

Developer's and User's Guide

AIS-data and Spatio-temporal Database - Port visits geo-solution

ESSNET BD II – WPE Tracking Ships

Version 2020-10-08

Prepared by Eleni Bisioti , ELSTAT, Greece

Contents

1	Introduction	1
2	Set up SSL connection to EC Dataplatfrom for PostgreSQL Database	2
3	Using pgAdmin to Connect to PostgreSQL DB Instance.....	2
4	Basic tables of the database :	4
5	Tables for each rough port area.....	5
6	Port of <i>Piraeus</i>	6
6.1	Central Piraeus port	8
6.2	Cargo Terminals	12
6.3	F2 table for the Port of PIRAEUS.....	14
7	Port of <i>Świnoujście</i>	15
7.1	F2 table for the port of Świnoujście	20
8	Port of <i>Amsterdam</i>	21
8.1	Tanker terminals	22
8.2	Tankers' arrivals at 7/3/2017	23
9	Port of <i>Rotterdam</i>	26

1 Introduction

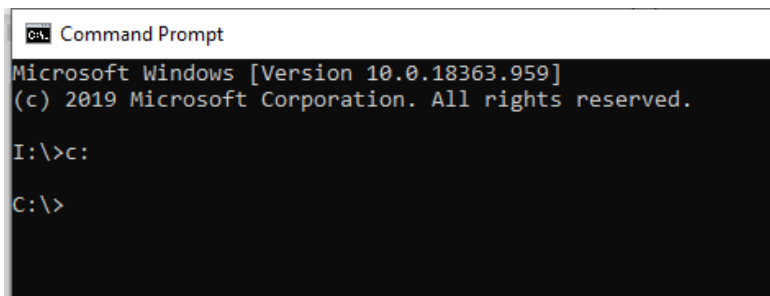
Port Visits Geo-Solution is implemented in PostgreSQL with PostGIS in a Database Instance named **estatdsl2531** on EC Dataplatfrom. Can be also deployed in a standalone computer or server. Prerequisites are the installation of PostgreSQL with PostGIS extention and PgAdmin. Developer/user should have basic skills on SQL Databases, understanding and running sql scripts , be familiar with AIS ships position reports (decoded AIS messages 1,2,3) , AIS static and voyage data (decoded AIS message 5) and working with coordinates on maps .

From the raw AIS messages (1,2,3 and 5) used as an input, we ended up creating an AIS spatio-temporal database (DB) of ships movements. The added value is that Spatio-temporal select queries give results interactively, positions and distances, due to geometry viewer, are placed on map without an extra visualization tool, in the same record field connecting two events with different timestamps as for example previous and next position of a ship is supported, creation of tables with geometric shapes as records and can be used in many cases one of which is the compilation of F2-table.

2 Set up SSL connection to EC Dataplatform for PostgreSQL Database

Before connecting to PostgreSQL Database, an SSL connection to EC Dataplatform has to be set up, following the steps below

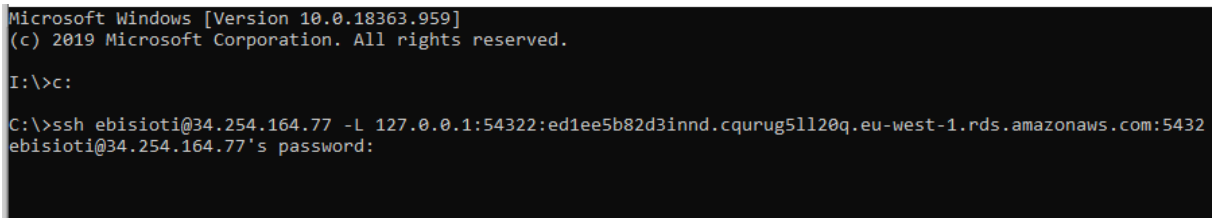
1. Invoke Command Prompt to your computer.
2. Change to drive C: (Write c: , press Enter)



```
Command Prompt
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

I:\>c:
C:\>
```

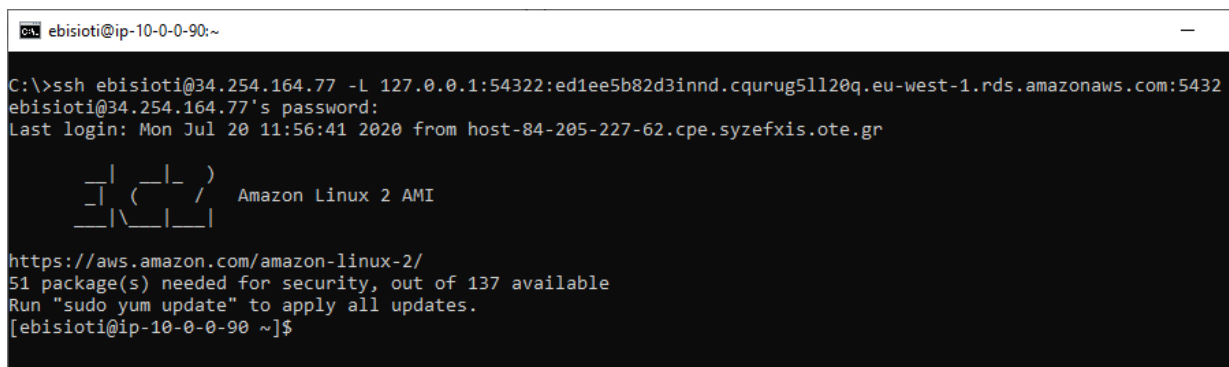
3. Write the command :
`ssh your_username@34.254.164.77 -L 127.0.0.1:54322:ed1ee5b82d3innd.cqurug5ll20q.eu-west-1.rds.amazonaws.com:5432`



```
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

I:\>c:
C:\>ssh ebisioti@34.254.164.77 -L 127.0.0.1:54322:ed1ee5b82d3innd.cqurug5ll20q.eu-west-1.rds.amazonaws.com:5432
ebisioti@34.254.164.77's password:
```

Give **your_password** (EC Dataplatform password)



```
ebisioti@ip-10-0-0-90:~
C:\>ssh ebisioti@34.254.164.77 -L 127.0.0.1:54322:ed1ee5b82d3innd.cqurug5ll20q.eu-west-1.rds.amazonaws.com:5432
ebisioti@34.254.164.77's password:
Last login: Mon Jul 20 11:56:41 2020 from host-84-205-227-62.cpe.syzefxis.ote.gr

 _ _ | ( _ | _ )
 _ | ( _ | _ |
 _ | \ _ | _ |

 Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
51 package(s) needed for security, out of 137 available
Run "sudo yum update" to apply all updates.
[ebisioti@ip-10-0-0-90 ~]$
```

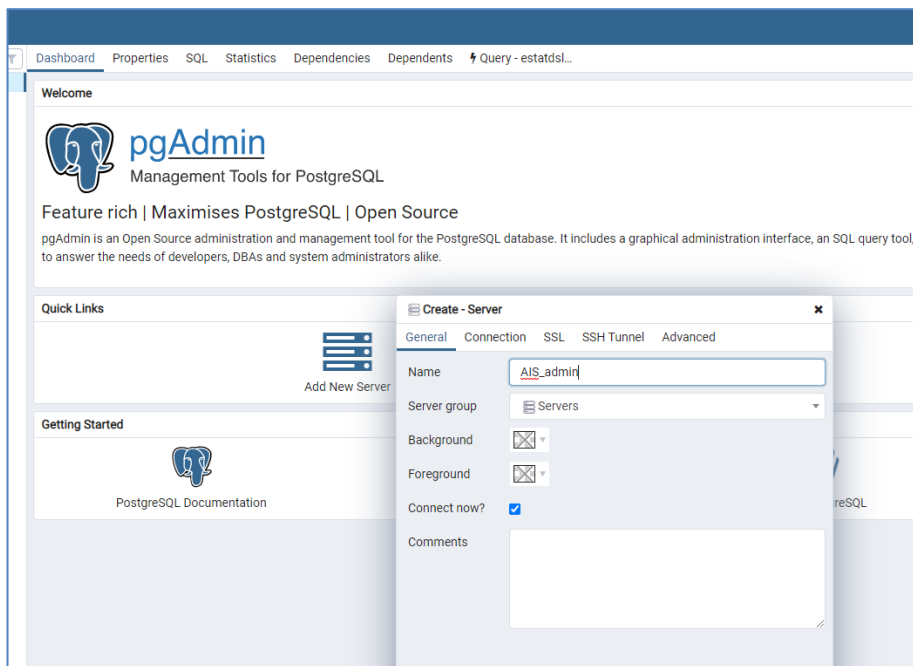
Then, minimize this screen (Do not exit or close it) and proceed launching PgAdmin

3 Using pgAdmin to Connect to PostgreSQL DB Instance

1. Install pgAdmin from <http://www.pgadmin.org/>. You can download and use pgAdmin without having a local instance of PostgreSQL on your client computer.
2. Launch the pgAdmin application on your client computer.
3. Choose Add Server from the File menu.

