: -9.393439860839376 Headin

Boat: 1 Lat: 38.61927368921026 Long: -9.403319414817016 Heading
```

Latitude & Longitude checks.

Discard data if out of range

```
SELECT *
INTO
event1bloboutput
FROM
Group2input
where (latitude>=-
90 and latitude<=90) and (longitude>=-
180 and longitude<=180)
```

```
SELECT *
INTO
event1bioutput
FROM
Group2input
where (latitude>=-
90 and latitude<=90) and (longitude>=-
180 and longitude<=180)
```



# CHALLENGE 2: Ranking Calculation

## Logic Building



```
# force boat to move south-west from Cascais
if (boat.Latitude > 15 and (boat.Heading < 205 or boat.Heading > 245)):
    boat.Heading = 225

# force boat to move south-east once it passes latitude 15 to move past Africa and South America
if (boat.Latitude > -50 and boat.Latitude <= 15 and (boat.Heading < 130 or boat.Heading > 170)):
    boat.Heading = 150

# once the boat is close to Antarctica, keep heading east towards Australia
if (boat.Latitude > -64 and boat.Latitude <= -50 and (boat.Heading < 70 or boat.Heading > 110)):
    boat.Heading = 90
```

Race Path used for calculation

Used Latitude & Longitude to calculate Rank

**Rationale:** Ranking should be determined by position of boat

**How:** Calculated as per the racing path

1st path: Boats travel South till latitude of -50. Latitude is decreasing so boat with minimum latitude is leading.

2nd path: Boats go east after reaching latitude of -50. Longitudes start increasing. Boat with highest longitude is leading.

# CHALLENGE 2: Ranking Calculation

## First Attempt: Python Implementation

```
In [10]: import pandas as pd

raw_data={'boat_id':[0,1,2,3,4,5,6,7,8,9], 'latitude':[40,10,-56,-60,-52,-20,-62,0,-55,-51],
          'longitude':[2,5,8,2,10,20,5,3,25,6]}

df=pd.DataFrame(raw_data)

df

df1=df[df['latitude']>-50]

df1.sort_values(by='latitude', inplace=True)

print(df1)

df2=df[df['latitude']<=-50]

df2.sort_values(by='longitude', ascending=False, inplace=True)

print(df2)

df3= df2.append(df1)

df3=df3.reset_index()

df3=df3.reset_index()

print(df3)

df3['rank']=df3['level_0']+1

df3
```

```
File Edit View Insert Cell Kernel Widgets Help
df3= df2.append(df1)
df3= df2.append(df1)

Out[10]:
```

	level_0	index	boat_id	latitude	longitude	rank	
	0	0	8	8	-55	25	1
Crosses-50:	1	1	4	4	-52	10	2
	2	2	2	2	-56	8	3
Ranking by	3	3	9	9	-51	6	4
Longitude	4	4	6	6	-62	5	5
	5	5	3	3	-60	2	6
Above -50:	6	6	5	5	-20	20	7
	7	7	7	7	0	3	8
Ranking by	8	8	1	1	10	5	9
Latitude	9	9	0	0	40	2	10

# CHALLENGE 2: Ranking Calculation

Final Calculation: Created query on SQL serverless pool on Azure Synapse Analytics to test it

```
Boat: 7 Lat: 38.58525627747327 Long: -9.43228725015361 H
Boat: 8 Lat: 38.59036505555702 Long: -9.4388482307643 He
Boat: 9 Lat: 38.584730321153614 Long: -9.433540428930971
Boat: 9 Lat: 38.584730321153614 Long: -9.433540428930971
Boat: 0 Lat: 38.58202359998281 Long: -9.468336796893402
```

The screenshot displays the Azure Synapse Analytics SQL editor interface. The query is a multi-part SQL statement designed to rank boats based on their corner, latitude, and longitude. The query is as follows:

