Course: GY7708 Geospatial Databases and Information Retrieval

Student Number: 219031729

CW1

219031729

07/03/2022

Part 1

```
#Loading the data into R
database <-
  read.csv('C:/Users/khalsz/Documents/geodatabaseCW1.csv')

class(database)
```

```
## [1] "data.frame"
```

```
#
head(database)
```

```
##      statue                    statueLabel            inception      lat
## 1 Q18159833          Edith Cavell Memorial 1920-01-01T00:00:00Z 51.50932
## 2 Q19899485           Statue of Queen Anne 1886-01-01T00:00:00Z 51.51370
## 3  Q7270543       Statue of Queen Victoria 1888-01-01T00:00:00Z 51.45220
## 4 Q20822061        Statue of Amy Winehouse 2014-01-01T00:00:00Z 51.54240
## 5 Q25311639 Statue of Florence Nightingale 1915-01-01T00:00:00Z 51.50730
## 6 Q25311668         Statue of Mary Seacole 2009-01-01T00:00:00Z 51.50020
##         lon    place             placeLabel
## 1 -0.127183
## 2 -0.099900
## 3 -2.599410 Q5146396 College Green, Bristol
## 4 -0.147800  Q149836            Camden Town
## 5 -0.132600
## 6 -0.118900 Q1143779    St Thomas' Hospital
##                                                       placeAdmin  depicted
## 1                                                              Q18159833
## 2                                                              Q19899485
## 3 City of Bristol, South West England, England, United Kingdom  Q7270543
## 4    London Borough of Camden, Greater London, London, England Q20822061
## 5                                                              Q25311639
## 6                      London, Greater London, London, England Q25311668
##           depictedLabel
## 1          Edith Cavell
## 2 Anne of Great Britain
## 3        Queen Victoria
## 4         Amy Winehouse
## 5  Florence Nightingale
## 6          Mary Seacole
##
```

```
depictedAltLabel
## 1
Edith Louisa Cavell
## 2
Anne, Anne Stuart, Anne, Lord of Ireland, Anne, Queen of England, Anne, Queen
of Great Britain, Anne, Queen of Ireland, Anne, Queen of Scotland, koningin
van Groot-BrittanniÃ« en Ierland Anne, Queen Anne, Queen Anne of Great
Britain, Queen of England Anne, Queen of Great Britain Anne, Queen of Great
Britain Anne Stuart, Queen of Scotland Anne, regina d'Inghilterra Anna I,
reine de Grande-Bretagne Anne
## 3 Victoria, Princess Victoria, Victoria of the United Kingdom, Alexandrina
Victoria, Alexandrina Hanover, Alexandrina Victoria von Hannover, koningin
van Groot-BrittannieÍ\217Ì\210 en Ierland Victoria, Queen of Great Britain
and Empress of India Victoria, Queen of Great Britain Victoria, Queen
Victoria, Queen of the United Kingdom, regina di Gran Bretagna e Irlanda
Victoria, Reina de Gran BretaÃ±a Victoria I, reine de Grande-Bretagne
Victoria, Victoria Alexandrina, Victoria Hanover, Victoria, Queen of Great
Britain
## 4
Amy J, Amy Jade Winehouse
## 5
Lady with the Lamp, Angel of Crimea, Miss Smith, Nightingale Florence
## 6
Mary Jane Grant, Mary Jane Seacole, Mrs. Seacole
##
depictedDescription
## 1
British nurse (1865-1915)
## 2 queen of England, queen of Scotland and queen of Ireland (1702â\200“07);
queen of Great Britain (1707â\200“14)
## 3
British monarch who reigned 1837-1901
## 4
British singer and songwriter
## 5                         English social reformer, statistician, and
founder of modern nursing (1820-1910)
## 6                                                                 British-
Jamaican businesswoman and nurse