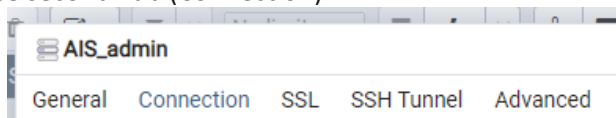


Click on Add New Server

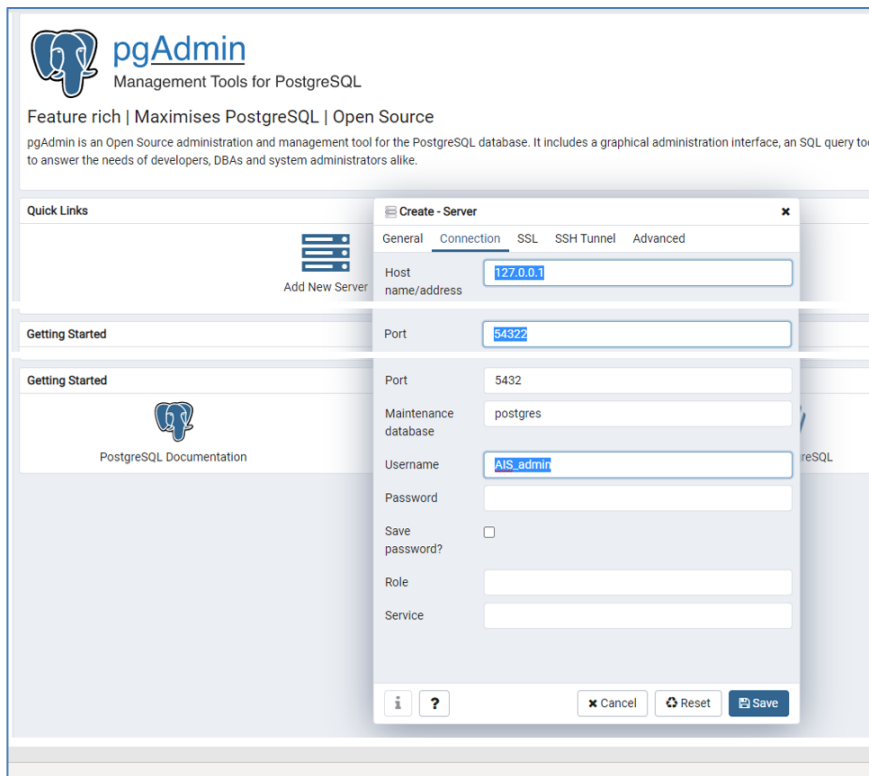


Name: AIS_admin

Choose second Tab (Connection)



Enter required information



Host name/address : 127.0.0.1

Port : 5432

Username: AIS_admin

Password: *****

and save.

4 Basic tables of the database :

emsalocs_201703d2 , which contains two days (6/3/2017 to 7/3/2017) decoded AIS messages 1,2,3 from EMSA uploaded from csv files Each csv file contains data about the following fields,

rec_time; mmsi; msgtype; lat; lon; rot; sog; cog; heading; navstatus; draught; eta; destination
which looks like that

2017-03-06T00:00:00;205202190;1;51.282003;4.367547;;0;320.8;;0;;;

2017-03-06T00:00:00;211211520;3;55.135915;12.64623;-720;10.6;174.5;169;0;;;

2017-03-06T00:00:00;211214670;1;53.475818;9.956457;;0;;;0;;;

emsaships_201703 which contains one month (1/3/2017 to 31/3/2017) AIS static (decoded AIS message 5) from EMSA uploaded from csv files Each csv file contains data about the following fields,

mmsi; msgtype; imo; vessel_name; callsign; shiptype_ais; v_length; v_width

which looks like that

229929000;5;9708875;AL ZUBARA;9HA3726;71;400;59

230202000;5;8503503;STEEL;OIVR;70;167;27

----- Create **BASIC** tables **emsalocs_201703d2** and **emsaships_201703** and upload data from csv files

CREATE TABLE emsalocs_201703d2

(

rec_time timestamp without time zone,

mmsi character varying ,

msgtype character varying ,

lat double precision,

lon double precision,

```

rot numeric,
sog numeric,
cog numeric,
heading character varying ,
navstatus character varying ,
draught character varying ,
eta character varying ,
destination character varying );

```

```

CREATE TABLE emsaships_201703
(
  mmsi character varying ,
  imo character varying ,
  msgtype character varying ,
  vessel_name character varying ,
  callsign character varying ,
  shiptype_ais character varying,
  s_length numeric,
  s_width numeric);

```

---- count number of records of emsalocs_201703d2 table

```
select count(*) from emsalocs_201703d2;
```

---- count number of records of emsaships_201703 table

```
select count(*) from emsaships_201703;
```

---- Field MMSI has 9 digits . Delete records with wrong number of digits for mmsi with code:

```

delete from emsalocs_201703d2
where length(mmsi)<9 or length(mmsi)>9; --28843 records deleted

```

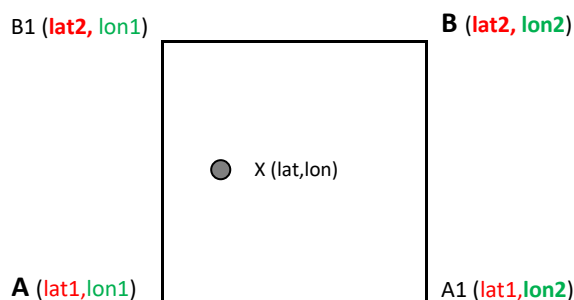
----The first digit of MMSI is a number from 2 to 7 and indicates the vessel's continent 2=Europe, 3=North or
 -----Central America, Caribbean, 4=Asia, 5=Oceania, 6=Africa, 7=South America) .

```

delete from emsalocs_201703d2
where to_number(mmsi,'9')<2 or to_number(mmsi,'9')>7; ---20987 records deleted

```

5 Tables for each rough port area



For a ship to be inside the rectangle its position X (lon, lat) should be $\text{lon1} < \text{lon} < \text{lon2}$ and $\text{lat1} < \text{lat} < \text{lat2}$

----- Create table pirlocs_201703d2 for Port of Piraeus

```

create table pirlocs_201703d2 as select * from emsalocs_201703d2
where
lon>23.499 and lon<23.8701 and lat>37.6081 and lat<37.9714;

```

----- Create table swinlocs_201703d2 for Port of Świnoujście

```
create table swinlocs_201703d2 as select * from emsalocs_201703d2
where
lon>14.250708 and lon<14.286217 and lat>53.88011 and lat<53.951968;
```

----- Create table amslocs_201703d2 for Port of Amsterdam

```
create table amslocs_201703d2 as select * from emsalocs_201703d2
where
lon>4.7298 and lon<4.8814 and lat>52.3878 and lat<52.4406;
```

----- Create table amslocs_201703d2 for Port of Rotterdam

```
create table rotlocs_201703d2 as select * from emsalocs_201703d2
where
lon>3.9491 and lon<4.4808 and lat>51.8695 and lat<51.9970;
```

6 Port of Piraeus



---- Create table pirlocs_201703d2 for Port of Piraeus

```
create table pirlocs_201703d2 as select * from emsalocs_201703d2
where lon>23.499 and lon<23.8701 and lat>37.6081 and lat<37.9714;
```

---- Create indexes and geometry point field (geom) from coordinates to table pirlocs_201703d2

```
create index pir_posindex on pirlocs_201703d2 (mmsi,lon,lat,rec_time);
alter table pirlocs_201703d2 add column geom geometry(point,4326);
update pirlocs_201703d2 set geom=ST_SetSRID(ST_MakePoint(lon,lat),4326);
create index geom_pirindex on pirlocs_201703d2 using GIST(geom);
```

--Create table **pir**movements_201703d2 (by enriching pirlocs_201703d2 table) with fields that show

-----previous (geom1) and next (geom2) position of a ship,

-----time needed to cover the distance between the two positions (duration_secs),

-----distance covered by the ship from position1 to position2 (dist)

```
create table pirmovements_201703d2 as (
SELECT
mmsi,day_when,start_ts,end_ts,geom1,geom2,lat1,lat2,lon1,lon2,
extract(epoch from (end_ts - start_ts)) AS duration_secs,
```

```

st_distance(st_transform(geom1, 28992), st_transform(geom2, 28992)) AS dist,
st_makeline(geom1,geom2)::geometry(LineString, 4326) AS geo_segment
FROM
  (SELECT
    mmsi,date_trunc('day',rec_time) as day_when,
    rec_time AS start_ts,
    lead(rec_time) OVER w AS end_ts,
    geom AS geom1,
    lead(geom) OVER w AS geom2,
    lat AS lat1,
    lead(lat) OVER w AS lat2,
    lon AS lon1,
    lead(lon) OVER w AS lon2
  FROM
    pirlocs_201703d2
  WINDOW w AS (PARTITION BY mmsi, date_trunc('day',rec_time) ORDER BY rec_time)
  ) as q);

```

----- delete from table **pir**movements_201703d2 records that show no movement

```
delete from pirmovements_201703d2 where geom2 is null or dist=0; --6445 records deleted
```

----Enrich table **pir**movements_201703d2 with columns that show

---the velocity (veloc),

---the difference of latitude (diflat2lat1) and longitude (diflon2lon1) when a ship is moving from position1 to position2

```
alter table pirmovements_201703d2
```

```

  add veloc numeric,
  add diflat2lat1 numeric,
  add diflon2lon1 numeric;

```

```
update pirmovements_201703d2
```

```
SET veloc=dist/duration_secs;
```

```
update pirmovements_201703d2
```

```
SET veloc=round(veloc,2);
```

```
update pirmovements_201703d2
```

```
SET diflat2lat1=(lat2-lat1)*1000;
```

```
update pirmovements_201703d2
```