```
1 select boat, latitude, longitude, corner,
2 (Rank() over(order by corner desc, c1)) as final_rank
3 FROM
4 (
5     SELECT
6     boat, latitude, longitude,
7     (case
8     when latitude<=-50.0 then (Rank() over( PARTITION by corner order by longitude desc))
9     else (Rank() over(PARTITION by corner order by latitude asc))
10    END) as c1
11    , corner
12 FROM
13 (
14     SELECT top 10 max((IIF (latitude>-50.0,0,1))) as corner, boat, min(latitude) as latitude,
15     max(longitude) as longitude, max(EventEnqueuedUtcTime) as EventEnqueuedUtcTime
16 FROM
17     OPENROWSET(
18     BULK 'https://lambdastorage1.blob.core.windows.net/blobcontainer1/sailingdata/year=2023/mont
19     FORMAT = 'PARQUET'
20 ) as rows
21 group by boat
22 order by EventEnqueuedUtcTime desc
23 ) as sub1
24 ) as sub2
```

The results pane shows a table with the following columns: boat, latitude, longitude, corner, and final\_rank. The data is as follows:

boat	latitude	longitude	corner	final_rank
8	38.0913246833492	-9.3951963115...	0	1
4	38.088686263089	-9.3962742971...	0	2
7	38.1709652366479	-9.4006709397...	0	3
6	38.2052391285217	-9.3917970245...	0	4
0	38.2196901053953	-9.3888393826...	0	5
3	38.2719460903217	-9.3845676078...	0	6
5	38.3433631971641	-9.3909616371...	0	7
2	38.3849291155278	-9.3849291155278	0	8

Why SQL: Cheaper option than Apache Spark pool

Sub Query 1: Created a new column "corner" to partition the boats into 2 parts: race South and race East

Sub Query 2: Ranked the 2 parts based on latitude and longitude respectively

Outer Query: To find combined rank of all 10 boats

Handle Duplicates: Used group by boat to remove duplicate values of boat if they occur.

# CHALLENGE 3: Batch Processing

- Second Output to Azure Blob storage which can be accessed from Synapse Analytics .
- Updated Blob storage to Azure Data Lake Gen 2
- Saved in Parquet format which saved data in batches of minimum 10 rows
- External table on folder containing all parquet files

Name	Last Modified	Content Type	Size
-426825278_000001e2234468ea323268435a3d8a2_1.parquet	05/04/2023, 13:24:35		1.5 KB
-426825278_00cd3598290746c4b19208965ca3b559_1.parquet	05/04/2023, 12:05:34		1.5 KB
-426825278_01074a6a27e24097bc1e5f9fda50d88_1.parquet	05/04/2023, 03:58:09		1.5 KB
-426825278_0140e7e0b6aa47ae9e331b19c0a6254a_1.parquet	05/04/2023, 13:09:34		1.4 KB
-426825278_019d7c8159bc4913a0f4658db0950640c_1.parquet	05/04/2023, 13:07:34		1.5 KB
-426825278_01bf6e7d7f61e4c48beafae6f4c5d2f76_1.parquet	05/04/2023, 12:38:17		1.5 KB
-426825278_023f8818fcd4ae58f58a7468bc4fb93_1.parquet	05/04/2023, 10:52:26		1.4 KB
-426825278_0284abb45854ebabb663cab1e8b3eb_1.parquet	05/04/2023, 10:47:26		1.5 KB
-426825278_044fa8d9d8b4deb9e85ce961bdcad91_1.parquet	05/04/2023, 15:04:30		1.5 KB
-426825278_04928c3f26c47eb08f729a22a9f2ba_1.parquet	05/04/2023, 12:06:34		1.5 KB
-426825278_04c4c86f92394d1ba09997ba07e3bb5_1.parquet	05/04/2023, 03:55:08		1.5 KB
-426825278_059e084361e14882b406b2b1e519b650_1.parquet	05/04/2023, 13:26:36		1.5 KB
-426825278_05d8d9755ba4f4c4c2bb152cd8e66_1.parquet	05/04/2023, 03:03:04		1.5 KB
-426825278_06807c5c47164660aed3b9e11ed70210_1.parquet	05/04/2023, 16:33:34		1.5 KB
-426825278_076a035144054a34bdc6ce94aac8a99ed_1.parquet	05/04/2023, 10:15:16		1.5 KB

# CHALLENGE 4: Publish to Power Bi Service

## 1st Attempt: Using CETAS created external table to store calculated ranks