##     creator         creatorLabel  X
## 1  Q2265709      George Frampton NA
## 2 Q19899493  Richard Claude Belt NA
## 3    Q85180   Joseph Edgar Boehm NA
## 4 Q65030346          Scott Eaton NA
## 5  Q4798807 Arthur George Walker NA
## 6  Q6775804      Martin Jennings NA
```

```
colnames(database)
```

```
##  [1] "statue"              "statueLabel"         "inception"
##  [4] "lat"                 "lon"                 "place"
##  [7] "placeLabel"          "placeAdmin"          "depicted"
## [10] "depictedLabel"       "depictedAltLabel"    "depictedDescription"
## [13] "creator"             "creatorLabel"        "X"

database <- database %>% select(-X)
#printing out the columns in the database
colnames(database)

##  [1] "statue"              "statueLabel"         "inception"
##  [4] "lat"                 "lon"                 "place"
##  [7] "placeLabel"          "placeAdmin"          "depicted"
## [10] "depictedLabel"       "depictedAltLabel"    "depictedDescription"
## [13] "creator"             "creatorLabel"

#checking if there is any column with completely unique values
length(apply( X = database, FUN = anyDuplicated, MARGIN = 2)) ==
length(names(database))

## [1] TRUE

#Hence we need to give it a column with unique variables that will serve as
it primary key
database<- database %>% mutate(ID = row_number())


#Checking if the combinationn of ID and Statue can make a unique composite
primary key
nrow(unique(database[,c('statue', 'ID')])) == nrow(database)

## [1] TRUE

#Hence the two are worth using as composite primary key

#SPlitting the variable that are not of atomic values in the depictedAltLabel
column
#to make it conform with the first normal form


database$depictedAltLabel <-
str_split(as.character(database$depictedAltLabel), ",")

database <- database %>% unnest(depictedAltLabel)

database <- database %>% select(-depicted)
#Removed depicted column because it literately means the same as the statue
column,
#and this violates the 1NF rules.
database$statueLabel <- str_replace_all(database$statueLabel, ',', ' ')
database$placeLabel <- str_replace_all(database$placeLabel, ',', ' ')
database$placeAdmin <- str_replace_all(database$placeAdmin, ',', ' ')
database$depictedLabel <- str_replace_all(database$depictedLabel, ',', ' ')
```

```
database$depictedDescription <- str_replace_all(database$depictedDescription,
',', ' ')

#SPliting the table to conform with second normal form.
database1 <- database %>% select( statue, inception, ID)  %>%
  group_by(statue, ID) %>% filter(row_number() == 1)



database2 <- database %>% select(statue, place, placeLabel, lon, lat,
                                 placeAdmin, depictedLabel,depictedAltLabel,
                                 statueLabel, depictedDescription,
                                 creator, creatorLabel)%>% group_by(statue)
%>%
                                 filter(row_number() == 1)



statue_DateTable <- database1
statueLoc_table <- database2 %>% ungroup() %>% select(statue, lon, lat,
creator, place) %>%
  group_by(statue) %>% filter(row_number() == 1)

statueplaceTable <- database2 %>% ungroup() %>% select(place,
placeLabel,placeAdmin)%>%
  group_by(place) %>% filter(row_number() == 1)
Depictedtable <- database2 %>% ungroup() %>% select(statue,
depictedAltLabel,depictedLabel,
                                                    depictedDescription) %>%
  group_by(statue) %>% filter(row_number() == 1)

creatortable <- database2 %>% ungroup() %>% select(creator, creatorLabel) %>%
  group_by(creator) %>% filter(row_number() == 1)
```

