HEAT WAVE

1. WATER
2. Check if input city has input data already generated
3. Extract Urban atlas code geometries to a shapefile
   * 1. Water (code: 50000)
4. Import shapefile to database
5. Geometry integrity check
   * 1. SQL query to check if geometries are valid
     2. If any not valid detected, then delete them
6. Add city ID column to imported table and add FK to related city table for it
7. Add cell ID column to imported table and add FK to related European grid table for it
8. Seek for gridded table, delete it if exists
9. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
10. Drop imported table
11. Adding columns:
    * 1. Albedo 0.07
      2. Emissivity 0.96
      3. Transmissivity 0.5
      4. vegetation shadow 1
      5. run off coefficient 0.1
12. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
13. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
14. Put all data generated for the specific city into final water table

1. ROADS
2. Check if input city has input data already generated
3. Extract Urban atlas codes geometries to a shapefile
   * 1. Fast transit roads and associated land (code: 12210)
     2. Other roads and associated land (code: 12220)
4. Import shapefile to database
5. Geometry integrity check
   * 1. SQL query to check if geometries are valid
     2. If any not valid detected, then delete them
6. Add city ID column to imported table and add FK to related city table for it
7. Add cell ID column to imported table and add FK to related European grid table for it
8. Seek for gridded table, delete it if exists
9. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
10. Drop imported table
11. Adding columns:
    * 1. Albedo 0.1
      2. Emissivity 0.9
      3. Transmissivity 0.15
      4. vegetation shadow 1
      5. run off coefficient 0.9
12. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
13. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
14. Put all data generated for the specific city into final roads table
15. RAILWAYS
16. Check if input city has input data already generated
17. Extract Urban atlas codes geometries to a shapefile
    1. Railways and associated land (12230)
18. Import shapefile to database
19. Geometry integrity check
    * 1. SQL query to check if geometries are valid
      2. If any not valid detected, then delete them
20. Add city ID column to imported table and add FK to related city table for it
21. Add cell ID column to imported table and add FK to related European grid table for it
22. Seek for gridded table, delete it if exists
23. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
24. Drop imported table
25. Adding columns:
    * 1. Albedo 0.2
      2. Emissivity 0.85
      3. Transmissivity 0.15
      4. vegetation shadow 1
      5. run off coefficient 0.2
26. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
27. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
    1. Adding hill shade building
       1. By default, set to 1
       2. If there is intersection with public military industrial set it to 0.9
       3. If there is intersection with low urban fabric set it to 0.9
       4. If there is intersection with medium urban fabric set it to 0.8
       5. If there is intersection with dense urban fabric set it to 0.6
28. Put all data generated for the specific city into final railways table
29. TREES
30. Check if input city has input data already generated
31. Extract Urban atlas codes geometries to a shapefile
    * 1. Forests (code: 31000 UA2012)
      2. Semi-natural areas (code: 30000 UA2006)
32. Import shapefile to database
33. Import STL shapefile data to same database table
34. Geometry integrity check
    * 1. SQL query to check if geometries are valid
      2. If any not valid detected, then delete them
35. Add city ID column to imported table and add FK to related city table for it
36. Add cell ID column to imported table and add FK to related European grid table for it
37. Seek for gridded table, delete it if exists
38. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
39. Drop imported table
40. Compute differences with previous layers
    * 1. Water
      2. Roads
      3. Railways
41. Geometry fixing to remove errors from geometries
42. Adding columns:
    * 1. Albedo 0.13
      2. Emissivity 0.97
      3. Transmissivity 0.25
      4. vegetation shadow 0
      5. run off coefficient 0.05
43. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
44. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
45. Adding hill shade green fraction
    * 1. Default is 0.37
46. Put all data generated for the specific city into final trees table
47. VEGETATION
48. Check if input city has input data already generated
49. GRASS setup
50. Extract Urban atlas codes geometries to a shapefile
    * 1. Green urban areas (code: 14100)
      2. Sports and leisure facilities (code: 14200)
      3. Herbaceous vegetation associations (code: 32000)
      4. Open spaces with little or no vegetations (code: 33000)
51. ESM Raster reclassifying: BU Area - Green NDVIx (band n. 40)
    * 1. Tress hold 25
52. Polygonization with GRASS
53. Import UA extraction to database
54. Import ESM polygonised to database together with UA data
55. Geometry integrity check
    * 1. SQL query to check if geometries are valid
      2. If any not valid detected, then delete them
56. Add city ID column to imported table and add FK to related city table for it