```
1
2
3 CREATE EXTERNAL TABLE boatranking WITH ( LOCATION = 'boatranking', DATA_SOURCE = BoatRanksDataLake, FILE_FORMAT = [RankingParquetFormat] ) AS
4 select boat, latitude, longitude, corner,
5 (Rank() over(order by corner desc, c1)) as final_rank
6 FROM
7 (
8     SELECT
9     boat, latitude, longitude,
10    (case
11        when latitude < -50.0 then (Rank() over( PARTITION by corner order by longitude desc))
12        else (Rank() over(PARTITION by corner order by latitude asc))
13    END) as c1
14 ) as sub1
15
16 (
17     SELECT top 10 (IIF (latitude > -50.0, 0, 1)) as corner, boat, latitude, longitude, EventProcessedUtcTime
18     FROM
19     OPENROWSET(
20         BULK 'https://lambdastorage1.blob.core.windows.net/blobcontainer1/sailingdata/year=2023/month=*/day=*/**',
21         FORMAT = 'PARQUET'
22     ) as rows
23     order by EventProcessedUtcTime desc
24 ) as sub1
25 ) as sub2
26
27
28 --drop external table dbo.boatranking
```

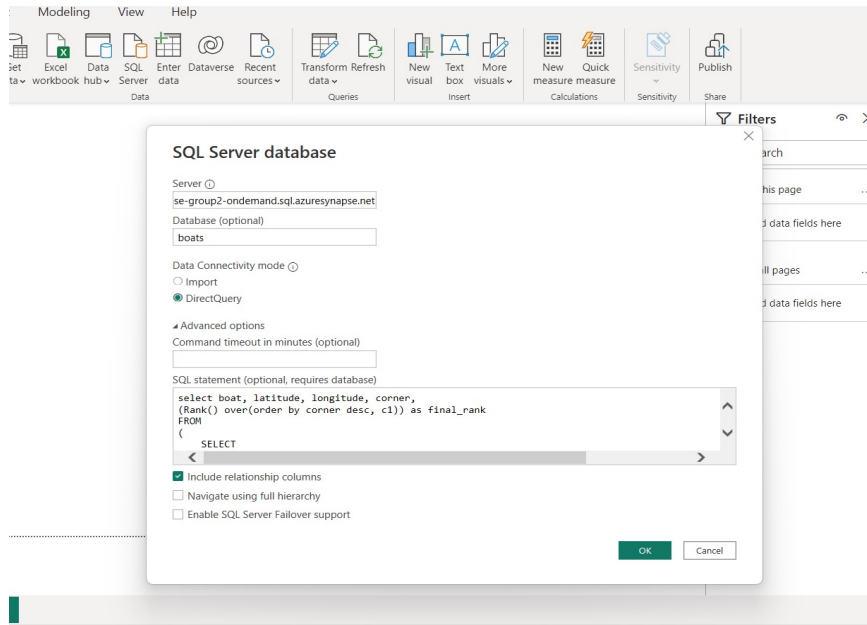
Created Second External Table with rank query to summarize data and calculate ranking.

**Problem!** Rank table was static. Didn't update as new data came in.



# CHALLENGE 4: Publish to Power BI Service

Final Solution: Rank Table created in Power BI desktop using Synapse SQL Database



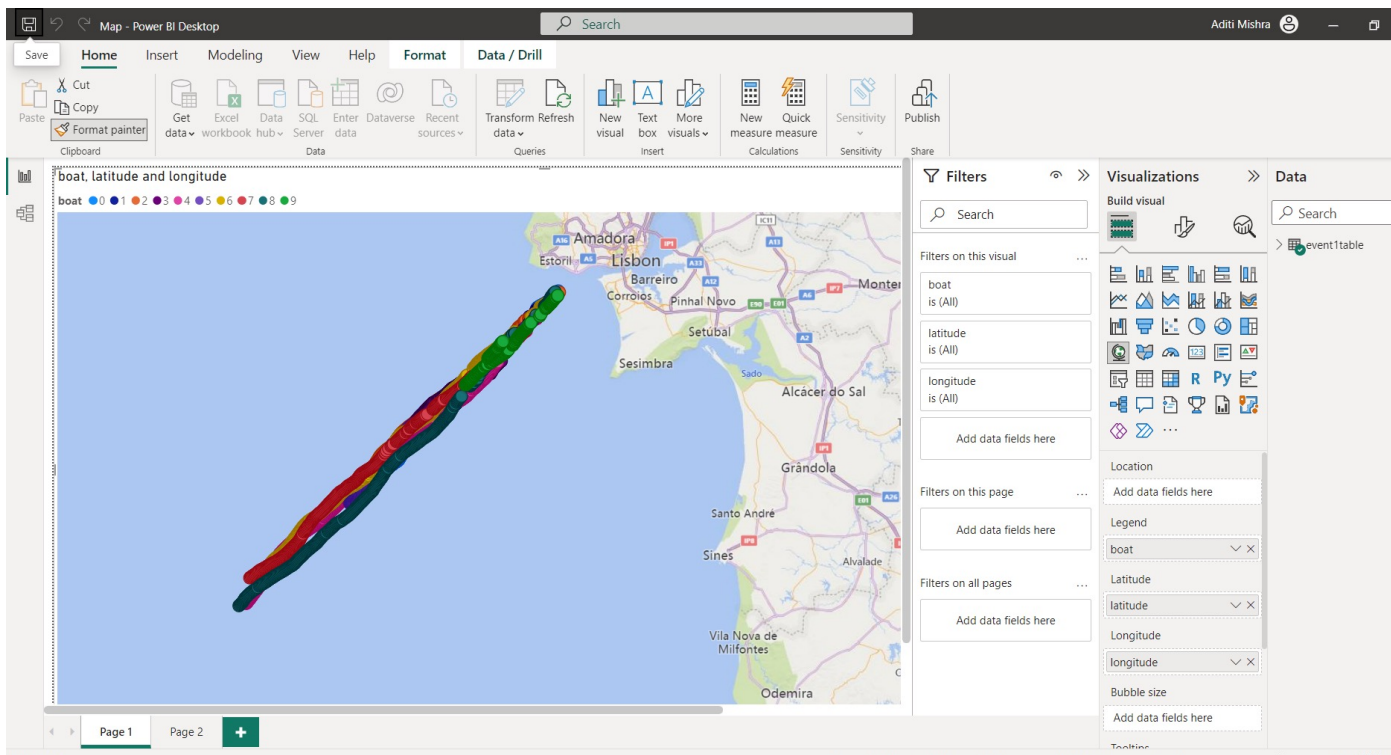
Imported External Table created in  
Serverless SQL Pool in Power  
BI Desktop

Connection mode: Direct Query

Published to power BI service

# CHALLENGE 4: Publish to Power Bi Service

## Map creation in Power BI desktop using Power BI Datasets





# DASHBOARD LIVE DATA COMPARISON