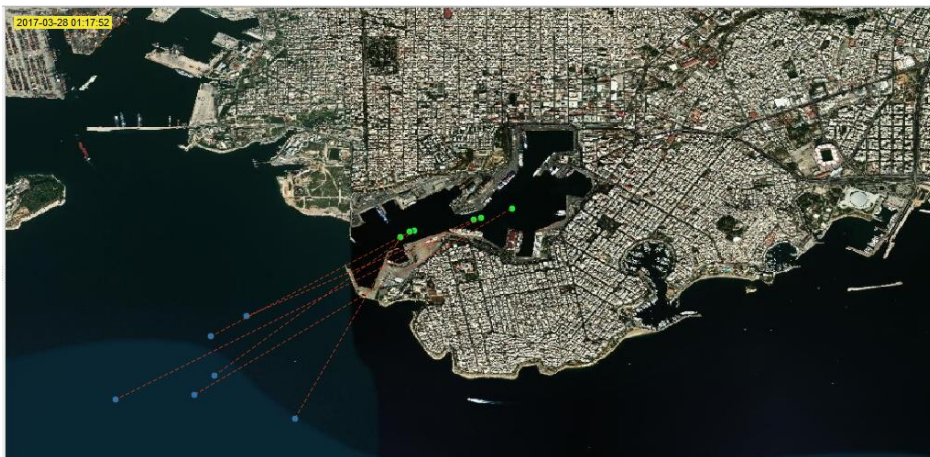
```
SET diflon2lon1=(lon2-lon1)*1000;
```


6.1 Central Piraeus port



----Selection of ships arriving at the port of Piraeus **CENTRAL PIRAEUS PORT** (passenger Ships ,Passenger/Ro-Ro Cargo Ships and Cruise Ships)

```
select * from pirmovements_201703d2
where
  veloc>0 and
  (diflat2lat1 between 0.0001 and 1000) and -----lat2>lat1 approaching Central Port entrance
  (diflon2lon1 between -1000 and 1000) and ----- default values
  (lon1 between -1000 and 1000) and ----- default values
  (lon2 between 23.6150 and 23.64999) and ----- longitude values of position 2 entering the Central Port
  (lat1 between 0 and 37.937 ) and ----- latitude values of position 1 until the entrance of the Central Port
  (lat2 between 37.937 and 1000) ----- latitude of position 2 entering the port
order by mmsi, start_ts
```



---Number of arrivals per ship (mmsi) for the port of Piraeus

```
select mmsi, count(mmsi) as arrivals from pirmovements_201703d2
where
  veloc>0 and
  (diflat2lat1 between 0.0001 and 1000) and -----lat2>lat1 approaching Central Port entrance
  (diflon2lon1 between -1000 and 1000) and ----- default values
```


(lon1 between -1000 and 1000) and ----- default values
(lon2 between 23.6150 and 23.64999) and ----- longitude values of position 2 entering the Central Port
(lat1 between 0 and 37.937) and ----- latitude values of position 1 until the entrance of the Central Port
(lat2 between 37.937 and 1000) ----- latitude of position 2 entering the port
group by mmsi order by mmsi;

Output1

a/a	MMSI	ARRIVALS
1	237021400	6
2	237023700	6
3	237024500	6
4	237032000	1
5	237240400	1
6	237611000	1
7	237641000	1
..		

-----Link the above query to table emsaships_201703 to enrich results with the available vessels' characteristics (from AIS data)

```
select
a.mmsi , a.arrivals,
b.imo, b.vessel_name, b.shiptype_ais,b.s_length,b.s_width
from
(select mmsi, count(mmsi) as arrivals from pirmovements_201703d2
where
veloc>0 and
(diflat2lat1 between 0.0001 and 1000) and -----lat2>lat1 approaching Central Port entrance
(diflon2lon1 between -1000 and 1000) and ----- default values
(lon1 between -1000 and 1000) and ----- default values
(lon2 between 23.6150 and 23.64999) and ----- longitude values of position 2 entering the Central Port
(lat1 between 0 and 37.937 ) and ----- latitude values of position 1 until the entrance of the Central Port
(lat2 between 37.937 and 1000) ----- latitude of position 2 entering the port
group by mmsi order by mmsi) a,
emsaships_201703 b
where a.mmsi=b.mmsi
order by a.mmsi
```

Output2

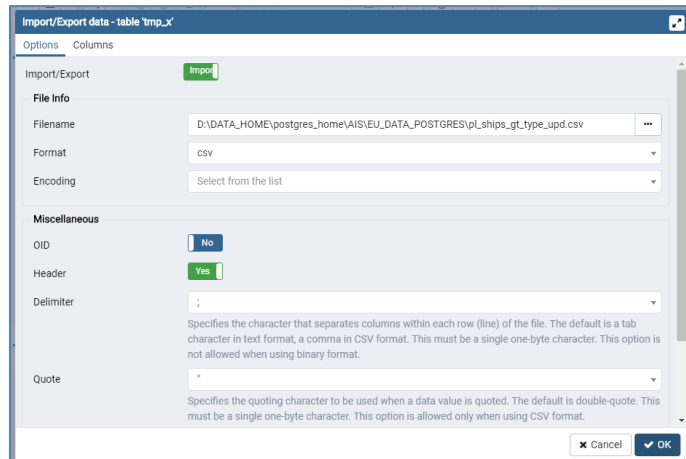
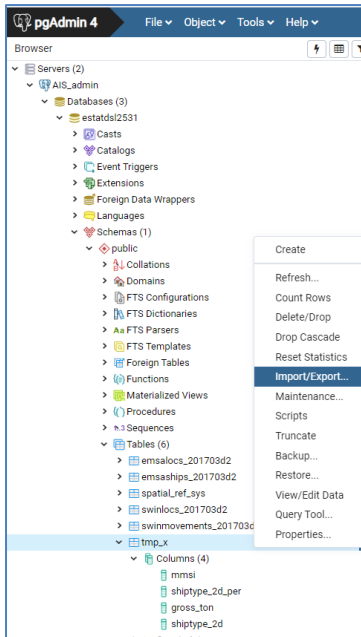
a/a	mmsi	arrivals	imo	vessel_name	shiptype_ais	s_length	s_width
1	237021400	6	8969343	AG NEKTARIOS AIGINAS	60	75	14
2	237023700	6	8331479	FLYING DOLPHIN XVIII			
3	237024500	6	8875700	FLYING DOLPHIN XXIX	40	34	6
4	237032000	1	7814058	KRITI II	60	192	94
5	237240400	1	8334756	ECO SAILOR			

-----Alter table emsaships_201703 add fields for ship type (shiptype_2d), description of ship type(shiptype_2d_per), gross tonnage (gross_ton), gross tonnage category (gross_ton_klim) according to Directive

```
alter table emsaships_201703
add shiptype_2d numeric,
add shiptype_2d_per character varying ,
add gross_ton numeric,
add gross_ton_klim character varying ;
```

-----Create temporary table tmp_x to upload csv file with characteristics for ship type (shiptype_2d), description of ship type (shiptype_2d_per), gross tonnage (gross_ton), gross tonnage category (gross_ton_klim) according to Directive for the above selected mmsi's

CREATE TABLE tmp_x (mmsi character varying, shiptype_2d_per character varying,gross_ton numeric,shiptype_2d numeric);



Csv file contains information as the following:

1	mmsi	shiptype_2d_per	gross_ton	shiptype_2d
2	237194800	Container	148390	31
3	239839000	Liquid bulk	793	10
4	241495000	General cargo, non-specialised	3.633	33
5	239943900	General cargo, non-specialised	1.839	33
6	237022000	General cargo, non-specialised	27.239	33
7	372482000	Dry bulk	1946	20
8	353258000	General cargo, non-specialised	1.851	33
9	358850000	Container	21648	31
10	353852000	Container	21633	31
11	240010200	General cargo, non-specialised	4.092	33
12	239839000	Passenger	158	25

-----Update ships table **emsaships_201703** from temporary table

UPDATE public.emsaships_201703

SET shiptype_2d=tmp_x.shiptype_2d, shiptype_2d_per=tmp_x.shiptype_2d_per, gross_ton=tmp_x.gross_ton

FROM public.tmp_x

WHERE emsaships_201703.mmsi=tmp_x.mmsi;

-----View the updated records

select * from public.emsaships_201703 where shiptype_2d_per is not null;

-----update field **gross_ton_klim** of ships table **emsaships_201703** according to Directive size classes using gross_ton

UPDATE emsaships_201703

SET gross_ton_klim = (CASE WHEN gross_ton BETWEEN 100 AND 499 THEN ' 01 (from 100 to 499 GT) '

WHEN gross_ton BETWEEN 500 AND 999 THEN ' 02 (from 500 to 999 GT) '

```

WHEN gross_ton BETWEEN 1000 AND 1999 THEN ' 03 (from 1 000 to 1 999 GT)'
WHEN gross_ton BETWEEN 2000 AND 2999 THEN ' 04 (from 2 000 to 2 999 GT)'
WHEN gross_ton BETWEEN 3000 AND 3999 THEN ' 05 (from 3 000 to 3 999 GT)'
WHEN gross_ton BETWEEN 4000 AND 4999 THEN ' 06 (from 4 000 to 4 999 GT)'
WHEN gross_ton BETWEEN 5000 AND 5999 THEN ' 07 (from 5 000 to 5 999 GT)'
WHEN gross_ton BETWEEN 6000 AND 6999 THEN ' 08 (from 6 000 to 6 999 GT)'
WHEN gross_ton BETWEEN 7000 AND 7999 THEN ' 09 (from 7 000 to 7 999 GT)'
WHEN gross_ton BETWEEN 8000 AND 8999 THEN ' 10 (from 8 000 to 8 999 GT)'
WHEN gross_ton BETWEEN 9000 AND 9999 THEN ' 11 (from 9 000 to 9 999 GT)'
WHEN gross_ton BETWEEN 10000 AND 19999 THEN ' 12 (from 10 000 to 19 999 GT)'
WHEN gross_ton BETWEEN 20000 AND 29999 THEN ' 13 (from 20 000 to 29 999 GT)'
WHEN gross_ton BETWEEN 30000 AND 39999 THEN ' 14 (from 30 000 to 39 999 GT)'
WHEN gross_ton BETWEEN 40000 AND 49999 THEN ' 15 (from 40 000 to 49 999 GT)'
WHEN gross_ton BETWEEN 50000 AND 79999 THEN ' 16 (from 50 000 to 79 999 GT)'
WHEN gross_ton BETWEEN 80000 AND 99999 THEN ' 17 (from 80 000 to 99 999 GT)'
WHEN gross_ton BETWEEN 100000 AND 149999 THEN ' 18 (from 100 000 to 149 999 GT)'
WHEN gross_ton BETWEEN 150000 AND 199999 THEN ' 19 (from 150 000 to 199 999 GT)'
WHEN gross_ton BETWEEN 200000 AND 249999 THEN ' 20 (from 200 000 to 249 999 GT)'
WHEN gross_ton BETWEEN 250000 AND 299999 THEN ' 21 (from 250 000 to 299 999 GT)'
WHEN gross_ton BETWEEN 300000 AND 3000000000000000 THEN ' 22 ( ≥ 300 000 GT)'
END);