57. Add cell ID column to imported table and add FK to related European grid table for it
58. Seek for gridded table, delete it if exists
59. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
60. Drop imported table
61. Compute differences with previous layers
    * 1. Water
      2. Roads
      3. Railways
      4. Trees
62. Geometry fixing to remove errors from geometries
63. Adding columns:
    * 1. Albedo 0.21
      2. Emissivity 0.96
      3. Transmissivity 0.30
      4. vegetation shadow 1
      5. run off coefficient 0.18
64. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
65. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
66. Put all data generated for the specific city into final vegetation table
67. AGRICULTURAL AREAS
68. Check if input city has input data already generated
69. Extract Urban atlas codes geometries to a shapefile
    * 1. Arable land (annual crops) (code: 21000)
      2. Permanent crops (code: 22000)
      3. Pastures (code: 23000)
      4. Complex and mixed cultivation patterns (code: 24000)
      5. Orchards (code: 25000)
70. Import UA extraction to database
71. Geometry integrity check
    * 1. SQL query to check if geometries are valid
      2. If any not valid detected, then delete them
72. Add city ID column to imported table and add FK to related city table for it
73. Add cell ID column to imported table and add FK to related European grid table for it
74. Seek for gridded table, delete it if exists
75. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
76. Drop imported table
77. Compute differences with previous layers
    * 1. Water
      2. Roads
      3. Railways
      4. Trees
      5. Vegetation
78. Geometry fixing to remove errors from geometries
79. Adding columns:
    * 1. Albedo 0.11
      2. Emissivity 0.95
      3. Transmissivity 0.30
      4. vegetation shadow 1
      5. run off coefficient 0.1
80. Adding FUA tunnel
    * 1. Default is 1
      2. if there is intersection with dense urban fabric set it to 1.2
      3. if there is intersection with medium urban fabric set it to 1.1
81. Adding building shadow
    * 1. Default is 1
      2. changing it to 0 if it intersects layers 9-12
82. Put all data generated for the specific city into final agricultural areas table
83. BUILT UP
84. Check if input city has input data already generated
85. GRASS setup
86. ESM Raster reclassifying: U Buildings (Band n. 50)
    * 1. Tress hold 45
87. Polygonization with GRASS
88. Import UA extraction to database
89. Import ESM polygonised to database together with UA data
90. Geometry integrity check
    * 1. SQL query to check if geometries are valid
      2. If any not valid detected, then delete them
91. Add city ID column to imported table and add FK to related city table for it
92. Add cell ID column to imported table and add FK to related European grid table for it
93. Seek for gridded table, delete it if exists
94. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
95. Drop imported table
96. Compute differences with previous layers
    * 1. Water
      2. Roads
      3. Railways
      4. Trees
      5. Vegetation
      6. Agricultural areas
97. Geometry fixing to remove errors from geometries
98. Removing unneeded columns
99. Adding columns:
    * 1. Albedo 0.2
      2. Emissivity 0.85
      3. Transmissivity 0.01
      4. vegetation shadow 1
      5. run off coefficient 0.9
100. Adding FUA tunnel
     * 1. Default is 1
       2. if there is intersection with dense urban fabric set it to 1.2
       3. if there is intersection with medium urban fabric set it to 1.1
101. Adding building shadow
     * 1. Default is 1
       2. changing it to 0 if it intersects layers 9-12
102. Put all data generated for the specific city into final built up table
103. BUILT OPEN SPACES
104. Check if input city has input data already generated
105. GRASS setup
106. ESM Raster reclassifying: BU Area - Open Space (band n. 30)
     * 1. Tress hold 30
107. Polygonization with GRASS
108. Import ESM polygonised to database together with UA data
109. Geometry integrity check
     * 1. SQL query to check if geometries are valid
       2. If any not valid detected, then delete them
110. Add city ID column to imported table and add FK to related city table for it
111. Add cell ID column to imported table and add FK to related European grid table for it
112. Seek for gridded table, delete it if exists
113. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
114. Drop imported table
115. Compute differences with previous layers
     * 1. Water
       2. Roads
       3. Railways
       4. Trees
       5. Vegetation
       6. Agricultural areas
       7. Built up
116. Geometry fixing to remove errors from geometries
117. Removing unneeded columns
118. Adding columns:
     * 1. Albedo 0.45
       2. Emissivity 0.9
       3. Transmissivity 0.05
       4. vegetation shadow 1
       5. run off coefficient 0.75
119. Adding FUA tunnel
     * 1. Default is 1
       2. if there is intersection with dense urban fabric set it to 1.2
       3. if there is intersection with medium urban fabric set it to 1.1
120. Adding building shadow
     * 1. Default is 1
       2. changing it to 0 if it intersects layers 9-12
     1. Adding hill shade building
        1. By default, set to 1
        2. If there is intersection with public military industrial set it to 0.9
        3. If there is intersection with low urban fabric set it to 0.9
