

## 1. Software Installation Requirements

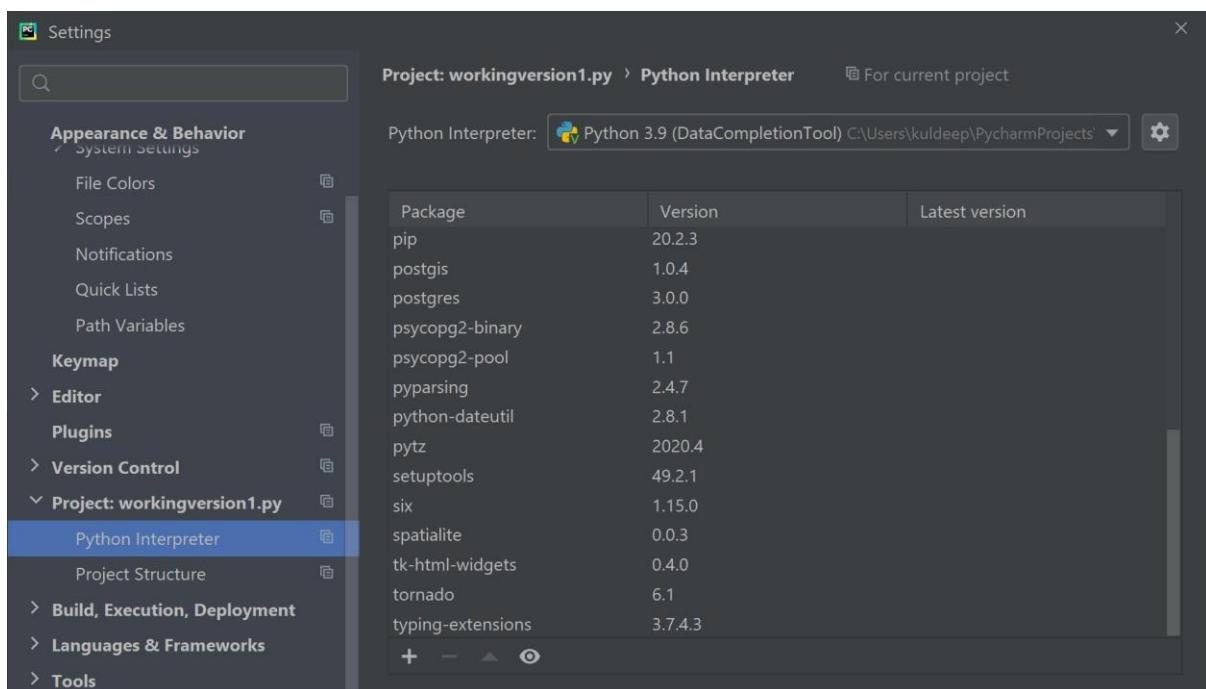
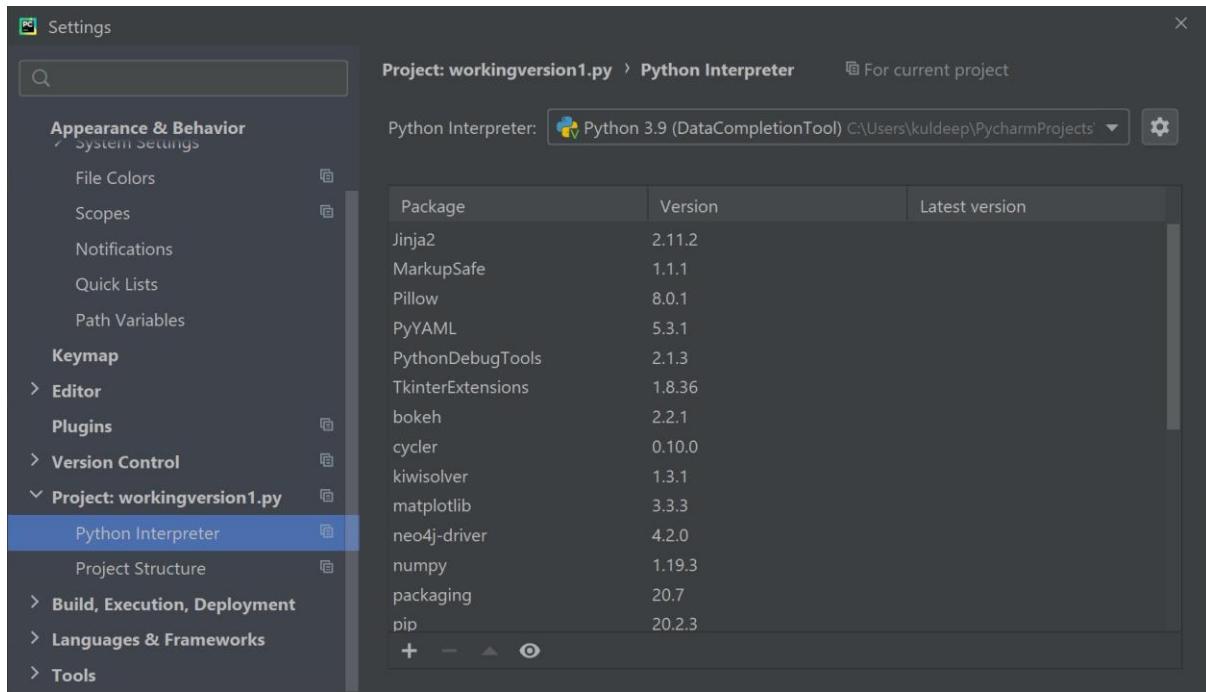
The main softwares required are shown below.