```

-----View the updated records

```
select * from public.emsaships_201703 where gross_ton_klim is not null;
```

-----View the updated records of interest (shiptype_2d has value only for maritime ships of interest)

```
select * from public.emsaships_201703 where shiptype_2d is not null;
```

----- Number of arrivals per ship (mmsi) for the port of Piraeus- enriched results with vessels' characteristics (according to Directive)
-----keeping ship types of interest

```

select
a.mmsi , a.arrivals, b.shiptype_2d, b.shiptype_2d_per, b.gross_ton, b.gross_ton_klim
from
(select mmsi, count(mmsi) as arrivals from pirmovements_201703d2
where
veloc>0 and
      (diflat2lat1 between 0.0001 and 1000) and -----lat2>lat1 approaching Central Port entrance
      (diflon2lon1 between -1000 and 1000) and ----- default values
      (lon1 between -1000 and 1000) and ----- default values
      (lon2 between 23.6150 and 23.64999) and ----- longitude values of position 2 entering the Central Port
      (lat1 between 0 and 37.937 ) and ----- latitude values of position 1 until the entrance of the Central Port
      (lat2 between 37.937 and 1000) ----- latitude of position 2 entering the port
group by mmsi order by mmsi) a,
emsaships_201703 b
where a.mmsi=b.mmsi and b.shiptype_2d in ('33','35','36')
order by a.mmsi

```

Output3

a/a	mmsi	arrivals	shiptype_2d	shiptype_2d_per	gross_ton	gross_ton_klim
1	237021400	6	33	Passenger/Ro-Ro Cargo Ship	1.091	03 (from 1 000 to 1 999 GT)
2	237023700	6	35	Passenger	162	01 (from 100 to 499 GT)
3	237024500	6	35	Passenger	162	01 (from 100 to 499 GT)
4	237032000	1	33	Passenger/Ro-Ro Cargo Ship	27.239	13 (from 20 000 to 29 999 GT)
5	237611000	1	33	Passenger/Ro-Ro Cargo Ship	37.482	14 (from 30 000 to 39 999 GT)

...

6.2 Cargo Terminals



----- create table to insert terminal polygons using coordinates -----

```
CREATE TABLE port_polys (poly_id bigserial primary key, name text, geom geometry (polygon, 4326));
```

```
INSERT INTO port_polys (name, geom)
SELECT 'pir_central_port',
ST_BuildArea (ST_GEOFROMTEXT('polygon((23.63458 37.95037,
23.64557 37.95029 ,
23.64639 37.94384,
23.63863 37.93632,
23.63094 37.93864,
23.62159 37.93372,
23.61974 37.94149,
23.63293 37.94591,
23.63458 37.95037
))',4326));
```

```
INSERT INTO port_polys (name, geom)
SELECT 'pir_cargo_term',
ST_BuildArea (ST_GEOFROMTEXT('polygon((23.57797 37.96124,
23.61179 37.96316,
23.61444 37.95749,
23.60819 37.94905,
23.59788 37.94873,
23.57861 37.95240,
23.57797 37.96124))',4326));
```

--- Select query that shows analytically the movements of ships before (geom1=position1) and inside cargo terminals polygon (geom2=position2)

```
SELECT a.mmsi, a.day_when, a.start_ts, a.end_ts,
       a.geom1, a.geom2, a.lat1, a.lat2, a.lon1, a.lon2, a.duration_secs,
       a.dist, a.geo_segment, a.veloc, a.diflat2lat1, a.diflon2lon1,
       pr.name,
       pr.geom
FROM pirmovements_201703d2 a,
```

port_polys pr

where

a.veloc>0 and

(a.diflat2lat1 between 0.0001 and 1000) and

(a.diflon2lon1 between -1000 and 1000) and

(a.lon1 between 23.5400 and 23.6500) and

(a.lon2 between 23.5660 and 23.6100) and

(a.lat1 between -1000 and 37.9511) and

(a.lat2 between 37.951 and 1000) and

st_contains(pr.geom, a.geom2) = true

order by a.start_ts;

----- Number of arrivals per ship (mmsi) for **the Cargo Terminals** port of Piraeus- enriched results with vessels' characteristics
(according to Directive)
-----keeping ship types of interest

```
select x.mmsi, x.arrivals, c.shiptype_2d, c.shiptype_2d_per, c.gross_ton,c.gross_ton_klim
from
(SELECT a.mmsi, count(a.mmsi) as arrivals
FROM pirmovements_201703d2 a,
      port_polys pr
```

where

a.veloc>0 and

(a.diflat2lat1 between 0.0001 and 1000) and

(a.diflon2lon1 between -1000 and 1000) and

(a.lon1 between 23.5400 and 23.6500) and

(a.lon2 between 23.5660 and 23.6100) and

(a.lat1 between -1000 and 37.9511) and

(a.lat2 between 37.951 and 1000) and

st_contains(pr.geom, a.geom2) = true

group by a.mmsi

order by a.mmsi

) x,

emsaships_201703 c

where x.mmsi=c.mmsi and c.shiptype_2d in ('31','32','33','34')

order by x.mmsi;

a/a	mmsi	arrivals	shiptype_2d	shiptype_2d_per	gross_ton	gross_ton_klim
1	220593000	1	31	Container	99.002	17 (from 80 000 to 99 999 GT)
2	235102681	1	31	Container	99.950	17 (from 80 000 to 99 999 GT)
3	237183800	1	34	Dry cargo barge	796	02 (from 500 to 999 GT)
4	237353500	3	33	General cargo, non-specialised	303	01 (from 100 to 499 GT)
5	241511000	1	31	Container	8.737	10 (from 8 000 to 8 999 GT)
6	247039300	1	32	Specialised cargo	37.726	14 (from 30 000 to 39 999 GT)
7	249136000	1	31	Container	17.964	12 (from 10 000 to 19 999 GT)

...

6.3 F2 table for the Port of PIRAEUS

----Select query for F2 table

```
select f.shiptype_2d || '-' || f.shiptype_2d_per as "Type", f.gross_ton_klim as "Size class", sum(f.arrivals) as "Total number of  
vessels (arrivals)", sum(f.arrivals*f.gross_ton) as "Total weight (in GT)"  
from  
(
```

----CENTRAL PORT

```
select  
a.mmsi , a.arrivals,b.shiptype_2d, b.shiptype_2d_per, b.gross_ton, b.gross_ton_klim  
from  
    (select mmsi, count(mmsi) as arrivals from pirmovements_201703d2  
    where  
    veloc>0 and  
    (diflat2lat1 between 0.0001 and 1000) and -----lat2>lat1 approaching Central Port entrance  
    (diflon2lon1 between -1000 and 1000) and ----- default values  
    (lon1 between -1000 and 1000) and ----- default values  
    (lon2 between 23.6150 and 23.64999) and ----- longitude values of position 2 entering the Central Port  
    (lat1 between 0 and 37.937 ) and ----- latitude values of position 1 until the entrance of the Central Port  
    (lat2 between 37.937 and 1000) ----- latitude of position 2 entering the port  
    group by mmsi order by mmsi) a,  
    emsaships_201703 b  
where a.mmsi=b.mmsi and b.shiptype_2d in ('33','35','36')
```

UNION

----CARGO TERMINALS

```
select x.mmsi, x.arrivals, c.shiptype_2d, c.shiptype_2d_per, c.gross_ton,c.gross_ton_klim  
from  
    (SELECT a.mmsi, count(a.mmsi) as arrivals  
    FROM pirmovements_201703d2 a,  
    port_polys pr  
    where  
    a.veloc>0 and  
    (a.diflat2lat1 between 0.0001 and 1000) and  
    (a.diflon2lon1 between -1000 and 1000) and  
    (a.lon1 between 23.5400 and 23.6500) and  
    (a.lon2 between 23.5660 and 23.6100) and  
    (a.lat1 between -1000 and 37.9511 ) and  
    (a.lat2 between 37.951 and 1000) and  
    st_contains(pr.geom, a.geom2) = true  
    group by a.mmsi  
    order by a.mmsi  
    ) x,
```

```
    emsaships_201703 c  
where x.mmsi=c.mmsi and c.shiptype_2d in ('31','32','33','34')
```

)f

```
group by f.shiptype_2d,f.gross_ton_klim,f.shiptype_2d_per  
order by f.shiptype_2d,f.gross_ton_klim,f.shiptype_2d_per
```

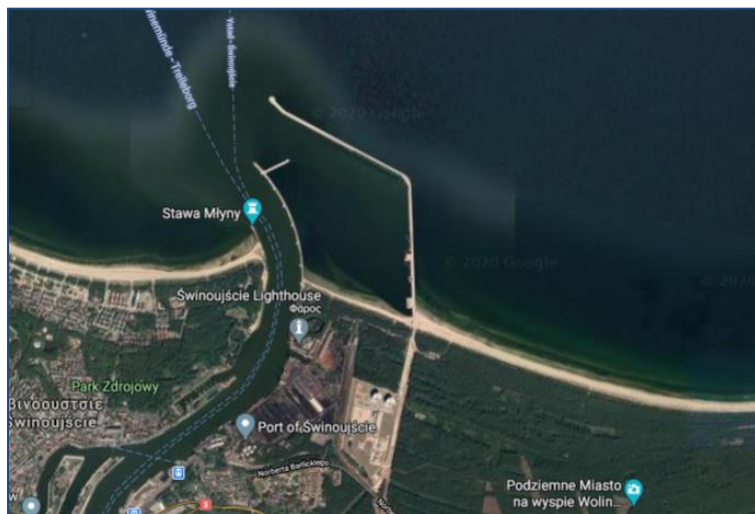
Output -F2 table for the port of Piraeus

Type	Size class	Total number of vessels (arrivals)	Total weight (in GT)
31-Container	09 (from 7 000 to 7 999 GT)	1	7.852
31-Container	10 (from 8 000 to 8 999 GT)	1	8.737
31-Container	11 (from 9 000 to 9 999 GT)	2	19.920
31-Container	12 (from 10 000 to 19 999 GT)	4	63.422
31-Container	13 (from 20 000 to 29 999 GT)	1	20.620

31-Container	17 (from 80 000 to 99 999 GT)	3	289.402
31-Container	19 (from 150 000 to 199 999 GT)	1	153.090
32-Specialised cargo	14 (from 30 000 to 39 999 GT)	2	76.075

...