building model for the tables

```
tablemodel <- dm(statue_DateTable, statueLoc_table,
statueplaceTable,Depictedtable,creatortable )

names(tablemodel)

## [1] "statue_DateTable" "statueLoc_table"  "statueplaceTable"
"Depictedtable"
## [5] "creatortable"
```

Checking for primary key and adding it to table

```
## # A tibble: 5 x 3
##   columns candidate why
##   <keys>  <lgl>     <chr>
## 1 statue  TRUE      ""
## 2 lon     TRUE      ""
## 3 lat     TRUE      ""
## 4 creator FALSE     "has duplicate values: Q2265709 (4), Q2591347 (3),
Q3215421~
## 5 place   FALSE     "has duplicate values:  (29)"

## # A tibble: 3 x 3
##   columns   candidate why
##   <keys>    <lgl>     <chr>
## 1 ID        TRUE      ""
## 2 statue    FALSE     "has duplicate values: Q47460184 (4), Q26492135 (2),
Q973~
## 3 inception FALSE     "has duplicate values: 1903-01-01T00:00:00Z (5),
1901-01-~

## # A tibble: 3 x 3
##   columns    candidate why
##   <keys>     <lgl>     <chr>
## 1 place      TRUE      ""
## 2 placeLabel TRUE      ""
## 3 placeAdmin FALSE     "has duplicate values: City of Bristol  South West
Engla~

## # A tibble: 4 x 3
##   columns             candidate why
##   <keys>              <lgl>     <chr>
## 1 statue              TRUE      ""
## 2 depictedAltLabel    FALSE     "has duplicate values: Victoria (23),
(4), Ann~
## 3 depictedLabel       FALSE     "has duplicate values: Queen Victoria
(23), Ann~
## 4 depictedDescription FALSE     "has duplicate values: British monarch who
reig~

## # A tibble: 2 x 3
##   columns      candidate why
##   <keys>       <lgl>     <chr>
## 1 creator      TRUE      ""
## 2 creatorLabel TRUE      ""

## -- Metadata --------------------------------------------------------------
------
## Tables: `statue_DateTable`, `statueLoc_table`, `statueplaceTable`,
`Depictedtable`, `creatortable`
## Columns: 17
## Primary keys: 5
## Foreign keys: 0

#Checking the link between tables  statue_DateTable with statueLoc_table
dm_enum_fk_candidates(
  dm = tablemodel_pks,
  table = statue_DateTable,
  ref_table = statueLoc_table
)
```

```
## # A tibble: 3 x 3
##   columns   candidate why
##   <keys>    <lgl>     <chr>
## 1 statue    TRUE      ""
## 2 inception FALSE     "values of `statue_DateTable$inception` not in
`statueLoc~
## 3 ID        FALSE     "Can't combine `value1` <integer> and `value1`
<character~
```

```
#Checking the link between tables  statueLoc_table with Depictedtable
dm_enum_fk_candidates(
  dm = tablemodel_pks,
  table = statueLoc_table,
  ref_table = Depictedtable
)
```

```
## # A tibble: 5 x 3
##   columns candidate why
##   <keys>  <lgl>     <chr>
## 1 statue  TRUE      ""
## 2 lon     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 3 lat     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 4 creator FALSE     "values of `statueLoc_table$creator` not in
`Depictedtable$~
## 5 place   FALSE     "values of `statueLoc_table$place` not in
`Depictedtable$st~
```

```
#Checking the link between tables _table with statueplaceTable
dm_enum_fk_candidates(
  dm = tablemodel_pks,
  table = statueLoc_table,
  ref_table = statueplaceTable
)
```

```
## # A tibble: 5 x 3
##   columns candidate why
##   <keys>  <lgl>     <chr>
## 1 place   TRUE      ""
## 2 statue  FALSE     "values of `statueLoc_table$statue` not in
`statueplaceTabl~
## 3 lon     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 4 lat     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 5 creator FALSE     "values of `statueLoc_table$creator` not in
`statueplaceTab~
```

```
#Checking the link between tables statueLoc_table with creatortable
dm_enum_fk_candidates(
  dm = tablemodel_pks,
  table = statueLoc_table,
  ref_table = creatortable
)
```

```
## # A tibble: 5 x 3
##   columns candidate why
##   <keys>  <lgl>     <chr>
## 1 creator TRUE      ""
## 2 statue  FALSE     "values of `statueLoc_table$statue` not in
`creatortable$cr~
## 3 lon     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 4 lat     FALSE     "Can't combine `value1` <double> and `value1`
<character>."
## 5 place   FALSE     "values of `statueLoc_table$place` not in
`creatortable$cre~
```