- i. Java 11.0.9 with jre 8
- ii. Python 3.9.1
- iii. Postgis Bundle 3.0.3 for PostgreSQL 10
- iv. Psycopg2 driver 2.8.6
- v. Matplotlib 3.3.3

The below image provides more details on the installed softwares.

Name	Installed On	Size	Version
Google Chrome	03-Dec-20		87.0.4280.88
Java(TM) SE Development Kit 11.0.9 (64-bit)	10-Dec-20	279 MB	11.0.9.0
Microsoft OneDrive	27-Dec-20	148 MB	20.201.1005.0009
Microsoft Update Health Tools	20-Nov-20	1.05 MB	2.70.0.0
Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.40664	22-Dec-20	20.5 MB	12.0.40664.0
Microsoft Visual C++ 2015-2019 Redistributable (x64) - 14.28.29334	22-Dec-20	22.1 MB	14.28.29334.0
Microsoft Visual C++ 2015-2019 Redistributable (x86) - 14.28.2934	22-Dec-20	19.8 MB	14.28.29334.0
Microsoft Visual C++ Build Tools	20-Dec-20	3.12 GB	14.0.25420.1
Microsoft Visual Studio Installer	22-Dec-20		2.8.3074.1022
PostGIS Bundle 3.0.3 for PostgreSQL x64 10 (remove only)	13-Dec-20		
PostgreSQL 10	22-Dec-20	450 MB	10
PyCharm Community Edition 2020.3	10-Dec-20		203.5981.165
Python 3.7.8 (32-bit)	22-Dec-20	161 MB	3.7.8150.0
Python 3.9.1 (64-bit)	11-Dec-20	103 MB	3.9.1150.0
Python Launcher	10-Dec-20	1.79 MB	3.9.7280.0
Update for Windows 10 for x64-based Systems (KB4023057)	09-Aug-19	1.42 MB	2.61.0.0
Visual Studio Community 2019	21-Dec-20		16.8.30804.86
Windows SDK AddOn	22-Dec-20	152 KB	10.1.0.0
Windows Software Development Kit - Windows 10.0.18362.1	22-Dec-20	2.35 GB	10.1.18362.1

The next two images shows the required python bundles, that are shown in pycharm interpreter details.



## 2. Preparing input

Initially create a folder /dcimsgs in a path, this path is later provided as input to the tool. The output gif is generated in this path.

First step is to manually create the database. In this example, we have created an employee database.

The data to be inserted is available in the form of .sql file. In particular is named as data.sql.

It must be imported into the employee database using SQL Shell (psql) command: \i C:/pathToFile/data.sql (beware of the / direction) - as shown in the image below.

```

SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (10.15)
WARNING: Console code page (437) differs from Windows code page (1252)
          8-bit characters might not work correctly. See psql reference
          page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \connect employee
You are now connected to database "employee" as user "postgres".
employee=# \i C:/pathToFile/data.sql

```

## a) Nodes

After loading .sql file into the database, the data is available for observation in the form of tables and functions. The nodes table provides details on the data loaded. In this particular example it provides information of employees. However, for the tool to work any attributes can be provided. The only requirement is the presence of a uid attribute unique for each row. The requirement of this tool is existence of 2 columns geopoint and geom, if there are other columns with same name then they have to be renamed.