7 Port of Świnoujście



---- Create table swinlocs_201703d2 for Port of Świnoujście with code:

```
create table swinlocs_201703d2 as select * from emsalocs_201703d2
where lon>14.250708 and lon<14.286217 and lat>53.88011 and lat<53.951968;
```

---- Create indexes and geometry point field (geom) from coordinates to table swinlocs_201703d2

```
create index swin_posindex on swinlocs_201703d2 (mmsi,lon,lat,rec_time);

alter table swinlocs_201703d2 add column geom geometry(point,4326);

update swinlocs_201703d2 set geom=ST_SetSRID(ST_MakePoint(lon,lat),4326);

create index geom_swinindex on swinlocs_201703d2 using GIST(geom);
```

--Create table swinmovements_201703d2 (by enriching swinlocs_201703d2 table) with fields that show

-----previous (geom1) and next (geom2) position of a ship,

-----time needed to cover the distance between the two positions (duration_secs),

-----distance covered by the ship from position1 to position2 (dist)

```
create table swinmovements_201703d2 as (
SELECT
mmsi,day_when,start_ts,end_ts,geom1,geom2,lat1,lat2,lon1,lon2,
extract(epoch from (end_ts - start_ts)) AS duration_secs,
st_distance(st_transform(geom1, 28992), st_transform(geom2, 28992)) AS dist,
st_makeline(geom1,geom2)::geometry(LineString, 4326) AS geo_segment
FROM (SELECT
      mmsi,date_trunc('day',rec_time) as day_when,
      rec_time AS start_ts,
      lead(rec_time) OVER w AS end_ts,
      geom AS geom1,
      lead(geom) OVER w AS geom2,
      lat AS lat1,
      lead(lat) OVER w AS lat2,
```

```
lon AS lon1,
lead(lon) OVER w AS lon2
FROM
swinlocs_201703d2
WINDOW w AS (PARTITION BY mmsi, date_trunc('day',rec_time) ORDER BY rec_time)
) as q);
```

---- delete from table swinmovements_201703d2 records that show no movement

```
delete from swinmovements_201703d2 where geom2 is null or dist=0; --2003 records deleted
```

----Enrich table swinmovements_201703d2 with columns that show

---the velocity (veloc),

---the difference of latitude (diflat2lat1) and longitude (diflon2lon1) when a ship is traveling from position1 to position2

```
alter table swinmovements_201703d2
```

```
add veloc numeric,
add diflat2lat1 numeric,
add diflon2lon1 numeric;
```

```
update swinmovements_201703d2
```

```
SET veloc=dist/duration_secs;
```

```
update swinmovements_201703d2
```

```
SET veloc=round(veloc,2);
```

```
update swinmovements_201703d2
```

```
SET diflat2lat1=(lat2-lat1)*1000;
```

```
update swinmovements_201703d2
```

```
SET diflon2lon1=(lon2-lon1)*1000;
```

----Selection of ships arriving at the port of Świnoujście

```
select * from swinmovements_201703d2
```

```
where
```

```
veloc>0 and
```

```
(diflat2lat1 between -1000 and 0) and
```

```
(diflon2lon1 between -1000 and 1000) and
```

```
(lon1 between -1000 and 1000) and
```

```
(lon2 between -1000 and 1000) and
```

```
(lat1 between 53.9220 and 53.9550 ) and
```

```
(lat2 between 53.90 and 53.9219)
```

```
order by mmsi, start_ts
```

-----lat2<lat1 approaching to port entrance

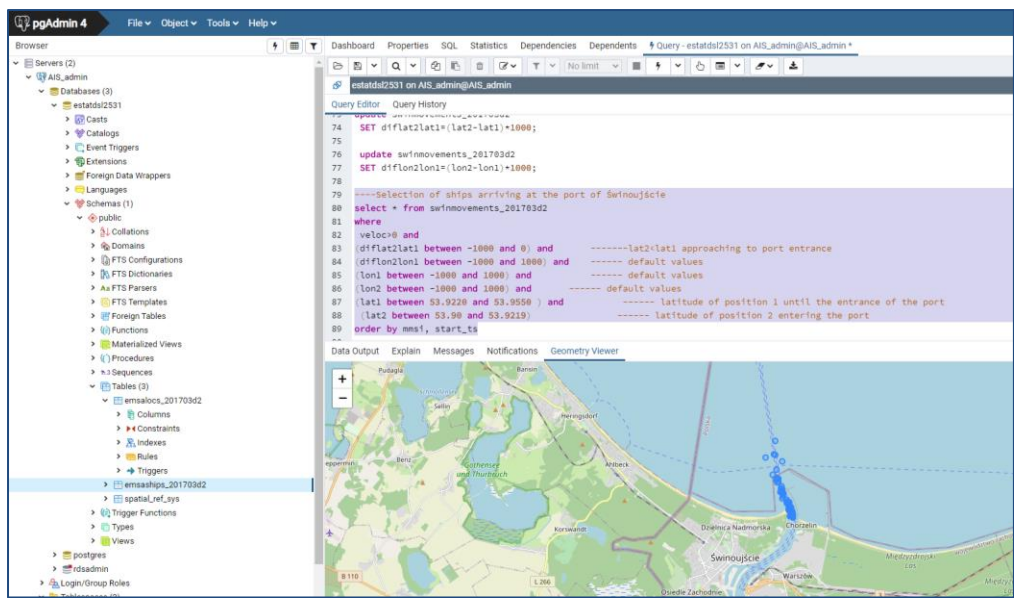
----- default values

----- default values

----- default values

----- latitude of position 1 until the entrance of the port

----- latitude of position 2 entering the port



---Number of arrivals per ship (mmsi) for the port of Świnoujście

```
select mmsi, count(mmsi) as arrivals from swinmovements_201703d2
where
veloc>0 and
(diflat2lat1 between -1000 and 0) and -----lat2<lat1 approaching to port entrance
(diflon2lon1 between -1000 and 1000) and ----- default values
(lon1 between -1000 and 1000) and ----- default values
(lon2 between -1000 and 1000) and ----- default values
(lat1 between 53.9220 and 53.9550 ) and ----- latitude of position 1 until the entrance of the port
(lat2 between 53.90 and 53.9219) ----- latitude of position 2 entering the port
group by mmsi order by mmsi;
```

Output1

a/a	mmsi	arrivals
1	205465000	1
2	209896000	2
3	210095000	2
4	211228170	1
5	211628260	1
6	212004000	1
7	212499000	3

...

-----Link the above query to table emsaships_201703 to enrich results with the available vessels' characteristics (from AIS data)

```
select
a.mmsi , a.arrivals,
b.imo, b.vessel_name, b.shiptype_ais,b.s_length,b.s_width
from
(select mmsi, count(mmsi) as arrivals from swinmovements_201703d2
where
veloc>0 and
(diflat2lat1 between -1000 and 0) and -----lat2<lat1 approaching to port entrance
(diflon2lon1 between -1000 and 1000) and ----- default values
(lon1 between -1000 and 1000) and ----- default values
(lon2 between -1000 and 1000) and ----- default values
```

(lat1 between 53.9220 and 53.9550) and ----- latitude of position 1 until the entrance of the port
 (lat2 between 53.90 and 53.9219) ----- latitude of position 2 entering the port
 group by mmsi order by mmsi) a,
 emsaships_201703 b
 where a.mmsi=b.mmsi
 order by a.mmsi

Output2

a/a	mmsi	arrivals	imo	vessel_name	shiptype_ais	s_length	s_width
1	205465000	1	9136101	FAST JEF	70	88	13
2	209896000	2	7527887	KOPERNIK	60	160	22
3	210095000	2	9019078	GALILEUSZ	60	150	23
4	211228170	1	6720834	ADLER XI	60	33	7
5	211628260	1					

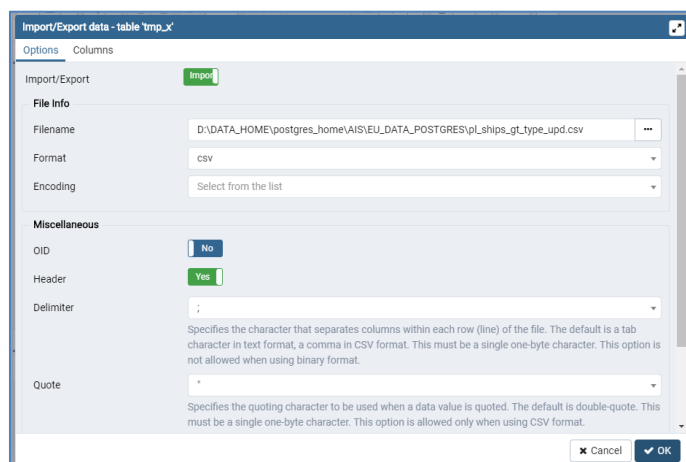
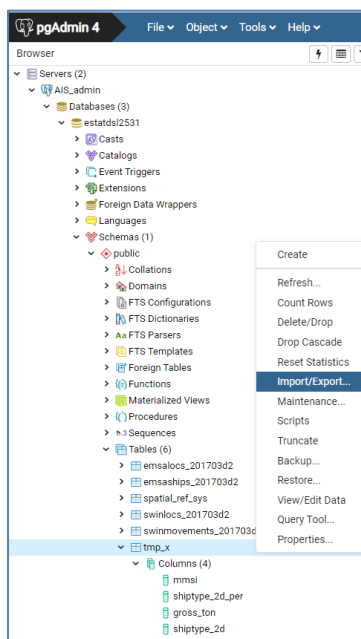
...