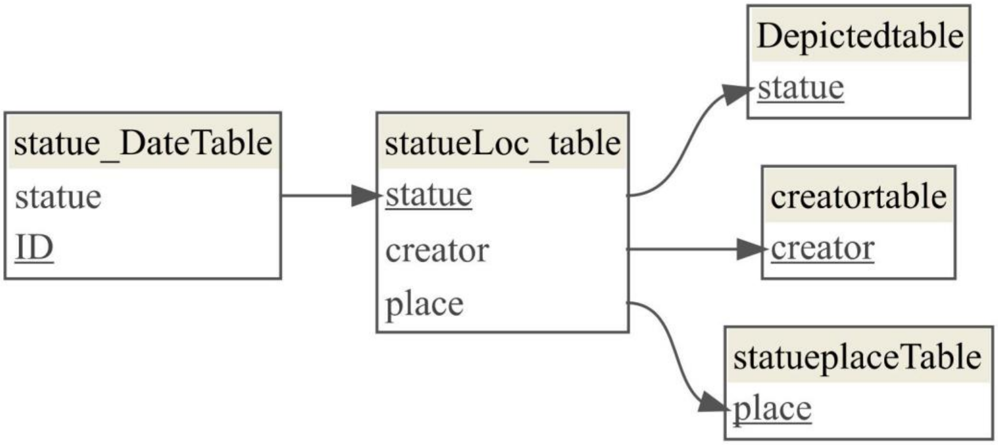
```
#Adding the foreign keys
complete_tablemodel <-
  tablemodel_pks %>%
  dm_add_fk(statue_DateTable, statue, statueLoc_table) %>%
  dm_add_fk(statueLoc_table, statue, Depictedtable) %>%
  dm_add_fk(statueLoc_table, place, statueplaceTable) %>%
  dm_add_fk(statueLoc_table, creator, creatortable)


complete_tablemodel
```

```
## -- Metadata --------------------------------------------------------------
------
## Tables: `statue_DateTable`, `statueLoc_table`, `statueplaceTable`,
`Depictedtable`, `creatortable`
## Columns: 17
## Primary keys: 5
## Foreign keys: 4
```

```
#Checking the integrity of the Database model.
complete_tablemodel %>%
  dm_examine_constraints()
```

```
## i All constraints satisfied.
```

```
#Visualizing the database relationship model
draw <- complete_tablemodel %>%
  dm_draw() %>%
  DiagrammeRsvg::export_svg() %>%
  htmltools::HTML() %>%
 htmltools::html_print()
draw
```



Creating Table for dataset GY7708_2021-22_Assignment_1–Statues-Wikidata.csv

- **Creating Table StatueLoc_Table**

DROP TABLE IF EXISTS StatueLoc_Table; CREATE TABLE StatueLoc_Table (statue CHARACTER VARYING NOT NULL, Lon Decimal(9,6), Lat Decimal(8,6), Creator CHARACTER VARYING, Place CHARACTER VARYING, PRIMARY KEY (statue), FOREIGN KEY (statue) REFERENCES DepictedTable(statue), FOREIGN KEY (place) REFERENCES StatuePlaceTable(place), FOREIGN KEY (creator) REFERENCES CreatorTable(creator)

- **Inserting values into table StatueLoc_Table**

INSERT INTO StatueLoc_Table VALUES (‘Q18159833’, ‘-0.127183’, ‘51.509324’, ‘Q2265709’, NULL);