uid	name	place	email	phone	empid	dateofbirth	salary	geom	geopoint
1	17138 Garrett Q Coyle	Bvdgwilree,Cook	garrett.c...	8159797...	211241	4/18/1983	109891	010100020E61000000000000001013...	(49.8430808857083,-179.917200512253)
2	31723 Elvina J Math...	NULL	elvina.m...	316-719...	519109	1/13/1983	41196	010100020E61000000000BB512...	(50.156113055311,-179.970604237635)
3	42237 Tien Z Gies	NULL	tien.gies...	NULL	NULL	3/13/1982	76853	010100020E61000000000EABD4...	(49.2441937793046,-178.53165985253)
4	18339 Sybil A Armstr...	Muskegon,Mus...	sybil.arm...	231-270...	770466	3/20/1979	NULL	010100020E6100000000068A4A...	(48.9428601749241,-178.564577178098)
5	18588 Reena H Chu...	Bmio,fhrreaseg...	NULL	8516362...	NULL	12/09/95	120300	010100020E61000000000D9817...	(53.3160860356875,-178.25359209425)
6	38300 Judson D Har...	Hendon,Rawlin...	judson.h...	965532	NULL	NULL	010100020E610000000001300...	(54.1474056378938,-178.080023456831)	
7	31439 Priscilla R Bliss	Hastings,St. Jo...	priscilla...	239-574...	NULL	NULL	NULL	010100020E610000000008D336...	(54.9093079031445,-179.39938327924)
8	10127 August R Stoll	New Hampshire...	august.st...	216-862...	NULL	3/22/1974	NULL	010100020E61000000000F0156...	(54.8468959257007,-179.737071914598)
9	42817 Wallace P Ca...	Saint David,Fult...	wallace.c...	NULL	412731	NULL	113825	010100020E61000000000E164A...	(56.7942014788277,-179.684786238242)
10	3354 Elena Y Dixson	LtoersaWholun	NULL	7562301...	NULL	11-06-80	117984	010100020E610000000002C76D...	(56.709913989529,-179.02195535674)
11	22784 Austin B Carb...	NULL	austin.ca...	423-815...	462035	10/21/1989	86444	010100020E6100000000080BD6...	(61.3389813303947,-178.831316794269)
12	2064 Man L Starrett	NULL	man.star...	236-633...	883153	5/13/1993	156988	010100020E61000000000E987F...	(63.4686746490188,-178.804689239711)
13	3140 Stacy I Hallam	Keh,nrlisaldCrk...	stacy.hall...	5290773...	NULL	06-10-79	NULL	010100020E6100000000000F73F...	(63.6483291457407,-178.630378074478)
14	48288 Kathleen A F...	Sabine Pass,Jef...	kathleee...	NULL	256858	08-03-82	NULL	010100020E61000000000C8578...	(60.3008527494967,-178.355896202847)
15	7934 Jadwiga W U...	Stratford,Marat...	jadwiga...	262-754...	325655	2/20/1982	68103	010100020E61000000000CADD...	(58.8176533924416,-178.073860646226)
16	12329 Agustín T Da...	NoMlentnwg.wo	agustin.d...	3836514...	NULL	1/31/1986	NULL	010100020E6100000000EE97D...	(59.9205198204145,-178.99215118913)

## b) Edges

The edges table provides information on the relationships between rows in the nodes table. It mainly consists of 2 uid attributes (example: uid1, uid2) that identify the concerned rows of nodes table. More attributes can be added in this table. However, for the tool to work atleast 2 uids must be present here as well as in nodes table.

	uid1	uid2
1	1	63637
2	2	31159
3	3	8085
4	4	19313
5	5	65213
6	6	15082
7	7	12337
8	8	52256
9	9	26845
10	10	74435
11	11	53570
12	12	42087
13	13	25991
14	14	45794
15	15	14193
16	16	69533
17	17	46541
18	18	33586

### c) Data completion rules

The data completion rules table provides information on the left hand side and right hand side of the data completion rules as well as an identifier for it. The rule has a sequence of and's/or's with balanced parenthesis as well as attribute-value pairs.