-----Alter table emsaships_201703 add fields for ship type (shiptype_2d), description of ship type(shiptype_2d_per), gross tonnage (gross_ton), gross tonnage category (gross_ton_klim) according to Directive

alter table emsaships_201703
 add shiptype_2d numeric,
 add shiptype_2d_per character varying ,
 add gross_ton numeric,
 add gross_ton_klim character varying ;

-----Create temporary table tmp_x to upload csv file with characteristics for ship type (shiptype_2d), description of ship type(shiptype_2d_per), gross tonnage (gross_ton), gross tonnage category (gross_ton_klim) according to Directive for the above selected mmsi's

CREATE TABLE tmp_x (mmsi character varying, shiptype_2d_per character varying,gross_ton numeric,shiptype_2d numeric);



Csv file contains the following information :

a/a	mmsi	shiptype_2d_per	gross_ton	shiptype_2d
1	205465000	General Cargo Ship	2066	33

2	209896000	Passenger/Ro-Ro Cargo Ship	14216	33
3	210095000	Passenger/Ro-Ro Cargo Ship	15848	33
4	211228170	Passenger	173	35

...

-----Update ships table **emsaships_201703** from temporary table

UPDATE public.emsaships_201703

SET shiptype_2d=tmp_x.shiptype_2d, shiptype_2d_per=tmp_x.shiptype_2d_per, gross_ton=tmp_x.gross_ton

FROM public.tmp_x

WHERE emsaships_201703.mmsi=tmp_x.mmsi;

-----View the updated records

select * from public.emsaships_201703 where shiptype_2d_per is not null;

-----update field **gross_ton_klim** of ships table **emsaships_201703** according to Directive size classes using gross_ton

UPDATE emsaships_201703

SET gross_ton_klim = (CASE WHEN gross_ton BETWEEN 100 AND 499 THEN ' 1 (from 100 to 499 GT)'
WHEN gross_ton BETWEEN 500 AND 999 THEN ' 2 (from 500 to 999 GT)'
WHEN gross_ton BETWEEN 1000 AND 1999 THEN ' 3 (from 1 000 to 1 999 GT)'
WHEN gross_ton BETWEEN 2000 AND 2999 THEN ' 4 (from 2 000 to 2 999 GT)'
WHEN gross_ton BETWEEN 3000 AND 3999 THEN ' 5 (from 3 000 to 3 999 GT)'
WHEN gross_ton BETWEEN 4000 AND 4999 THEN ' 6 (from 4 000 to 4 999 GT)'
WHEN gross_ton BETWEEN 5000 AND 5999 THEN ' 7 (from 5 000 to 5 999 GT)'
WHEN gross_ton BETWEEN 6000 AND 6999 THEN ' 8 (from 6 000 to 6 999 GT)'
WHEN gross_ton BETWEEN 7000 AND 7999 THEN ' 9 (from 7 000 to 7 999 GT)'
WHEN gross_ton BETWEEN 8000 AND 8999 THEN ' 10 (from 8 000 to 8 999 GT)'
WHEN gross_ton BETWEEN 9000 AND 9999 THEN ' 11 (from 9 000 to 9 999 GT)'
WHEN gross_ton BETWEEN 10000 AND 19999 THEN ' 12 (from 10 000 to 19 999 GT)'
WHEN gross_ton BETWEEN 20000 AND 29999 THEN ' 13 (from 20 000 to 29 999 GT)'
WHEN gross_ton BETWEEN 30000 AND 39999 THEN ' 14 (from 30 000 to 39 999 GT)'
WHEN gross_ton BETWEEN 40000 AND 49999 THEN ' 15 (from 40 000 to 49 999 GT)'
WHEN gross_ton BETWEEN 50000 AND 79999 THEN ' 16 (from 50 000 to 79 999 GT)'
WHEN gross_ton BETWEEN 80000 AND 99999 THEN ' 17 (from 80 000 to 99 999 GT)'
WHEN gross_ton BETWEEN 100000 AND 149999 THEN ' 18 (from 100 000 to 149 999 GT)'
WHEN gross_ton BETWEEN 150000 AND 199999 THEN ' 19 (from 150 000 to 199 999 GT)'
WHEN gross_ton BETWEEN 200000 AND 249999 THEN ' 20 (from 200 000 to 249 999 GT)'
WHEN gross_ton BETWEEN 250000 AND 299999 THEN ' 21 (from 250 000 to 299 999 GT)'
WHEN gross_ton BETWEEN 300000 AND 3000000000000000 THEN ' 22 (≥ 300 000 GT)'
END) ;

-----View the updated records

select * from public.emsaships_201703 where gross_ton_klim is not null;

-----View the updated records of interest (shiptype_2d has value only for maritime ships of interest)

select * from public.emsaships_201703 where shiptype_2d is not null;

----- Number of arrivals per ship (mmsi) for the port of Świnoujście - enriched results with vessels' characteristics (according to Directive)

-----keeping ship types of interest

select

a.mmsi , a.arrivals, b.shiptype_2d, b.shiptype_2d_per, b.gross_ton, b.gross_ton_klim

from

(select mmsi, count(mmsi) as arrivals from swinmovements_201703d2

where

veloc>0 and

(diflat2lat1 between -1000 and 0) and -----lat2<lat1 approaching to port entrance

(diflon2lon1 between -1000 and 1000) and ----- default values

(lon1 between -1000 and 1000) and ----- default values

(lon2 between -1000 and 1000) and ----- default values

(lat1 between 53.9220 and 53.9550) and ----- latitude of position 1 until the entrance of the port

(lat2 between 53.90 and 53.9219) ----- latitude of position 2 entering the port

group by mmsi order by mmsi) a,

emsaships_201703 b

where a.mmsi=b.mmsi and b.shiptype_2d is not null

order by a.mmsi

Output3

a/a	mmsi	arrivals	shiptype_2d	shiptype_2d_per	gross_ton	gross_ton_klim
1	205465000	1	33	General Cargo Ship	2.066	4 (from 2 000 to 2 999 GT)
2	209896000	2	33	Passenger/Ro-Ro Cargo Ship	14.216	12 (from 10 000 to 19 999 GT)
3	210095000	2	33	Passenger/Ro-Ro Cargo Ship	15.848	12 (from 10 000 to 19 999 GT)
4	211228170	1	35	Passenger	173	1 (from 100 to 499 GT)
5	212004000	1	33	Passenger/Ro-Ro Cargo Ship	14.417	12 (from 10 000 to 19 999 GT)
6	212499000	3	33	Passenger/Ro-Ro Cargo Ship	26.796	13 (from 20 000 to 29 999 GT)
7	231711000	1	32	Refrigerated Cargo Ship	2.999	4 (from 2 000 to 2 999 GT)
8	244674000	1	33	General Cargo Ship	5.418	7 (from 5 000 to 5 999 GT)
9	246199000	1	33	General Cargo Ship	2.056	4 (from 2 000 to 2 999 GT)
10	246546000	1	20	Cement Carrier	3.087	5 (from 3 000 to 3 999 GT)

...

7.1 F2 table for the port of Świnoujście

```

select f.shiptype_2d || '-' || f.shiptype_2d_per as "Type", f.gross_ton_klim as "Size class", sum(f.arrivals) as "Total number of
vessels (arrivals)", sum(f.arrivals*f.gross_ton) as "Total weight (in GT)"
from
(
select
a.mmsi, a.arrivals, b.shiptype_2d, b.shiptype_2d_per, b.gross_ton, b.gross_ton_klim
from
(select mmsi, count(mmsi) as arrivals from swinmovements_201703d2
where
veloc>0 and
(diflat2lat1 between -1000 and 0) and -----lat2<lat1 approaching to port entrance
(diflon2lon1 between -1000 and 1000) and ----- default values
(lon1 between -1000 and 1000) and ----- default values
(lon2 between -1000 and 1000) and ----- default values
(lat1 between 53.9220 and 53.9550 ) and ----- latitude of position 1 until the entrance of the port
(lat2 between 53.90 and 53.9219) ----- latitude of position 2 entering the port
group by mmsi order by mmsi) a,
emsaships_201703 b
where a.mmsi=b.mmsi and b.shiptype_2d is not null
order by a.mmsi
)f
group by f.shiptype_2d, f.gross_ton_klim, f.shiptype_2d_per

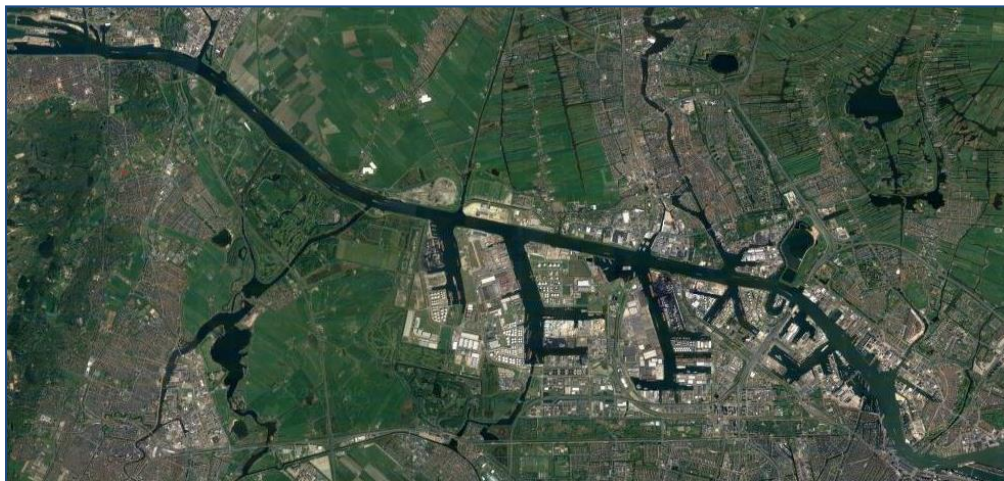
```

