

112673790 Yatna Verma

Ques 2. See image q2.png

Ques 3. See image q3_screen_shot_1.png and q3_screen_shot_2.png

Ques 4.

Ques No	Without Index	With Index
2	22.49.56.618967 - 22.49.56.294112 ~ 0.61- 0.29 ~ 0.32 sec	22.51.38.765688 - 22.51.38.664838 ~ 0.76 - 0.66 ~ 0.10 sec
3	19.09.27.304521 - 19.08.44.205969 ~ 9.27 - 8.44 ~ 43 sec	19.16.12.433251 - 19.16.05.662286 ~16.12 - 16.05 ~ 7 sec

Ques 5

Pseudo code for merging zip codes

Step 1. Get the zip with lowest population (Lets call it x)

Step 2. If (x's population > average pop) GOTO END

Else GOTO *step 3*

Step 3. Get x's neighbour and select the one with the lowest population (lets call it n)

Step 4 Update x's population = x's population + n's population

x's shape = x's shape U n's shape

Step 5 Delete n's entry

Step 6 GOTO *step 1*

Step 7 END

BENEFIT OF ABOVE ALGORITHM -

Always ensure that zip code with minimum population zip merges with its lowest populated neighbour. Thus the 2 conditions stated by instructor on piazza i.e -

1) the merge should be as few zips as possible, and/or

2) if there are multiple neighboring zip candidates to merge and all of them qualify, we select the ones that have least aggregate after the merging.

Both these conditions are satisfied.

How to run -

1. Run 'db2 -tvf mergezip.sql' which does the following
 - Calculates average for zip in uszip table after removing zip with 0 population.
 - Create table my_dup for performing further operation
 - Creates indexes on my_dup
2. Set `iterations` variable to the desired no. of iterations (100 take >10min on my machine)
Or specify a very high number so that the loop stops only when all zips are > avg population.
3. Run 'db2 -td@ -f stored_procedure_q5.sql' to create SP `merge_neighbour(?)`.
4. Call procedure 'db2 "call merge_neighbour(?)'

cse532.my_dup Table Explanation

<i>zip</i>	<i>pop</i>	<i>shape</i>	<i>m_zip</i>	<i>is_m</i>
zip code	Population of zip code (merged population if zipcodes are combined)	shape of zip code (merged shape if zipcodes are combined)	List of zip codes merged	1 - not merged yet (default) 2 - merged 0 - no neighbour (can't be merged)

Results after running 300 iterations

1. See *q5_average_pop.png* for average population calculated (9383.82)
2. See *q5_merged_zipcodes_after_300_iter.png* for a list of zip codes merged and their merged population after 300 iterations or *q5_merged_after_300_iter.txt* for list of zipcodes which were either merged (`is_m=2`) or which had no neighbours to merge (`is_m=0`).

```

~ -- db2inst1@9f6b7d984cfb:/database/todb/hw3 -- docker exec -it db2 bash -c su - db2inst1
[db2inst1@9f6b7d984cfb hw3]$ db2 "select zip, pop, varchar(m_zip,50) as merged_zip_codes from cse532.my_dup where is_m=2 order by length(m_zip) desc"

```

ZIP	POP	MERGED_ZIP_CODES
10173	16581	10173,10170,10165,10017
82630	103	82630,82630,82646,82648
76573	3361	76573,76523,76511
19732	7442	19732,19735,19807
88417	93	88417,88421,88431
43463	37020	43463,43551
53501	18581	53501,53548
42037	1685	42037,42411
43786	1172	43786,45745
41366	10977	41366,41339
84515	61	84515,84540
95974	282	95974,95920
37851	514	37851,48940
14168	1627	14168,14129
49084	31850	49084,49022
76824	594	76824,76832
58505	28057	58505,58501
69171	4515	69171,69138
17039	5308	17039,17073
19112	12272	19112,19153
69353	2388	69353,69334
12862	286	12862,12886
12864	1202	12864,12842
12872	90	12872,12858
64065	1789	64065,64139
36859	1311	36859,36871
36865	22094	36865,36832
12911	4116	12911,12944
12933	1091	12933,12934
21031	24359	21031,21030
36910	691	36910,36922
34101	10470	34101,34102
78851	68	78851,78871
32830	13718	32830,34747
36616	13236	36616,36610
80938	621	80938,80927
98430	4852	98430,98439
65784	773	65784,65760
30275	53027	30275,30263
20535	1630	20535,20004
81021	102	81021,81045
30334	5035	30334,30303
50839	2058	50839,50841
18936	27874	18936,19454
36766	1121	36766,36768
96061	161	96061,96063
24570	3073	24570,24556
24581	629	24581,24599
30451	10910	30451,30439
96155	45	96155,95721

- See *q5_lowest_50_before_300_iter.png* for top 50 zip codes with lowest population before merging and *q5_lowest_50_after_300_iter.png* for top 50 zip codes with lowest population after merging for **300 iterations**.

BEFORE

```
~ — db2inst1@9f6b7d984cfb:/database/todb/hw3 — docker
[db2inst1@9f6b7d984cfb hw3]$ db2 "select zip, pop from cse532"

ZIP      POP
-----
30334    1
33122    1
38702    1
91371    1
10165    2
10170    2
10173    2
10911    2
20701    2
32399    2
54641    2
58505    2
79942    2
10103    3
63045    3
66118    3
90095    3
18936    4
21031    4
78335    4
98353    4
72119    5
74477    5
98641    5
14893    6
56658    6
79105    6
93943    6
20118    7
35082    7
38132    7
76949    7
88268    7
99695    7
20535    8
21705    8
24130    8
95974    8
96155    8
10199    9
11973    9
59081    9
28280    10
34101    10
55450    10
79058    10
79353    10
81227    10
82630    10
99732    10

50 record(s) selected.
```

AFTER

```
[[db2inst1@9f6b7d984cfb hw3]$ db2 "select zip, pop"

ZIP      POP
-----
98353    4
78335    4
79105    6
99695    7
92338    12
99638    18
56741    23
99767    24
43446    25
97604    28
99726    30
99583    35
77440    37
78060    37
78338    37
82845    37
88055    37
64066    38
25048    38
35577    38
30664    38
59547    38
38958    38
55029    38
35990    38
32530    38
97712    38
97817    38
68969    38
18912    39
17742    39
59545    39
85341    39
82061    39
82242    39
43336    39
59082    39
88264    39
88414    39
56146    40
93064    40
79326    40
84304    40
47536    40
45145    40
50227    40
59276    40
76836    41
38704    41
63962    41

50 record(s) selected.
```