INSERT INTO StatueLoc_Table VALUES (‘Q19899485’, ‘-0.0999’, ‘51.5137’, ‘Q19899493’, NULL);

INSERT INTO StatueLoc_Table VALUES (‘Q7270543’, ‘-2.59941’, ‘51.4522’, ‘Q85180’, ‘Q5146396’);

INSERT INTO StatueLoc_Table VALUES (‘Q20822061’, ‘-0.1478’, ‘51.5424’, ‘Q65030346’, ‘Q149836’);

```
INSERT INTO StatueLoc_Table VALUES ( ‘Q25311639’, ‘-0.1326’, ‘51.5073’, ‘Q4798807’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q25311668’, ‘-0.1189’, ‘51.5002’, ‘Q6775804’,
‘Q1143779’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26276238’, ‘-1.425885048’, ‘55.01777672’,
‘Q4723544’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26309866’, ‘0.367989874’, ‘51.44400088’,
‘Q8016365’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26639967’, ‘-0.1741’, ‘51.520272’, ‘Q1879407’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26658880’, ‘-0.088828’, ‘51.467863’,
‘Q16029268’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26406629’, ‘0.709666’, ‘51.534182’,
‘Q48110870’, ‘Q203995’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26669778’, ‘-0.766643’, ‘51.338879’,
‘Q87344165’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26677719’, ‘0.092567’, ‘51.769811’,
‘Q5423877’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q27080880’, ‘-0.18605’, ‘51.505386’, ‘Q161167’,
‘Q822607’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q27084123’, ‘-0.160803’, ‘51.618432’,
‘Q5625860’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q23034696’, ‘-0.1044’, ‘51.511’, ‘Q3215421’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q15615686’, ‘-2.89232’, ‘53.186’, ‘Q3087183’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q15979175’, ‘-1.5579’, ‘53.8111’, ‘Q2265709’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q15979535’, ‘-2.73632’, ‘53.4542’, ‘Q2265709’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q28548283’, ‘-0.107181’, ‘51.514311’, NULL,
‘Q26317815’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26482744’, ‘-0.169284’, ‘50.824629’,
‘Q2591347’, ‘Q989616’);
```

```
INSERT INTO StatueLoc_Table VALUES ( ‘Q26484518’, ‘-0.567055’, ‘51.425009’,
‘Q1759124’, ‘Q5474586’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26491275’, ‘-2.224786’, ‘53.012338’,
‘Q3215421’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26492135’, ‘-0.339174’, ‘53.743765’,
‘Q5719317’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17527162’, ‘-0.133067’, ‘51.5005’, ‘Q5480239’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26513391’, ‘-2.938544’, ‘54.898224’,
‘Q2591347’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26522424’, ‘-1.470821’, ‘52.917075’,
‘Q6470152’, ‘Q6670716’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26525766’, ‘-1.431351’, ‘54.986269’,
‘Q4711340’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26546446’, ‘-1.503436’, ‘53.367785’,
‘Q4723544’, ‘Q5375973’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q26562015’, ‘-2.45075’, ‘50.618765’, ‘Q5537017’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17540305’, ‘-1.5104’, ‘52.408’, ‘Q472840’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17552291’, ‘-1.61162’, ‘54.9703’, ‘Q2835166’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17552314’, ‘-1.61975’, ‘54.9785’, ‘Q2265709’,
‘Q7374975’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17775696’, ‘-4.92338’, ‘55.9456’, ‘Q15964470’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q17841622’, ‘-4.42299’, ‘55.845’, ‘Q14833533’,
‘Q17841607’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q29482042’, ‘-3.0015425’, ‘53.044206’,
‘Q87344165’, ‘Q7079013’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q47460184’, ‘-1.903073’, ‘52.479536’,
‘Q2591347’, ‘Q5765413’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q47885025’, ‘-1.1339’, ‘52.6345’, ‘Q7441068’,
‘Q6519199’);
```

```
INSERT INTO StatueLoc_Table VALUES ( ‘Q52157841’, ‘-0.127222222’, ‘51.50072222’,
‘Q10444417’, ‘Q1368556’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q56635583’, ‘-4.225503’, ‘57.475868’,
‘Q110779160’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q59296592’, ‘0.641666667’, ‘51.80083333’,
‘Q1428438’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q59773345’, ‘-2.24305556’, ‘53.47777778’,
‘Q59779567’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q97394804’, ‘-2.59681’, ‘51.45458’, ‘Q355811’,
‘Q7721919’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q96614428’, ‘-4.30843’, ‘55.862068’,
‘Q110779510’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q96623117’, ‘-2.954023’, ‘55.944499’,
‘Q2364675’, ‘Q1011336’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q96623123’, ‘-3.136358’, ‘55.933728’,
‘Q110779825’, NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q108862158’, ‘-2.486964’, ‘53.748859’,
‘Q7407615’, ‘Q188313’);

INSERT INTO StatueLoc_Table VALUES ( ‘Q110863651’, ‘-0.553’, ‘51.4416’, ‘Q6130606’,
NULL);

INSERT INTO StatueLoc_Table VALUES ( ‘Q107394352’, ‘-0.1866’, ‘51.50575’, ‘Q13637425’,
‘Q207385’);“)
```

