Assignment – Spatial Data and Asset / Facilities Management

Submission Date: see Moodle, **note pre-submission topic declaration**

Topic areas (maximum 3 students per topic area) can be selected on a first come first served basis

Submission Method (links to all forms on Moodle):

- *Pre-Submission –*
  - *Declare your topic title, description and the names, geometry types and dimensions of your three location-based, nested, assets - to avoid risks of plagiarism your topic area should come from an existing list, maximum of 3 students per topic area. Online form*
- *Submission*
  - *A PDF of your pyramid, uploaded via Moodle / Turnitin*
  - *Completing an online form with information about the decisions you are making*
  - *Five separate SQL scripts via an online form*
  - *A PDF of your map and 3D visualisation uploaded via Moodle/Turnitin*

This assignment is worth 70% of the marks for the module.

- *An assignment is an independent piece of work – your work should not be identical to, or similar to, that of any other student or the work we do in class. If two assignments are very similar we will follow UCL procedures for plagiarism – see the guidelines here: https://www.ucl.ac.uk/students/exams-and-assessments/plagiarism*

- *If you have questions about this assignment, please post them on Moodle*
  - *That way everyone is given the same information*
  - *That way I remember what I've said to you and don't mark you down for doing something that I wasn't expecting*
  - *Any questions should be generic – as this is an assessment which will gauge how much you've learned during the module I won't be able to solve very specific assignment-related problems for you.*

- *This is a digital submission – it is up to you to ensure that the files you upload to Turnitin or the online submission process are not corrupt in any way (in Turnitin you might be able to do this by downloading the uploaded files to check them,* ***for the online submissions you will receive an e-mail which you should keep as evidence of successful submission)***

NB: You are limited to a maximum of 10 e-mails per day to avoid overloading the testing system

Database Design and Build (70%)

Overview

The assignment involves the selection of an **asset management topic** of your choice for which you will create a hierarchical pyramid to show how the features (assets) and **decisions based on information** about those features nest upwards.

The pyramid should have 3 levels of spatial nesting

You will then make a list of 7 **asset-management decisions**. Two of the decisions must use data from lower level decisions (i.e. bottom-> middle and middle -> top).

You will then create the physical database for your organisation, insert some data and demonstrate using queries how this data can be used as evidence for the decisions.

Step 1 - Topic Pre-Selection/Declaration (online form)

Selecting a Topic Area and a Topic (link to online form will be given in Moodle)

You are required to select a topic area from the available list (link will be made available in Moodle), and then provide a topic title and list your three spatial asset tablenames and geometry types and dimensions. These must be nested (i.e. the top level contains the middle level and the middle level contains the bottom level).

There is no specific deadline for this task but you should make sure you lock in your topic BEFORE doing any other work on the assignment!

At least one of your pyramid geometries MUST be 3D[1]

Specifically[2]:

- The top level should be a 3D volume or a 2D or 3D area
- The middle level should be a 3D volume or a 2D or 3D area (provided it nests inside the top level using st_contains)
- The bottom level should be a 2D or 3D point or a 2D or 3D line or a 2D or 3D area or a 3D volume (provided it nests inside the middle level using st_contains)

To give you an idea of the types of assets/facilities that you might think about, in the past we have had assignments relating to mountain rescue huts, zoos, cycle hire, logistics/delivery organisations, holiday letting, infrastructure management companies, lost property offices, forest maintenance, highways agencies, water supply

[1] To ensure your pyramid is nested correctly, if you are using 3D geometry we will first take the 2D footprint and then check the nesting, as st_contains only works on 2D data

[2] You will also have other tables as part of your assignment – these can be spatial or non-spatial, and the spatial tables can be 2D or 3D

companies and many more. You may also find some case studies online using search terms such as built asset management, GIS, key performance indicators, KPI.

Your organisation may relate to your work, MSc degree, research or another area of interest. You could include decisions related to condition surveys, to short term replacement costs for an asset, to long term asset portfolio costs and purchasing / upgrade decisions. You could look at fire prevention, reactive building repairs, planned maintenance, human comfort. You could also look at information that would be useful to users of a facility.