**Query Editor**

```

1 SELECT * FROM data.inputdrules
2 ORDER BY idcrid ASC, inputlhs ASC, inputrhs ASC

```

**Data Output**

idcrid	inputlhs	inputrhs
IDCR01	(name='Chana P Wendell')and(place='BMCirwnt,aoanhoc')and(email='NULL')	(phone='NULL')
IDCR02	((name='Jewell T Desroches')or(email='jewell.desroches@apple.com'))and(phone='NULL')	(empid='726568')
IDCR03	((name='Velma Y Figueiredo')and(phone='490493379'))or(empid='NULL')	(dateofbirth='NULL')
IDCR04	((name='Garret E McGrew')or(empid='NULL'))and(email='NULL')	(phone='6307621882')
IDCR05	((name='Karie K Gamo')and(dateofbirth='NULL'))or(salary='NULL')	(place='LagiroatnDdijm')
IDCR06	((name='Dione K Whittley')or(salary='107837'))and(place='NULL')	(email='NULL')
IDCR07	((name='Maynard B Rush')and(salary='NULL'))or(email='maynard.rush@yahoo.com')	(place='Sree,contk')
IDCR08	((name='Norah C Dendy')or(dateofbirth='28-10-68'))and(place='NULL')	(salary='72482')
IDCR09	((name='Felipe Z Angulo')and(empid='NULL'))or(phone='NULL')	(email='felipe.angulo@ntlworld.com')
IDCR10	((name='Millicent L Millford')or(phone='NULL'))and(dateofbirth='20-10-86')	(empid='913954')
IDCR11	((name='Carl Z Carrion')and(email='NULL'))or(empid='NULL')	(phone='980368212')
IDCR12	((name='Dennis H Pafford')or(place='NULL'))and(phone='7532238850')	(email='dennis.pafford@hotmail.com')
IDCR13	((email='yuri.teter@yahoo.com')and(phone='231-553-8548'))or(name='Yuri J Teter')	(place='Flint,Genesee')
IDCR14	((email='julio.gossard@ntlworld.com')or(empid='227022'))and(dateofbirth='20-10-64')	(name='Julio W Gossard')
IDCR15	((email='NULL')and(dateofbirth='31-07-69'))or(name='Hilton T McCann')	(place='NULL')
IDCR16	(name='Young F Hou')or(email='young.hou@ntlworld.com')	(place='D,etwreocly')

## d) Record Matching Rules

The record matching rules table provides information on the identifier for the rule as well as the rule condition itself. The rule consists of balanced parenthesis as well as combination of and's/or's between attribute-value pairs of database.

**Query Editor**

```

1 SELECT * FROM data.inputmrules
2 ORDER BY inputrule ASC

```

**Data Output**

imrid	inputrule
IMMR04	((empid='469615')or(place='Porterville,Tulare')or(name='Lissa T Gervais'))and(email='lissa.gervais@gmail.com')
IMMR05	((name='Barbara T Gobert')or(place='ReLebunviews')or(phone='208-753-3960'))and(empid='211446')
IMMR07	((name='Bob Malick')or(phone='319-929-5849'))or(dateofbirth='18-04-75')and(salary='166017')
IMMR03	((name='Cassy J Polo')and(empid='281690'))and(dateofbirth='08-10-63')
IMMR14	((name='Florida E Encinas')and(empid='352309'))or((email='florida.encinas@gmail.com')and(dateofbirth='22-01-84'))
IMMR13	((name='Gerry A Karp')or(phone='6124297615'))or(dateofbirth='16-06-76')or((place='Berlin,Rensselaer')and(empid='225527'))
IMMR08	((name='Odell E Brock')and(dateofbirth='13-11-73'))or(email='odell.brock@microsoft.com')or(place='Fort Jones,Siskiyou')
IMMR12	((name='Rosalie N Hampson')or(empid='282055'))or(place='Canadian,Pittsburg')and(email='rosalie.hampson@yahoo.com')
IMMR15	((name='Vikki I Banh')or(dateofbirth='20-06-92'))and(phone='210-943-3339')or(salary='41233')
IMMR09	((name='Yoko P Hudock')or(salary='175362'))or(dateofbirth='17-05-86')or(place='Lowell,Middlesex')or(email='yoko.hudock@aol.com')or(empid='510728')and(phone='339-11-06-88')
IMMR10	(dateofbirth='11-06-88')and(salary='58163')and(empid='426413')
IMMR02	(name='Booker P Devita')and(empid='385057')
IMMR06	(name='Charlene W Meads')or(email='charlene.meads@yahoo.co.ir')or(empid='180225')
IMMR11	(name='Jennette I Wixom')and(email='jennette.wixom@cox.net')
IMMR01	(name='Kristin V Plain')and(place='Dayton,Dayton')

## e) Queries

The queries table consists of where conditions of queries log collected in a table. It has 2 columns an identifier for query and the actual query condition itself.