- **Creating Table Statue_DateTable**

```
DROP TABLE IF EXISTS Statue_DateTable; CREATE TABLE Statue_DateTable ( statue
CHARACTER VARYING NOT NULL, inception Timestamp, ID SMALLINT NOT NULL,
PRIMARY KEY (ID), FOREIGN KEY (statue) REFERENCES StatueLoc_Table(statue), CHECK
(ID > 0)
```

- **Inserting values into table Statue_DateTable**

```
INSERT INTO Statue_DateTable VALUES ( ‘Q18159833’, ‘1920-01-01T00:00:00Z’, ‘1’);

INSERT INTO Statue_DateTable VALUES ( ‘Q19899485’, ‘1886-01-01T00:00:00Z’, ‘2’);

INSERT INTO Statue_DateTable VALUES ( ‘Q7270543’, ‘1888-01-01T00:00:00Z’, ‘3’);

INSERT INTO Statue_DateTable VALUES ( ‘Q20822061’, ‘2014-01-01T00:00:00Z’, ‘4’);

INSERT INTO Statue_DateTable VALUES ( ‘Q25311639’, ‘1915-01-01T00:00:00Z’, ‘5’);
INSERT INTO Statue_DateTable VALUES ( ‘Q25311668’, ‘2009-01-01T00:00:00Z’, ‘6’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26276238’, ‘1902-01-01T00:00:00Z’, ‘7’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26309866’, ‘1957-01-01T00:00:00Z’, ‘8’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26639967’, ‘1897-01-01T00:00:00Z’, ‘9’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26658880’, ‘1929-01-01T00:00:00Z’, ‘10’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26406629’, ‘1898-01-01T00:00:00Z’, ‘11’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26669778’, ‘1904-01-01T00:00:00Z’, ‘12’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26677719’, ‘1956-01-01T00:00:00Z’, ‘13’);
INSERT INTO Statue_DateTable VALUES ( ‘Q27080880’, ‘1887-01-01T00:00:00Z’, ‘14’);
INSERT INTO Statue_DateTable VALUES ( ‘Q27084123’, ‘1862-01-01T00:00:00Z’, ‘15’);
INSERT INTO Statue_DateTable VALUES ( ‘Q23034696’, ‘1896-01-01T00:00:00Z’, ‘16’);
INSERT INTO Statue_DateTable VALUES ( ‘Q15615686’, ‘1903-01-01T00:00:00Z’, ‘17’);
INSERT INTO Statue_DateTable VALUES ( ‘Q15979175’, ‘1903-01-01T00:00:00Z’, ‘18’);
INSERT INTO Statue_DateTable VALUES ( ‘Q15979535’, ‘1906-01-01T00:00:00Z’, ‘19’);
INSERT INTO Statue_DateTable VALUES ( ‘Q28548283’, ‘1905-01-01T00:00:00Z’, ‘20’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26482744’, ‘1901-01-01T00:00:00Z’, ‘21’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26484518’, ‘1887-01-01T00:00:00Z’, ‘22’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26491275’, ‘1903-01-01T00:00:00Z’, ‘23’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26492135’, ‘1903-01-01T00:00:00Z’, ‘24’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26492135’, ‘1925-01-01T00:00:00Z’, ‘25’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17527162’, ‘1705-01-01T00:00:00Z’, ‘26’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26513391’, ‘1902-01-01T00:00:00Z’, ‘27’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26522424’, ‘1914-01-01T00:00:00Z’, ‘28’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26525766’, ‘1913-01-01T00:00:00Z’, ‘29’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26546446’, ‘1904-01-01T00:00:00Z’, ‘30’);
INSERT INTO Statue_DateTable VALUES ( ‘Q26562015’, ‘1902-01-01T00:00:00Z’, ‘31’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17540305’, ‘1949-01-01T00:00:00Z’, ‘32’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17552291’, ‘1903-01-01T00:00:00Z’, ‘33’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17552314’, ‘1906-01-01T00:00:00Z’, ‘34’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17775696’, ‘1896-01-01T00:00:00Z’, ‘35’);
INSERT INTO Statue_DateTable VALUES ( ‘Q17841622’, ‘1901-01-01T00:00:00Z’, ‘36’);
INSERT INTO Statue_DateTable VALUES ( ‘Q29482042’, ‘1904-01-01T00:00:00Z’, ‘37’);
INSERT INTO Statue_DateTable VALUES ( ‘Q47460184’, ‘1901-01-01T00:00:00Z’, ‘38’);
INSERT INTO Statue_DateTable VALUES ( ‘Q47460184’, ‘1901-01-01T00:00:00Z’, ‘39’);
INSERT INTO Statue_DateTable VALUES ( ‘Q47460184’, ‘1951-01-01T00:00:00Z’, ‘40’);
INSERT INTO Statue_DateTable VALUES ( ‘Q47460184’, ‘1951-01-01T00:00:00Z’, ‘41’);
INSERT INTO Statue_DateTable VALUES ( ‘Q47885025’, ‘2018-01-01T00:00:00Z’, ‘42’);
INSERT INTO Statue_DateTable VALUES ( ‘Q52157841’, ‘2018-01-01T00:00:00Z’, ‘43’);
INSERT INTO Statue_DateTable VALUES ( ‘Q56635583’, ‘1896-01-01T00:00:00Z’, ‘44’);
INSERT INTO Statue_DateTable VALUES ( ‘Q59296592’, ‘1994-01-01T00:00:00Z’, ‘45’);
INSERT INTO Statue_DateTable VALUES ( ‘Q59773345’, ‘2018-01-01T00:00:00Z’, ‘46’);
INSERT INTO Statue_DateTable VALUES ( ‘Q97394804’, ‘2020-07-15T00:00:00Z’, ‘47’);
INSERT INTO Statue_DateTable VALUES ( ‘Q97394804’, ‘2020-07-15T00:00:00Z’, ‘48’);
INSERT INTO Statue_DateTable VALUES ( ‘Q96614428’, ‘2018-01-01T00:00:00Z’, ‘49’);
INSERT INTO Statue_DateTable VALUES ( ‘Q96623117’, ‘1995-01-01T00:00:00Z’, ‘50’);
INSERT INTO Statue_DateTable VALUES ( ‘Q96623123’, ‘2014-01-01T00:00:00Z’, ‘51’);
INSERT INTO Statue_DateTable VALUES ( ‘Q108862158’, ‘2019-01-01T00:00:00Z’, ‘52’);
INSERT INTO Statue_DateTable VALUES ( ‘Q110863651’, ‘2015-01-01T00:00:00Z’, ‘53’);
INSERT INTO Statue_DateTable VALUES ( ‘Q107394352’, ‘2021-01-01T00:00:00Z’, ‘54’);
```