You should not use a university or school example as this would be too similar to the example we covered in class and would be plagiarism

Step 2 - Decisions (online form)

List the Decisions (link to online form will be given in Moodle)

1. Write a description of the **7 decisions** that your database will support
    a. **The decisions need context** and should include some information as to what information is needed but also WHY the information is needed
        i. e.g. they should not just be 'list the total area of all the shops' (the 'what') – but should have some context –
            - 'we need to make sure that there are enough security staff to cover the mall, with a ratio of one security person per 100 sq m' (and then in the SQL calculate the area of the shops and divide by 100 to get the number of security staff) OR 'we need to find the largest shop so that we can speak to the owner about splitting their space as we need to find a location to store glass for some glass repairs'
            - 'we need to find the maintenance times for all the ovens so that we can compare that to the maintenance times of the new ovens we're planning to buy'.

        ii. In other words, you need to be clear on what the information will actually be used for.

2. You should have two decisions that build on information provided by a lower level query. You should make use of VIEWS to achieve this.
    a. Create a view for the decision corresponding to the bottom level of the pyramid, and for the decision query select from this view
    b. Create a view for the decision corresponding to the middle level of the pyramid that includes data from the view from the bottom level of the pyramid, and for the decision query select from this view
    c. Create a view for the decision corresponding to the top level of the pyramid, that includes data from the middle level view, and for the decision query select from this view

Step 3 – Pyramid (turnitin upload)

1. Create a THREE LEVEL pyramid for this organisation. This should include the names of the spatial asset/features that are at the bottom, middle and top levels of your pyramid
    a. These names must be IDENTICAL to the name of the table that stores data for this feature.
    b. All table names should be lower case
    c. All three tables (assets) must match the tables you declared in the pre-submission stage
    d. All three features must be spatial – i.e. mappable (they can be 2D or 3D depending on how they were originally declared)

e. The features must nest – i.e. features at the lowest level must be contained inside features at the middle level, and features at the middle level contained inside features at the upper level.

Include a cover sheet in this PDF as Moodle requires a minimum of 35 words for plagiarism checks.

Step 4 - Database creation (5 script files, online form)

For this part of the assignment:

1. Script creation as follows:
   a. Create an SQL script that contains the SQL used to create the tables in your database system. Call this script: **createtable_ucfsxxx.**txt (where ucfsxxx is your UCL login). All table names should be lower case, use _ to separate out any words (not spaces or -)
      - Make sure you use addGeometryColumn to add location columns – the SQL testing scripts will not work if this is not present

   b. Create a separate SQL script for all the constraints. Call this script: **createconstraints_ucfsxxx.txt**

   c. Create a separate SQL script to populate each table with data - call this script **insertdata_ucfsxxx.txt**:
      - A minimum of 1 row of data for the top level of the pyramid
      - A minimum of 3 rows of data for the middle level (nested within the top level data)
      - A minimum of 9 rows of data for the bottom level (nested within the 3 rows of the middle level)
      - A minimum of THREE rows of data for all other tables.

      The data should be sufficient to allow you to test out the SQL for your listed decisions. **The resulting data for each decision should contain at least one row of data**

      So that we can test your work independently, your SQL must create ALL the data required from scratch – don't use any data that has been imported via QGIS / sourced from third parties. The spatial data you create does not need to be very sophisticated in terms of geometry complexity.

      All data should use a projected coordinate system (i.e. not lat/lng[3] - if you don't know the projected coordinate system for your location use British National Grid)

[3] As lat/lng units are degrees/minutes/seconds when you try and measure a length or area you get a very strange answers

d. Upload a script that creates a series of views that will help to simplify your queries – minimally, you should have 3 views, one for each level of the pyramid that aggregates from the lower levels. Call this ***createviews_ucfsxxx.txt***

e. Create a script file with the 7 SQL queries that provide data used to support the decisions you listed in the first part of the assignment. Call this ***decisions_ucfsxxx.txt***

The answers to each decision should contain at least one row of data

2. Use the provided test system (link on Moodle) to upload and test your scripts. You will receive an e-mail report each time you run the test. Keep this as evidence that you have submitted your scripts.

