HEAT WAVE INPUT LAYERS

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