The screenshot shows the pgAdmin 4 interface with the following details:

- Servers:** PostgreSQL 10 (selected)
- Databases:** employee (selected)
- Tables:** queries (selected)
- Query Editor:**

```

1 SELECT * FROM data.queries
2 ORDER BY qid ASC

```
- Data Output:**

qid	query
Q01	(name='Micah O Herlihy') or (place='Abingdon,Washington')
Q02	(name='Lesley C Bessler') and (place='Stockton,Tooele')
Q03	((name='Brady H Simmett') or (empid='205441')) and (dateofbirth='08-10-72')
Q04	((empid='NULL') or (place='Adams Center,Jefferson') or (name='Apryl S Hargis')) and (email='NULL')
Q05	((name='Hildegard J Salcedo') or (place='Addison,Washington') or (phone='207-746-0970')) and (empid='172785')
Q06	(name='Riley R Gosser') and (dateofbirth='NULL') and (empid='897979')
Q07	((name='Elene E Crammer') or (phone='216-806-3451')) or ((dateofbirth='16-02-93')) and (salary='199468')
Q08	((name='Rene H Kautz') or (dateofbirth='23-12-62')) and (email='rene.kautz@hotmail.com') and (place='Alabama,Genesee')
Q09	((name='Deidre X Probst') or (salary='87916') or (dateofbirth='07-10-80') or (place='Sioux Falls,Minnehaha') or (empid='533441')) and (phone='605-883-3692')
Q10	((dateofbirth='28-06-74') or (salary='51255') and (empid='709815'))
Q11	(name='Raquel N Fisher') and (email='raquel.fisher@aol.com')
Q12	((name='Leticia R Puglisi') or (empid='946809') or (place='Apache Junction,Pinal')) and (email='lricia.puglisi@bellsouth.net')
Q13	((name='Britt G Morrone') or (phone='215-374-2595')) or ((dateofbirth='30-01-93') or ((place='Aristes,Columbia') and (empid='303835')))
Q14	((name='Shirlene S Sipos') or (empid='362169')) or ((email='shirlene.sipos@shell.com') and (dateofbirth='25-06-77'))
Q15	(name='Merrill J Hutchins') and (place='A,Detslanwaoe') and (phone='5618558926')
Q16	(place='Atlanta,Fulton') and (empid='828832')

Once the database is prepared, the input to the tool consists of a series of entries in the form of database name, username, password, port number and path of where the images and gifs must be stored.

The screenshots show the PyCharm Run tab with the following inputs:

```

Run: main
C:\Users\kuldeep\PycharmProjects\DataCompletionTool\venv\Scripts\python.exe C:/Users/kuldeep/PycharmProjects/DataCompl
-----
command line input ..
enter database name ..
employee
enter username ..

```

The second screenshot shows the continuation of the inputs:

```

Run: main
C:\Users\kuldeep\PycharmProjects\DataCompletionTool\venv\Scripts\python.exe C:/Users/kuldeep/PycharmProjects/DataCompl
-----
command line input ..
enter database name ..
employee
enter username ..
postgres
enter password ..
|_

```

```

Run: main x
C:\Users\kuldeep\PycharmProjects\DataCompletionTool\venv\Scripts\python.exe C:/Users/kuldeep/PycharmProjects/DataComp
-----
command line input ..
enter database name ..
employee
enter username ..
postgres
enter password ..
abc
enter port number ..
5432
path to directory where the images can be saved ..
C:\Users\kuldeep\OneDrive\Desktop\

started bootstrapping ..
-----
bootstrapping midway ..
done with bootstrapping
-----
```