Some hints

- The SQL scripts should be manually created – i.e. typed by you
- You must use the provided PostGIS database
- The first line of the createtable script should drop cascade the ucfsxxx schema (if it exists) and create it again
- All the SQL should work in your schema – e.g. your create table scripts should be similar to: *create table ucfsxxx.buildings (……),* your insert scripts *insert into ucfsxxx.buildings* and queries *select from ucfsxxx.buildings*
- Do not include views in the decision queries – use a separate views file
- Include comments in your scripts to make them easy to read – use -- to mark the comments
  - ** Any comments should be prefixed by -- (not /* */)
- Each element (create table, constraint, row insert, decision query etc) in the script should be separated by a ;
- Make sure that the filenames are correct. The filename should contain the text e.g. createtable_ucxxxxx etc with your UCL username and all files should be .txt format (not rtf or doc or similar)

Step 5 – Map and 3D visualisation and queries (one PDF via Turnitin)

Upload a PDF to Moodle containing a **QGIS screenshot** showing the spatial data you create and a **3D screenshot**. The screenshots should include the entire QGIS/FME windows (including menus and layer list) so that it clearly shows that the data is spatial and the map has been created from data in the database.

Marking Process

This assignment will be marked semi-automatically (nearly fully automatically), so it is very important that you follow instructions to the letter. Automated marking includes (but is not limited to):

- Running all the SQL scripts you upload and creating your database – this will NOT be done on the system we use in class, as we have a separate test database for the purpose so it is important to make sure your scripts run from beginning to end
    - Don't forget to drop cascade and create the schema in the createtable script
    - Don't forget to use your UCL login ucxxxxx as the schema name
    - Make sure that all your tables and views are created in the schema

- Making sure you use the exact table names for the three assets that you declared in the initial form
- Checking that the geometry for the three levels of your pyramids actually does nest as expected
- Checking that all your tables have
    - Primary keys
    - Unique constraints
- Checking that your data is valid – e.g. that you have the required rows of data in all the tables, that primary/foreign key constraints are enforced, that geometry is correct

Marking Scheme

Marks will be awarded as follows:

** NB – if your assignment is too close to that covered in class we may reduce your overall mark. If your files don't work with the automated marking system we may reduce your mark **

Component	Maximum Mark out of 70	Comment/Hint (in addition to information above)
Topic declaration (via online form)	4	You must make sure that the table names and geometry types and dimensions you provide here are those you use in the final SQL. 0 marks for this component if this is not the case. Make sure at least one geometry is 3D and that the geometry nests as required.
List of decisions (via online form)	7	Marks will be deducted if your decisions aren't actually decisions but are just lists of data. Make sure that an actual decision is being made and is clearly described and is linked to one of the aims. For example – 'identify all the rooms in the university with details of all the buildings' is not a decision. However 'identify all the rooms in the university with details of all the buildings, as we need to know which buildings contain the largest rooms so that we can decide which building to prioritise for painting, as we need to have large rooms available at the start of next term' is a decision about where to paint first.
Pyramid (upload to Moodle) Map and 3D visualization (upload to Moodle)	6	Pyramid diagram must include all the components shown in the Centennial example (e.g. list of decisions, hierarchy of the assets). QGIS screenshot must include the layer list clearly showing the colours for each layer (i.e. the entire QGIS screen, not just the map). The 2D map must show all three of your spatial features each nested within the other. The 3D visualisation must show all three of your spatial layers - i.e. provide a screenshot of all of the FME screen
SQL – CREATE TABLE	3	0 if the script does not run as an entire script OR you do not use the correct schema, or if you don't DROP CASCADE and CREATE the schema first.
SQL – CONSTRAINTS	3	0 if the script does not run as an entire script OR you do not use the correct schema. This means that if your CREATE TABLE script has not worked, you could also get 0 for this component.

SQL – INSERT DATA	5	0 if the script does not run as an entire script OR you do not use the correct schema. This means that if your CREATE TABLE or CONSTAINTS scripts have not worked, you could also get 0 for this component. Your data must include 3D spatial data. The data does not need to be complex – e.g. you can create simple boxes for the geometry. You MUST have at least three tables that contain spatial data, matching the levels of your pyramid
Parameters Table	7	1 mark per query that makes use of the parameters table either via a nested query or a join (0 marks if the parameters table is not created and populated correctly or is not used). During testing we may manually insert new values into the parameters table to make sure your queries use the latest values of the parameters.
7 queries	35	Maximum of 5 marks per query depending on complexity. Queries that don't work will be awarded 0 marks. Queries where the answers don't provide useful information for the corresponding decision will be awarded 0 marks (for useful information, think about results that a decision maker who *doesn't know anything about your database* will be able to use). Examples of complexity include (but are not limited to): - Queries that JOIN information from 5 tables or more - Queries that make extensive use of spatial functions Examples of answers that are not useful: a single column that is called count but doesn't say what it is counting; a list of ID values without any detail of what they are.