```
1  def Marker(Segments, radius = 100):
2  # OSM Marker
3    global connection
4    connection = ConnectionHandler() # handles Python <> PostgreSQL DB
5
6    for Segment in Segments:
7      MarkSegment(Segment, radius)
8    return Segments
9
10 def MarkSegment(Segment, radius):
11   for (i, distinct_location) in Segment.DistinctLocations:
12     nearby_vector = connection.query_location(distinct_location, radius)
13     Segment.mark_with_vector(nearby_vector, i)
14
15 class ConnectionHandler:
16 def query_location(self, location, radius):
17   sql_command = self.sql_cmd_radius(location, radius)
18   # builds the query
19   # SELECT <interesting_columns> FROM table WHERE distance < radius
20
21   rows, column_names = self.run_command( sql_command )
22
23   pairs = extract_all_pairs(rows, column_names)
24   # returns pairs of column names and their values
25   # for example "highway=primary", ...
26
27   nearby_vector = [0] * number_of_observed_pairs
28
29   for pair in pairs
30     if pair in observed_pairs:
31       index = indice_dictionary[pair]
32       nearby_vector[ index ] += 1
33   return nearby_vector
```