- **Creating table DepictedTable**

```
DROP TABLE IF EXISTS DepictedTable; CREATE TABLE DepictedTable ( statue
CHARACTER VARYING NOT NULL, DepictedAltLabel CHARACTER VARYING, DepictedLabel
CHARACTER VARYING, DepictedDescription CHARACTER VARYING, PRIMARY KEY
(statue) );“)
```

- **Inserting values into table DepictedTable**

```
INSERT INTO DepictedTable VALUES (‘statue’, ‘depictedAltLabel’, ‘depictedLabel’,
‘depictedDescription’);
```

```
INSERT INTO DepictedTable VALUES (‘Q18159833’, ‘Edith Louisa Cavell’, ‘Edith Cavell’, ‘British nurse (1865-1915)’);

INSERT INTO DepictedTable VALUES (‘Q19899485’, ‘Anne’, ‘Anne of Great Britain’, ‘queen of England queen of Scotland and queen of Ireland (1702?07) queen of Great Britain (1707?14)’);

INSERT INTO DepictedTable VALUES (‘Q7270543’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q20822061’, ‘Amy J’, ‘Amy Winehouse’, ‘British singer and songwriter’);

INSERT INTO DepictedTable VALUES (‘Q25311639’, ‘Lady with the Lamp’, ‘Florence Nightingale’, ‘English social reformer statistician and founder of modern nursing (1820-1910)’);

INSERT INTO DepictedTable VALUES (‘Q25311668’, ‘Mary Jane Grant’, ‘Mary Seacole’, ‘British-Jamaican businesswoman and nurse’);

INSERT INTO DepictedTable VALUES (‘Q26276238’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26309866’, ‘Matoaka’, ‘Pocahontas’, ‘Algonquian princess’);

INSERT INTO DepictedTable VALUES (‘Q26639967’, ‘Mrs Siddons’, ‘Sarah Siddons’, ‘Welsh actress’);

INSERT INTO DepictedTable VALUES (‘Q26658880’, ‘Catherine Mumford’, ‘Catherine Booth’, ‘Mother of The Salvation Army’);

INSERT INTO DepictedTable VALUES (‘Q26406629’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26669778’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26677719’, ‘Dame Elisabeth Frink’, ‘Elisabeth Frink’, ‘English sculptor and printmaker (1930-1993)’);

INSERT INTO DepictedTable VALUES (‘Q27080880’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q27084123’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q23034696’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);
INSERT INTO DepictedTable VALUES (‘Q15615686’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q15979175’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q15979535’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q28548283’, ‘Mary’, ‘Mary Queen of Scots’, ‘Scottish monarch and queen consort of France (1542-1587)’);

INSERT INTO DepictedTable VALUES (‘Q26482744’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26484518’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26491275’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26492135’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q17527162’, ‘Anne’, ‘Anne of Great Britain’, ‘queen of England queen of Scotland and queen of Ireland (1702?07) queen of Great Britain (1707?14)’);

INSERT INTO DepictedTable VALUES (‘Q26513391’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26522424’, ‘Lady with the Lamp’, ‘Florence Nightingale’, ‘English social reformer statistician and founder of modern nursing (1820-1910)’);

INSERT INTO DepictedTable VALUES (‘Q26525766’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26546446’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q26562015’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q17540305’, ‘Godgifu’, ‘Lady Godiva’, ‘Anglo-Saxon noblewoman Countess of Leicester’);

INSERT INTO DepictedTable VALUES (‘Q17552291’, ‘Victoria’, ‘Queen Victoria’, ‘British monarch who reigned 1837-1901’);
INSERT INTO DepictedTable VALUES (‘Q17552314’, ‘Victoria’, ‘Queen Victoria’, ‘British
monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q17775696’, ‘Highland Mary’, ‘Mary Campbell’,
‘lover of Robert Burns’);

INSERT INTO DepictedTable VALUES (‘Q17841622’, ‘Victoria’, ‘Queen Victoria’, ‘British
monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q29482042’, ‘Victoria’, ‘Queen Victoria’, ‘British
monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q47460184’, ‘Victoria’, ‘Queen Victoria’, ‘British
monarch who reigned 1837-1901’);

INSERT INTO DepictedTable VALUES (‘Q47885025’, NULL, ‘Alice Hawkins’, ‘Leicester
suffragette’);

INSERT INTO DepictedTable VALUES (‘Q52157841’, ‘Dame Millicent Garrett Fawcett’,
‘Millicent Garrett Fawcett’, ‘English suffragist (1847-1929)’);

INSERT INTO DepictedTable VALUES (‘Q56635583’, ‘Flora Macdonald’, ‘Flora MacDonald’,
‘Scotland heroine; (1722-1790)’);

INSERT INTO DepictedTable VALUES (‘Q59296592’, ‘Dorothy Leigh Sayers’, ‘Dorothy L.
Sayers’, ‘English crime writer playwright essayist and Christian writer (1893-1957)’);

INSERT INTO DepictedTable VALUES (‘Q59773345’, ‘Emmeline Goulden’, ‘Emmeline
Pankhurst’, ‘English suffragist (1858-1928)’);

INSERT INTO DepictedTable VALUES (‘Q97394804’, NULL, ‘Jen Reid’, ‘protestor of Edward
Colston statue depicted by A Surge of Power (Jen Reid) 2020’);

INSERT INTO DepictedTable VALUES (‘Q96614428’, NULL, ‘Mary Barbour’, ‘British
politician (1875-1958)’);

INSERT INTO DepictedTable VALUES (‘Q96623117’, NULL, ‘Jackie Crookstone’, ‘rioter
massacre victim’);

INSERT INTO DepictedTable VALUES (‘Q96623123’, ‘Helen Murray Prentice’, ‘Helen
Crummy’, ‘Social Activist’);

INSERT INTO DepictedTable VALUES (‘Q108862158’, ‘Barbara Anne Betts’, ‘Barbara Castle
Baroness Castle of Blackburn’, ‘British politician (1910?2002)’);

INSERT INTO DepictedTable VALUES (‘Q110863651’, ‘Queen Elizabeth’, ‘Elizabeth II’,
‘queen of the UK Canada Australia and New Zealand and head of the Commonwealth of
Nations’);

INSERT INTO DepictedTable VALUES (‘Q107394352’, ‘Diana’, ‘Diana Princess of Wales’,
‘first wife of Charles Prince of Wales (1961-1997)’);“)
```

- **Creating table CreatorTable**

```
DROP TABLE IF EXISTS CreatorTable; CREATE TABLE CreatorTable ( creator CHARACTER VARYING NOT NULL, creatorlabel CHARACTER VARYING, PRIMARY KEY (creator)
```