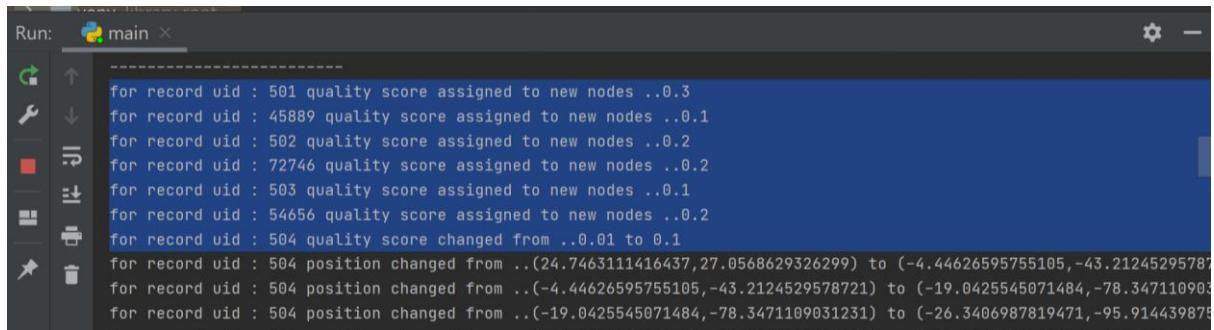
### 3. Output

The output consists of a sequence of steps of visualizations, altering data completion rules and efficiently executing them

#### a) Visualization

In the visualization part, spatial positions are assigned to data records corresponding to display positions. Their color intensity is continuously modified based on movement arising due to application of record matching rules.

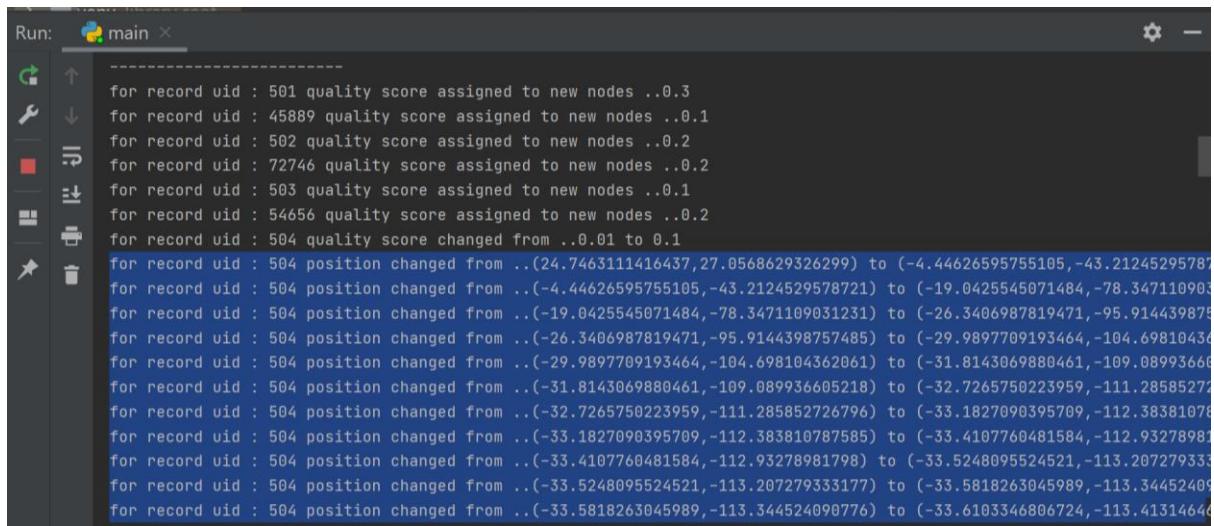
##### i. Quality Score



```
Run: main ×
-----
for record uid : 501 quality score assigned to new nodes ..0.3
for record uid : 45889 quality score assigned to new nodes ..0.1
for record uid : 502 quality score assigned to new nodes ..0.2
for record uid : 72746 quality score assigned to new nodes ..0.2
for record uid : 503 quality score assigned to new nodes ..0.1
for record uid : 54656 quality score assigned to new nodes ..0.2
for record uid : 504 quality score changed from ..0.01 to 0.1
for record uid : 504 position changed from ..(24.7463111416437,27.0568629326299) to (-4.44626595755105,-43.21245295787
for record uid : 504 position changed from ..(-4.44626595755105,-43.2124529578721) to (-19.0425545071484,-78.347110903
for record uid : 504 position changed from ..(-19.0425545071484,-78.3471109031231) to (-26.3406987819471,-95.914439875
```