Output F2 table for the port of Świnoujście

Type	Size class	Total number of vessels (arrivals)	Total weight (in GT)
10-Chemical/Oil Products Tanker	5 (from 3 000 to 3 999 GT)	1	3478
20-Bulk Carrier	13 (from 20 000 to 29 999 GT)	2	46.523
20-Cement Carrier	5 (from 3 000 to 3 999 GT)	1	3.087
31-Container Ship	5 (from 3 000 to 3 999 GT)	1	3.999
32-Refrigerated Cargo Ship	4 (from 2 000 to 2 999 GT)	1	2.999
33-Passenger/Ro-Ro Cargo Ship	12 (from 10 000 to 19 999 GT)	8	129.641
33-Passenger/Ro-Ro Cargo Ship	13 (from 20 000 to 29 999 GT)	9	233.882
33-General Cargo Ship	3 (from 1 000 to 1 999 GT)	2	3.567

33-General Cargo Ship	4 (from 2 000 to 2 999 GT)	6	13.896
33-General Cargo Ship	7 (from 5 000 to 5 999 GT)	2	11.047
35-Passenger	1 (from 100 to 499 GT)	1	173

8 Port of Amsterdam



---- Create table **amslocs_201703d2** for Port of Amsterdam

```
create table amslocs_201703d2 as select * from emsalocs_201703d2
where lon>4.7298 and lon<4.8814 and lat>52.3878 and lat<52.4406;
```

---- Create indexes and geometry point field (geom) from coordinates to table **amslocs_201703d2**

```
create index ams_posindex on amslocs_201703d2 (mmsi,lon,lat,rec_time);
alter table amslocs_201703d2 add column geom geometry(point,4326);
update amslocs_201703d2 set geom=ST_SetSRID(ST_MakePoint(lon,lat),4326);
create index geom_amsindex on amslocs_201703d2 using GIST(geom);
```

--Create table **amsmovements_201703d2** (by enriching **amslocs_201703d2** table) with fields that show
 ----previous (geom1) and next (geom2) position of a ship,
 ----time needed to cover the distance between the two positions (duration_secs),
 ----distance covered by the ship from position1 to position2 (dist)

```
create table amsmovements_201703d2 as (
SELECT
mmsi,day_when,start_ts,end_ts,geom1,geom2,lat1,lat2,lon1,lon2,
extract(epoch from (end_ts - start_ts)) AS duration_secs,
st_distance(st_transform(geom1, 28992), st_transform(geom2, 28992)) AS dist,
st_makeline(geom1,geom2)::geometry(LineString, 4326) AS geo_segment
FROM
(SELECT
mmsi,date_trunc('day',rec_time) as day_when,
rec_time AS start_ts,
```

```

lead(rec_time) OVER w AS end_ts,
geom AS geom1,
lead(geom) OVER w AS geom2,
lat AS lat1,
lead(lat) OVER w AS lat2,
lon AS lon1,
lead(lon) OVER w AS lon2
FROM
amslocs_201703d2
WINDOW w AS (PARTITION BY mmsi, date_trunc('day',rec_time) ORDER BY rec_time)
) as q);

```

----- delete from table **ams**movements_201703d2 records that show no movement

```
delete from amsmovements_201703d2 where geom2 is null or dist=0; --6445 records deleted
```

----Enrich table **ams**movements_201703d2 with columns that show

---the velocity (veloc),

---the difference of latitude (diflat2lat1) and longitude (diflon2lon1) when a ship is moving from position1 to position2

```
alter table amsmovements_201703d2
```

```
add veloc numeric,
```

```
add diflat2lat1 numeric,
```

```
add diflon2lon1 numeric;
```

```
update amsmovements_201703d2
```

```
SET veloc=dist/duration_secs;
```

```
update amsmovements_201703d2
```

```
SET veloc=round(veloc,2);
```

```
update amsmovements_201703d2
```

```
SET diflat2lat1=(lat2-lat1)*1000;
```

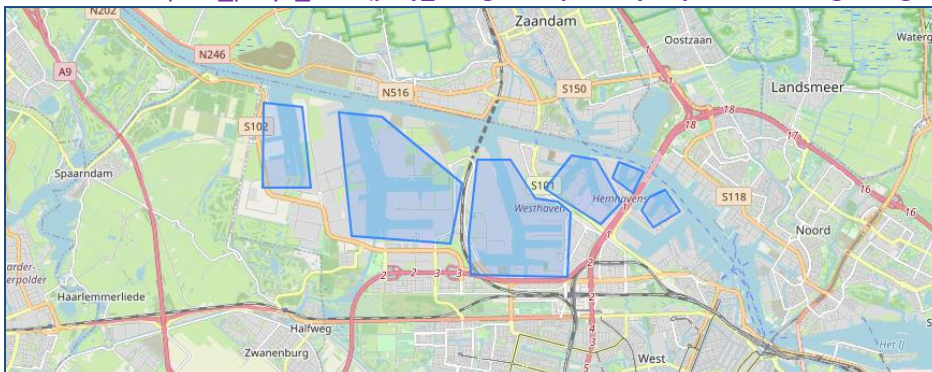
```
update amsmovements_201703d2
```

```
SET diflon2lon1=(lon2-lon1)*1000;
```

8.1 Tanker terminals

--- Create table and insert polygon areas – tanker terminals in port of Amsterdam

```
CREATE TABLE port_polys_ams (poly_id bigserial primary key, name text, geom geometry (polygon, 4326));
```



```
INSERT INTO port_polys_ams (name, geom)
```

```
select 'ams10',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.73962 52.4265,
4.75198 52.4256,
4.75472 52.4099,
4.73928 52.4099,
4.73962 52.4265))',4326));
```

```
INSERT INTO port_polys_ams (name, geom)
select 'ams20',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.76376 52.4247,
4.77738 52.4237,
4.80312 52.4107,
4.79935 52.399,
4.76788 52.4006,
4.76376 52.4247))',4326));
```

```
INSERT INTO port_polys_ams (name, geom)
select 'ams30',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.80782 52.4154,
4.81869 52.4155,
4.82647 52.4079,
4.83356 52.4073,
4.83791 52.4051,
4.83665 52.3925,
4.80598 52.393,
4.80507 52.4003,
4.80782 52.4154))',4326));
```

```
INSERT INTO port_polys_ams (name, geom)
select 'ams40',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.83825 52.4163,
4.84592 52.4154 ,
4.85393 52.407,
4.84741 52.4029,
4.82967 52.4098,
4.83825 52.4163))',4326));
```

```
INSERT INTO port_polys_ams (name, geom)
select 'ams50',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.85313 52.4149,
4.86102 52.413,
4.85753 52.4089,
4.85101 52.4119,
4.86305 52.4086,
4.85313 52.4149))',4326));
```

```
INSERT INTO port_polys_ams (name, geom)
select 'ams60',
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.86294 52.408
4.86812 52.4096
4.87295 52.4053
4.86405 52.4023
4.86039 52.4055
4.86294 52.408))',4326));
```

8.2 Tankers' arrivals at 7/3/2017

---Query to find vessels (key value mmsi) inside tanker terminals for 6 and 7 March 2017:

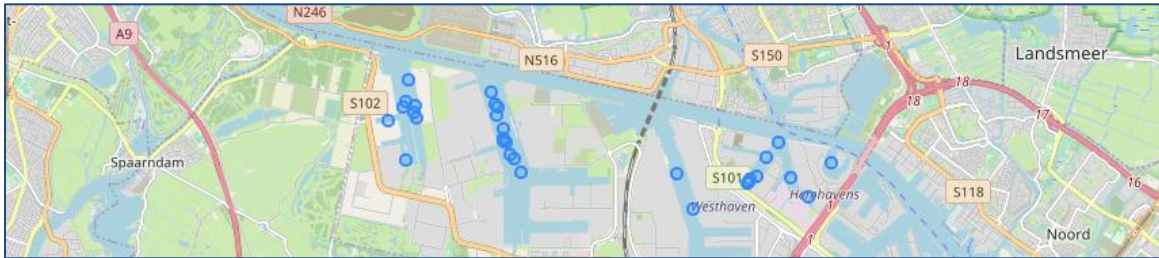
---Condition1 : AIS ship type 80 (=tankers) .