        4. If there is intersection with medium urban fabric set it to 0.8
        5. If there is intersection with dense urban fabric set it to 0.6
121. Put all data generated for the specific city into final built open spaces table
122. DENSE URBAN FABRIC
123. Check if input city has input data already generated
124. Extract Urban atlas codes geometries to a shapefile
     * 1. Continuous Urban fabric (S.L. > 80%) (code: 11100)
       2. Discontinuous Dense Urban Fabric (S.L.: 50% - 80%) (code: 11210)
125. Import UA extraction to database
126. Geometry integrity check
     * 1. SQL query to check if geometries are valid
       2. If any not valid detected, then delete them
127. Add city ID column to imported table and add FK to related city table for it
128. Add cell ID column to imported table and add FK to related European grid table for it
129. Seek for gridded table, delete it if exists
130. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
131. Drop imported table
132. Compute differences with previous layers
     * 1. Water
       2. Roads
       3. Railways
       4. Trees
       5. Vegetation
       6. Agricultural areas
       7. Built up
       8. Built open spaces
133. Geometry fixing to remove errors from geometries
134. Adding columns:
     * 1. Albedo 0.065
       2. Emissivity 0.9
       3. Transmissivity 0.01
       4. Run off coefficient 0.7
       5. context 1
135. Put all data generated for the specific city into final dense urban fabric table
136. MEDIUM UBRAN FABRIC
137. Check if input city has input data already generated
138. Extract Urban atlas codes geometries to a shapefile
     * 1. Discontinuous Medium Density Urban Fabric (S.L.: 30% - 50%) (code: 11220)
139. Import UA extraction to database
140. Geometry integrity check
     * 1. SQL query to check if geometries are valid
       2. If any not valid detected, then delete them
141. Add city ID column to imported table and add FK to related city table for it
142. Add cell ID column to imported table and add FK to related European grid table for it
143. Seek for gridded table, delete it if exists
144. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
145. Drop imported table
146. Compute differences with previous layers
     * 1. Water
       2. Roads
       3. Railways
       4. Trees
       5. Vegetation
       6. Agricultural areas
       7. Built up
       8. Built open spaces
147. Geometry fixing to remove errors from geometries
148. Adding columns:
     * 1. Albedo 0.11
       2. Emissivity 0.9
       3. Transmissivity 0.02
       4. Run off coefficient 0.5
       5. context 0.8
149. Put all data generated for the specific city into final medium urban fabric table
150. LOW URBAN FABRIC
151. Check if input city has input data already generated
152. Extract Urban atlas codes geometries to a shapefile
     * 1. Discontinuous Low Density Urban Fabric (S.L.: 10% - 30%) (code: 11230)
       2. Discontinuous very low density urban fabric (S.L. < 10%) (code: 11240)
       3. Isolated Structures (code: 11300)
153. Import UA extraction to database
154. Geometry integrity check
     * 1. SQL query to check if geometries are valid
       2. If any not valid detected, then delete them
155. Add city ID column to imported table and add FK to related city table for it
156. Add cell ID column to imported table and add FK to related European grid table for it
157. Seek for gridded table, delete it if exists
158. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
159. Drop imported table
160. Compute differences with previous layers
     * 1. Water
       2. Roads
       3. Railways
       4. Trees
       5. Vegetation
       6. Agricultural areas
       7. Built up
       8. Built open spaces
161. Geometry fixing to remove errors from geometries
162. Adding columns:
     * 1. Albedo 0.15
       2. Emissivity 0.9
       3. Transmissivity 0.05
       4. Run off coefficient 0.4
       5. context 0.5
163. Put all data generated for the specific city into final low urban fabric table
164. PUBLIC MILITARY INDUSTRIAL
165. Check if input city has input data already generated
166. Extract Urban atlas codes geometries to a shapefile
     * 1. Industrial, commercial, public, military and private units (12100)
167. Import UA extraction to database
168. Geometry integrity check
     * 1. SQL query to check if geometries are valid
       2. If any not valid detected, then delete them
169. Add city ID column to imported table and add FK to related city table for it
170. Add cell ID column to imported table and add FK to related European grid table for it
171. Seek for gridded table, delete it if exists
172. Generate gridded geometries table by SQL query by taking the current city cells by intersecting the city bounding box and the European grid, then doing intersection of those city cells with all imported geometries and taking the union of geometries intersecting each cell
173. Drop imported table
174. Compute differences with previous layers
     * 1. Water
       2. Roads
       3. Railways
       4. Trees
       5. Vegetation
       6. Agricultural areas
       7. Built up
       8. Built open spaces
175. Geometry fixing to remove errors from geometries
176. Adding columns:
     * 1. Albedo 0.13
       2. Emissivity 0.9
       3. Transmissivity 0.05
       4. Run off coefficient 0.5
       5. context 0.5
177. Put all data generated for the specific city into final public military industrial table