## ii. Change in position of nodes

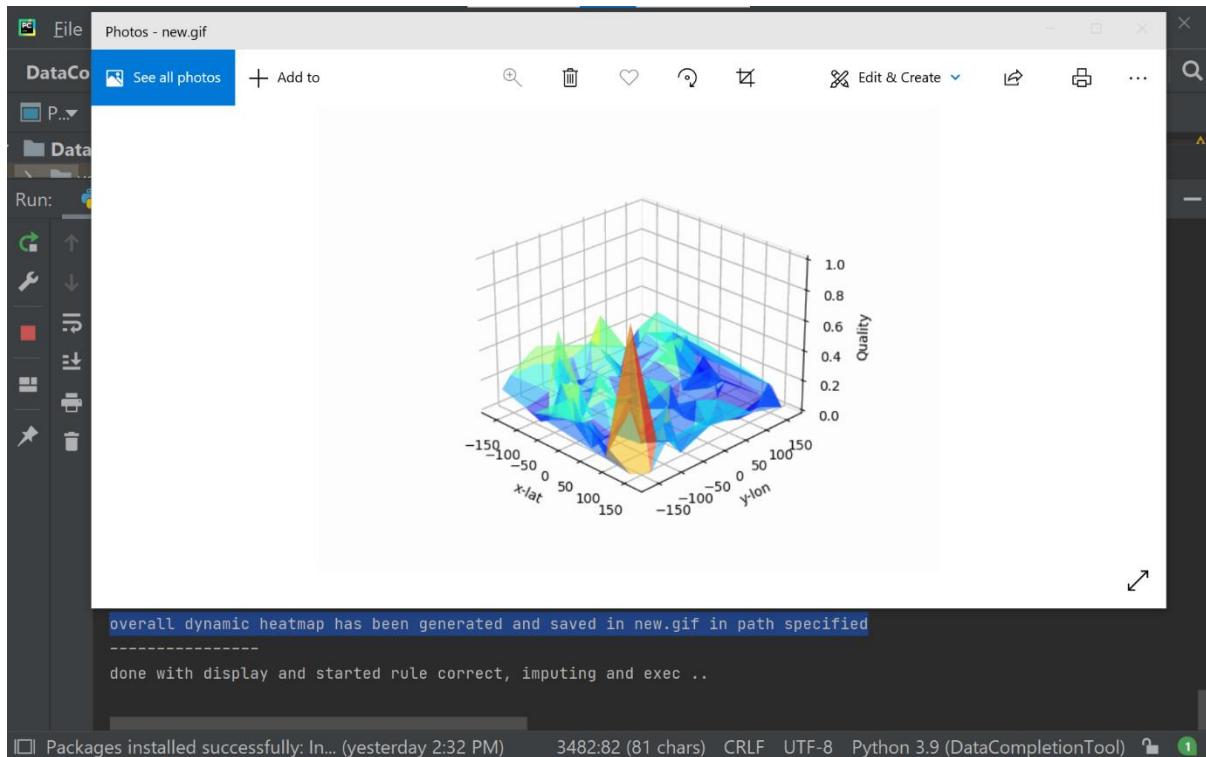
The spatial positions of nodes are also continuously changing due to the application of record matching rules and leveraging their properties



```
Run: main ×
-----
for record uid : 501 quality score assigned to new nodes ..0.3
for record uid : 45889 quality score assigned to new nodes ..0.1
for record uid : 502 quality score assigned to new nodes ..0.2
for record uid : 72746 quality score assigned to new nodes ..0.2
for record uid : 503 quality score assigned to new nodes ..0.1
for record uid : 54656 quality score assigned to new nodes ..0.2
for record uid : 504 quality score changed from ..0.01 to 0.1
for record uid : 504 position changed from ..(24.7463111416437,27.0568629326299) to (-4.44626595755105,-43.21245295787
for record uid : 504 position changed from ..(-4.44626595755105,-43.2124529578721) to (-19.0425545071484,-78.347110903
for record uid : 504 position changed from ..(-19.0425545071484,-78.3471109031231) to (-26.3406987819471,-95.914439875
for record uid : 504 position changed from ..(-26.3406987819471,-95.9144398757485) to (-29.9897709193464,-104.698104362061)
for record uid : 504 position changed from ..(-29.9897709193464,-104.698104362061) to (-31.8143069880461,-109.08993666
for record uid : 504 position changed from ..(-31.8143069880461,-109.089936605218) to (-32.7265750223959,-111.28585272
for record uid : 504 position changed from ..(-32.7265750223959,-111.285852726796) to (-33.1827090395709,-112.38381078
for record uid : 504 position changed from ..(-33.1827090395709,-112.383810787585) to (-33.4107760481584,-112.93278981
for record uid : 504 position changed from ..(-33.4107760481584,-112.93278981798) to (-33.5248095524521,-113.207279333
for record uid : 504 position changed from ..(-33.5248095524521,-113.207279333177) to (-33.5818263045989,-113.34452409
for record uid : 504 position changed from ..(-33.5818263045989,-113.344524090776) to (-33.6103346806724,-113.41314640
```

## iii. Generated heatmap

Apart from the textual generated output, the tool also produces output in the form of dynamic heatmap that reflects continuously changing quality scores and display spatial locations of records based on application of record matching rules with their properties.