---Condition2 : The tankers have to be moving (velocity>0.2)

---Condition3 : The tankers are arriving at the terminal so one previous position (geom1) is outside terminal



-----and the following position (geom2) is inside the terminal



---Condition4 : Filter out inland waters' vessels which do not have imo number. (IMO number should have 7 digits)

```

SELECT a.mmsi, a.day_when, a.start_ts, a.end_ts,
       a.geom1, a.geom2, a.duration_secs,
       a.dist, a.geo_segment, a.veloc,
       pr.name,
       pr.geom,
       v.vessel_name,
       v.shiptype_ais
FROM   amsmovements_201703d2 a,
       port_polys_ams pr,
       emsships_201703 v
       where
         v.mmsi=a.mmsi and
         v.shiptype_ais='80' and
         length(v.imo)=7 and
         a.veloc>0.2 and
         st_contains(pr.geom, a.geom1) = false and
         st_contains(pr.geom, a.geom2) = true
       order by a.mmsi,a.start_ts

```

----- movements table

----- tanker terminals (polygons) table

----- vessels register (characteristics) table

--- AIS ship type 80 (=tankers)

--- Filter out inland waters' vessels

---tankers are moving

---Condition3

---Condition3

---Query selection of tankers (ais_type=80) that enter the terminals (polygon areas) on 7/3/2017

```

select al.t_mmsi, al.t_imo, al.t_name, al.day_observed
from
  (SELECT
    ams_t.v_mmsi as t_mmsi,
    ams_t.v_imo as t_imo,
    ams_t.v_name as t_name,
    ams_t.arrival_day AS day_observed,
    lead(ams_t.arrival_day) OVER w AS second_day,
    lag(ams_t.arrival_day) OVER w AS first_day
  FROM
    (select b.mmsi as v_mmsi, b.imo as v_imo, b.vessel_name as v_name, b.day_when as arrival_day
     from
       (SELECT a.mmsi, v.imo, a.day_when, a.start_ts, a.end_ts,

```



```

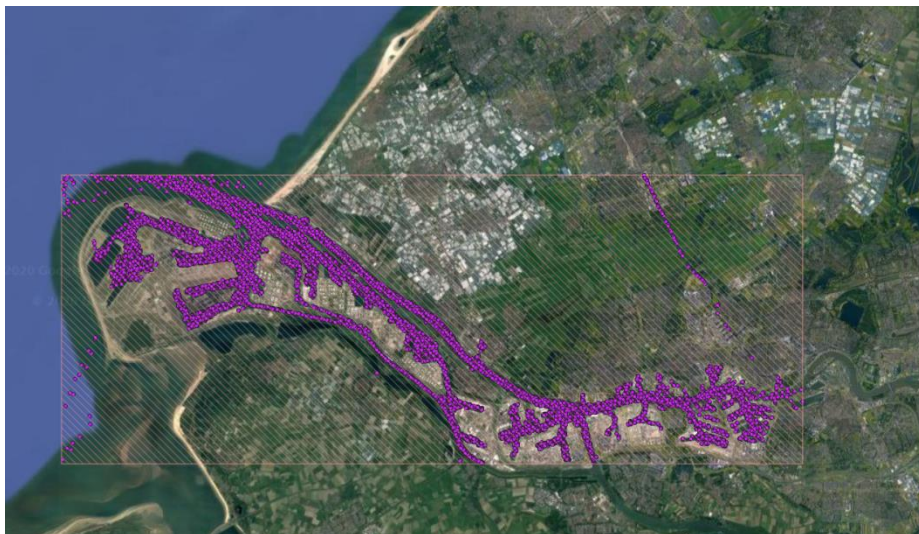
a.geom1, a.geom2, a.duration_secs,
a.dist, a.geo_segment, a.veloc,
pr.name,
pr.geom,
v.vessel_name,
v.imo,
v.shiptype_ais
FROM amsmovements_201703d2 a,      ---- movements table
port_polys_ams pr,                ---- tanker terminals (polygons) table
emsaships_201703 v                 ---- vessels register (characteristics) table
where
v.mmsi=a.mmsi and
v.shiptype_ais='80' and            --- AIS ship type 80 (=tankers)
length(v.imo)=7 and               --- Filter out inland waters' vessels
a.veloc>0.2 and                   ---tankers are moving
st_contains(pr.geom, a.geom1) = false and ---Condition3
st_contains(pr.geom, a.geom2) = true  ---Condition3
order by a.mmsi,a.start_ts
) b
group by b.mmsi, b.imo ,b.vessel_name,b.day_when
order by b.mmsi,b.vessel_name,b.day_when
) ams_t
WINDOW w AS (PARTITION BY ams_t.v_mmsi ORDER BY ams_t.v_mmsi)
) al
where
al.second_day is null and al.first_day is null ---exclude vessels observed at 6/3/2017
and al.day_observed='2017-03-07 00:00:00'

```

-Output—Arrivals of tankers at Port of Amsterdam's terminals (polygon areas) at 7/3/2017

a/a	mmsi	imo	vessel_name	arrival_day
1	215178000	9527635	MURRAY STAR	7/3/2017 0:00
2	245573000	9229051	THUN GLOBE	7/3/2017 0:00
3	248221000	9479802	KEY SOUTH	7/3/2017 0:00
4	249329000	9732694	HAFNIA LOTTE	7/3/2017 0:00
5	249512000	9388704	SICHEM EAGLE	7/3/2017 0:00
6	256210000	9301897	MARVEA	7/3/2017 0:00
7	259737000	9254977	VADERO HIGHLANDER	7/3/2017 0:00
8	538002776	9323364	USMA	7/3/2017 0:00
9	255806147	9396660	CLIO	7/3/2017 0:00

9 Port of Rotterdam



---- Create table `rotslocs_201703d2` for Port of Rotterdam

```
create table rotslocs_201703d2 as select * from emsalocs_201703d2
where lon>3.9491 and lon<4.4808 and lat>51.8695 and lat<51.9970;
```

---- Create indexes and geometry point field (`geom`) from coordinates to table `rotslocs_201703d2`

```
create index rot_posindex on rotslocs_201703d2 (mmsi,lon,lat,rec_time);
alter table rotslocs_201703d2 add column geom geometry(point,4326);
update rotslocs_201703d2 set geom=ST_SetSRID(ST_MakePoint(lon,lat),4326);
create index geom_rotindex on rotslocs_201703d2 using GIST(geom);
```

--Create table `rotmovements_201703d2` (by enriching `rotslocs_201703d2` table) with fields that show

-----previous (`geom1`) and next (`geom2`) position of a ship,
-----time needed to cover the distance between the two positions (`duration_secs`),
-----distance covered by the ship from position1 to position2 (`dist`)

```
create table rotmovements_201703d2 as (
SELECT
mmsi,day_when,start_ts,end_ts,geom1,geom2,lat1,lat2,lon1,lon2,
extract(epoch from (end_ts - start_ts)) AS duration_secs,
st_distance(st_transform(geom1, 28992), st_transform(geom2, 28992)) AS dist,
st_makeline(geom1,geom2)::geometry(LineString, 4326) AS geo_segment
FROM
(SELECT
mmsi,date_trunc('day',rec_time) as day_when,
rec_time AS start_ts,
lead(rec_time) OVER w AS end_ts,
geom AS geom1,
lead(geom) OVER w AS geom2,
lat AS lat1,
lead(lat) OVER w AS lat2,
lon AS lon1,
lead(lon) OVER w AS lon2
FROM
```


rotlocs_201703d2

```
WINDOW w AS (PARTITION BY mmsi, date_trunc('day',rec_time) ORDER BY rec_time)
) as q);
```

----- delete from table **rot**movements_201703d2 records that show no movement

delete from rotmovements_201703d2 where geom2 is null or dist=0; --25188 records deleted

----Enrich table **rot**movements_201703d2 with columns that show

---the velocity (veloc),

---the difference of latitude (diflat2lat1) and longitude (diflon2lon1) when a ship is moving from position1 to position2

alter table rotmovements_201703d2

add veloc numeric,

add diflat2lat1 numeric,

add diflon2lon1 numeric;

update rotmovements_201703d2

SET veloc=dist/duration_secs;

update rotmovements_201703d2

SET veloc=round(veloc,2);

update rotmovements_201703d2

SET diflat2lat1=(lat2-lat1)*1000;

update rotmovements_201703d2

SET diflon2lon1=(lon2-lon1)*1000;

---Insert to port_polys table terminal polygon for Rotterdam using coordinates

INSERT INTO port_polys (name, geom)

SELECT 'rotterdam',

ST_BuildArea (ST_GEOMFROMTEXT('polygon((3.9491 51.9970,

4.4808 51.9970,

4.4808 51.8695,

3.9491 51.8695,

3.9491 51.9970))',4326));



---Insert to port_polys table a test (small polygon) terminal polygon using coordinates

INSERT INTO port_polys (name, geom)

SELECT 'rot_poly1',

```
ST_BuildArea (ST_GEOMFROMTEXT('polygon((4.06879 51.97586,  
4.09591 51.95610,  
4.09695 51.93344,  
4.03072 51.92459,  
3.96187 51.95787,  
3.98925 51.98534,  
4.03332 51.98486,  
4.06879 51.97586  
)',4326));
```

-----Test of movements in 'rot_poly1' (small polygon)

```
SELECT a.mmsi, a.day_when, a.start_ts, a.end_ts,  
       a.geom1, a.geom2, a.lat1, a.lat2, a.lon1, a.lon2, a.duration_secs,  
       a.dist, a.geo_segment, a.veloc, a.diflat2lat1, a.diflon2lon1,  
       pr.name,  
       pr.geom  
FROM rotmovements_201703d2 a,  
     port_polys pr
```

```
where  
pr.name='rot_poly1' and  
a. day_when='2017-03-06 00:00:00' and  
a.veloc>0.7 and  
st_contains(pr.geom, a.geom2) = true  
order by a.mmsi,a.start_ts;
```

.....