Extending Access with SQL Server – including Geography data

An eclectic collection of examples, including -

- A map using Leaflet
- Persisted and non Persisted Computed Columns
- Scalar Functions
- Stored Procedures
- Triggers
- Spatial Indexes

https://accessusergroups.org/europe/event/access-europe-2023-11-01/

See also Colin's talk Annotating Google Maps in Access 2022/12

Map in new browser control using JS

- <head>
- <title>Access Europe Nov 2023</title>
- <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
- <script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>
- <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.3/dist/leaflet.css" integrity="sha256-kLaT2GOSpHechhsozzB+flnD+zUyjE2LlfWPgU04xyI=" crossorigin=""/>
- <style>#map {height: 700px;}</style>
- </head>
- <body>
-
- <script src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js" integrity = "sha256-WBkoXOwTeyKclOHuWtc+i2uENFpDZ9YPdf5Hf+D7ewM=" crossorigin=""></script>
- <script type="text/javascript">
- var map = L.map('map').setView([51.5, 4], 5);
- L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {attribution: '© OpenStreetMap contributors' }).addTo(map);
- L.marker([51.9,-2]).addTo(map).bindPopup(decodeURI('Person A')).openPopup();
- </script>
- </body>

Computed Column based on a Scalar Function

- Can define and run complex business rules on the server
- Computed column appears as a Table Field in Access
- Have to unreference the Function if you want to change it
  - `Cannot ALTER 'dbo.NextDrinkDt' because it is being referenced by object 'Person'.`

Non persisted column based on several tables – easy to use in Access but could be slow with large tables

```
-- Declare the return variable here
DECLARE @NextDrinkDt datetime
DECLARE @Person_LastMeetingDate datetime
DECLARE @IntervalDays smallint

-- Get the date of latest drink
Select  @NextDrinkDt = MAX(Consumption_Date)
FROM            dbo.Consumption
WHERE        (Consumption_Person_Id = @PersonId)

-- Get the Interval in Days between drinks, depending on addiction level.  Also get the date of Last  Meeting
Select @IntervalDays =  BeerAddiction.BeerAddiction_IntervalDays, @Person_LastMeetingDate = Person_LastMeetingDate
FROM         dbo.Person INNER JOIN dbo.BeerAddiction ON Person.Person_BeerAddiction_Id = BeerAddiction_Id
WHERE        (Person.Person_Id = @PersonId)

-- Next drink is interval days plus date of last drink (or last meeting date if no prev drink) or a fixed date if no history
Select  @NextDrinkDt = isnull(DATEADD (DAy, @IntervalDays, isnull(@NextDrinkDt, @Person_LastMeetingDate)),'2022-07-01')

RETURN @NextDrinkDt
```

Persisted Geography column based on single table – based on relatively static data

```
CREATE FUNCTION [dbo].[LatLngToPoint] (@LAT float, @LNG float) RETURNS geography
with schemabinding AS
BEGIN
DECLARE @RESULT geography
if @LAT is not null and @LNG is not null
SET @RESULT = [geography]::Point(@LAT,@LNG,4326)
RETURN  @RESULT
END
GO
```

Point (geography Data Type)

Spatial Reference defines the coordinate system – eg 4326

SSMS – Spatial results

- Limited use but can help to show that things are working
- Points not much use – Person table
- Areas can be shown - Person_PubAreaSpatialResults.sql
  - Counties
  - The display of Labels can be erratic !

Lots of Geography methods

CountiesInAreaOrderDesc.sql

Trigger to look up a point in a table of polygons

```
UPDATE Person
SET Person_County_Id = OSMap_CeremonialCounties.ID
FROM dbo.Person
INNER JOIN INSERTED
ON Person.Person_Id = INSERTED.Person_Id
LEFT JOIN DELETED
ON INSERTED.Person_Id = DELETED.Person_Id
LEFT JOIN dbo.OSMap_CeremonialCounties WITH (INDEX
(SPATIAL_OSMap_CeremonialCounties))
ON Person.Person_Point.STIntersects(OSMap_CeremonialCounties.Polygon) = 1
WHERE INSERTED.Person_Point IS NOT NULL
AND (
DELETED.Person_Point IS NULL
OR INSERTED.Person_Point.STEquals(DELETED.Person_Point) = 0
)
```

Spatial Indexes

- Use a spatial index to improve performance of **some** Geographic operations
- Until June 2023 we had to drop and create the index in order to relink the table in Access – thanks to Colin for getting this fixed, as creating the index can be VERY slow on Azure!!

```
CREATE SPATIAL INDEX [SPATIAL_OSMap_LocalAuthority] ON
[dbo].[OSMap_LocalAuthority]

([Polygon])USING  GEOGRAPHY_AUTO_GRID

WITH (CELLS_PER_OBJECT = 12, STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING = OFF,
ONLINE = OFF) ON [PRIMARY]
```

With Spatial index -12ms
Without index – 23135ms

```
set statistics io on
set statistics time on

SELECT UkPlace_Name, LA_Name, UkPlace_Population
FROM UkPlace
INNER JOIN OSMap_LocalAuthority
ON UkPlace_Point.STIntersects(OSMap_LocalAuthority.Polygon) = 1
WHERE UkPlace_Id<=2

SELECT  UkPlace_Name, LA_Name, UkPlace_Population
FROM UkPlace
INNER JOIN OSMap_LocalAuthorityNoIndex
ON UkPlace_Point.STIntersects(OSMap_LocalAuthorityNoIndex.Polygon) = 1
WHERE UkPlace_Id<=2
```

Don’t open Tables with Geography columns in Access

- Slow to open where a Geography field represents large polygons – Access can’t use the field anyway
- Use a View to return what you need

Sources of Geographic data – credit to Martin Newman

The OS boundary line data comes from here https://osdatahub.os.uk/downloads/open/BoundaryLine *The data is available as shape files and so these have to be imported into SQL server where it is stored as spatial polygon data using the WSG84 lat long coordinate system.*

Firstly, it may seem obvious to use dotMorten | Shape2SQL (sharpgis.net) . However this does *not* do the necessary coordinate transformations and *must not* be used. Instead use GDAL. Gdal.org does not have binaries, you have to build your own if you go there. However, https://www.gisinternals.com/ has windows binaries so I downloaded from there. Note that their GDAL is included with QGIS.

I used OGR2OGR, a command line app in GDAL to import to a local SQL database. This command creates the table as well as filling it with data

ogr2ogr -f MSSQLSpatial "MSSQL:Provider=SQLNCLI11;server=server\SQLservername;database=OSCeremonialCounties;trusted_connection=yes" "C:\somewhere\district_borough_unitary_region.shp" -t_srs "EPSG:4326”

-f means “output format” and the next few parameters define the sql server.
The local filename is obviously the data from the OS open data. Note that you need the .shp file and the .shx,.prj and.dbf files

-t_srs "EPSG:4326” defines the transform to WGS84 Lat/long (EPSG = SRID in MS parlance)

Other sources of spatial data – lots more!

- https://freegisdata.rtwilson.com/
- https://simplemaps.com/data/gb-cities