## b) Transformation and execution

### i. Generated examples

Next, the tool generates examples for data transformations from the database using combination of data completion rules and record matching rules along with their properties

```
Run: main ×
done with display and started rule correct, imputing and exec ..
generated examples are ..[('10/29/1980', '3/18/1966'), ('181802', '123135'), ('Porterville,Tulare', 'San Jose,Santa C...
transform rule found is ..[[ "&0[('SPLIT', ',', ','), ('SUBSTRING', '3:12')] * 0[('CONCAT', ''), ('CONSTANT', '('), ('SUBSTR...
transform rule found is ..[[ "&0[('CONSTANT', ')'), ('CONCAT', ''), ('SUBSTRING', '0:5')] $ 0[('SPLIT', '('), ('CONCAT', ')...

```

### ii. Generated transform rule

Next, the tool produces the actual data transformation rules from the examples generated in previous step

```

Run: main x
done with display and started rule correct, imputing and exec ..
-----
generated examples are ..[('10/29/1980', '3/18/1966'), ('181802', '123135'), ('Porterville,Tulare', 'San Jose,Santa Cl...
-----
transform rule found is ..[[<&0[('SPLIT', ','), ('SUBSTRING', '3:12')]*0[('CONCAT', ''), ('CONSTANT', '('), ('SUBSTR...
-----
transform rule found is ..[[<&0[('CONSTANT', ')'), ('CONCAT', ''), ('SUBSTRING', '0:5')]]$0[('SPLIT', '('), ('CONCAT',...

```

### iii. Altered rules

Next, the tool produces a set of altered rules via the application of earlier data transformation rules generated

```

Run: main x
-----
altered rule using transformation ..(empid="94")and(dateofbirth="21-07-96")*(place="Cleveland,Stutsman")
-----
altered rule using transformation ..(name="is H")and(place="NULL")and(phone="7532238850")*(email="dennis.pafford@hotma...
-----
altered rule using transformation ..(name="h ")and(dateofbirth="28-10-68")and(place="NULL")*(salary="72482")

```

### iv. Imputed rules

Next, the tool produces a set of imputed data completion rules with NULL values in the data completion rule filled in from record matching rules

```

Run: main x
-----
imputed rule post transformation..(name="e K ")and(salary="107837")and(place="NU")*(email="dionekw@yahoo.com")
-----
imputed rule post transformation..(name="Maynard B Rush")and(salary="NU")and(email="maynard.rush@yahoo.com")*(place="S...
-----
imputed rule post transformation..(name="et")and(empid="22")and(email="NULL")*(phone="6307621882")

```

### v. Modified rules executable found via summary

Next, we identify a set of data completion rules that can be executed using a summary of incomplete database built using combination of data completion rules and record matching rules

```

Run: main x
1.modified&imputed rule can be executed after searching bloom filter via fds..(name="a P ")and(place="BMcirwnt,aoanh...
1.modified&imputed rule can be executed after searching bloom filter via fds..(name="a P ")and(place="BMcirwnt,aoanh...
2.modified&imputed rule can be executed after searching bloom filter via tfds ..(name="a P ")and(place="BMcirwnt,aoanh...
1.modified&imputed rule can be executed after searching bloom filter via fds..(name="h ")and(dateofbirth="28-10-68")an...
2.modified&imputed rule can be executed after searching bloom filter via tfds ..(name="h ")and(dateofbirth="28-10-68")

```

## vi. Records updated using modified rule

Next, we apply the modified data completion rules on the incomplete database and compute the number of records modified as a result

```
Run: main ×
22 number of records updated from the new rule (name="e K ")and(salary="37")and(place="NU")*(email="dionekw@yahoo.com")
-----
overall rank ..0.23871022358884816
-----
process finished ..
-----
>> Process finished with exit code 0
```

## vii. Overall rank

Next we compute the overall rank for the data completion process leveraging the queries table

```
Run: main ×
-----
overall rank ..0.23871022358884816
-----
process finished ..
-----
>> Process finished with exit code 0
```