- **Inserting Values into table CreatorTable**

```
INSERT INTO CreatorTable VALUES (‘Q2265709’, ‘George Frampton’);

INSERT INTO CreatorTable VALUES (‘Q19899493’, ‘Richard Claude Belt’);

INSERT INTO CreatorTable VALUES (‘Q85180’, ‘Joseph Edgar Boehm’);

INSERT INTO CreatorTable VALUES (‘Q65030346’, ‘Scott Eaton’);

INSERT INTO CreatorTable VALUES (‘Q4798807’, ‘Arthur George Walker’);

INSERT INTO CreatorTable VALUES (‘Q6775804’, ‘Martin Jennings’);

INSERT INTO CreatorTable VALUES (‘Q4723544’, ‘Alfred Turner’);

INSERT INTO CreatorTable VALUES (‘Q8016365’, ‘William Ordway Partridge’);

INSERT INTO CreatorTable VALUES (‘Q1879407’, ‘L?on-Joseph Chavalliaud’);

INSERT INTO CreatorTable VALUES (‘Q16029268’, ‘George Edward Wade’);

INSERT INTO CreatorTable VALUES (‘Q48110870’, ‘Joseph William Swynnerton’);

INSERT INTO CreatorTable VALUES (‘Q87344165’, ‘Henry Price’);

INSERT INTO CreatorTable VALUES (‘Q5423877’, ‘F. E. McWilliam’);

INSERT INTO CreatorTable VALUES (‘Q161167’, ‘Princess Louise’);

INSERT INTO CreatorTable VALUES (‘Q5625860’, ‘Joseph Durham’);

INSERT INTO CreatorTable VALUES (‘Q3215421’, ‘Charles Bell Birch’);

INSERT INTO CreatorTable VALUES (‘Q3087183’, ‘F. W. Pomeroy’);

INSERT INTO CreatorTable VALUES (NULL, NULL);

INSERT INTO CreatorTable VALUES (‘Q2591347’, ‘Thomas Brock’);

INSERT INTO CreatorTable VALUES (‘Q1759124’, ‘Prince Victor of Hohenlohe-Langenburg’);

INSERT INTO CreatorTable VALUES (‘Q5719317’, ‘Henry Charles Fehr’);

INSERT INTO CreatorTable VALUES (‘Q5480239’, ‘Francis Bird’);

INSERT INTO CreatorTable VALUES (‘Q6470152’, ‘Lady Feodora Gleichen’);

INSERT INTO CreatorTable VALUES (‘Q4711340’, ‘Albert Toft’);
INSERT INTO CreatorTable VALUES (‘Q5537017’, ‘George Blackall Simonds’);

INSERT INTO CreatorTable VALUES (‘Q472840’, ‘William Reid Dick’);

INSERT INTO CreatorTable VALUES (‘Q2835166’, ‘Alfred Gilbert’);

INSERT INTO CreatorTable VALUES (‘Q15964470’, ‘David Watson Stevenson’);

INSERT INTO CreatorTable VALUES (‘Q14833533’, ‘Francis John Williamson’);

INSERT INTO CreatorTable VALUES (‘Q7441068’, ‘Sean Hedges-Quinn’);

INSERT INTO CreatorTable VALUES (‘Q10444417’, ‘Gillian Wearing’);

INSERT INTO CreatorTable VALUES (‘Q110779160’, ‘Andrew Davidson’);

INSERT INTO CreatorTable VALUES (‘Q1428438’, ‘John Doubleday’);

INSERT INTO CreatorTable VALUES (‘Q59779567’, ‘Hazel Reeves’);

INSERT INTO CreatorTable VALUES (‘Q355811’, ‘Marc Quinn’);

INSERT INTO CreatorTable VALUES (‘Q110779510’, ‘Andrew Brown’);

INSERT INTO CreatorTable VALUES (‘Q2364675’, ‘David Annand’);

INSERT INTO CreatorTable VALUES (‘Q110779825’, ‘Tim Chalk’);

INSERT INTO CreatorTable VALUES (‘Q7407615’, ‘Sam Holland’);

INSERT INTO CreatorTable VALUES (‘Q6130606’, ‘James Butler’);

INSERT INTO CreatorTable VALUES (‘Q13637425’, ‘Ian Rank-Broadley’);“)
```

- **Creating Table StatuePlaceTable**

```
DROP TABLE IF EXISTS StatuePlaceTable; CREATE TABLE StatuePlaceTable ( place CHARACTER VARYING NOT NULL, placeLabel CHARACTER VARYING, placeAdmin CHARACTER VARYING, PRIMARY KEY (place)
```

- **Inserting values into table StatuePlaceTable**

```
INSERT INTO StatuePlaceTable VALUES (‘Q5146396’, ‘College Green Bristol’, ‘City of Bristol South West England England United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q149836’, ‘Camden Town’, ‘London Borough of Camden Greater London London England’);

INSERT INTO StatuePlaceTable VALUES (‘Q1143779’, ‘St Thomas’’ Hospital’, ‘London Greater London London England’);

INSERT INTO StatuePlaceTable VALUES (‘Q203995’, ‘Southend-on-Sea’, ‘Southend-on-Sea Essex East of England England’);
INSERT INTO StatuePlaceTable VALUES (‘Q822607’, ‘Kensington Gardens’, ‘City of
Westminster
Greater London London England’);

INSERT INTO StatuePlaceTable VALUES (‘Q26317815’, ‘Queen of Scots House’, ‘City of
London London England United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q989616’, ‘Hove’, ‘Brighton and Hove East Sussex
South East England England’);

INSERT INTO StatuePlaceTable VALUES (‘Q5474586’, ‘Founder’‘s Building’, ‘Runnymede
Surrey England United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q6670716’, ‘Derbyshire Royal Infirmary’, ‘Derby
City of Derby Derbyshire East Midlands’);

INSERT INTO StatuePlaceTable VALUES (‘Q5375973’, ‘Endcliffe Park’, ‘Sheffield South
Yorkshire Yorkshire and the Humber England’);

INSERT INTO StatuePlaceTable VALUES (‘Q7374975’, ‘Royal Victoria Infirmary’, ‘Newcastle
upon Tyne Newcastle upon Tyne Tyne and Wear North East England’);

INSERT INTO StatuePlaceTable VALUES (‘Q17841607’, ‘Dunn Square’, ‘Renfrewshire
Scotland United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q7079013’, ‘Offa’, ‘Wrexham County Borough
Wales United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q5765413’, ‘Victoria Square’, ‘Birmingham
Birmingham West Midlands West Midlands’);

INSERT INTO StatuePlaceTable VALUES (‘Q6519199’, ‘Leicester Market’, ‘Leicester City of
Leicester Leicestershire England’);

INSERT INTO StatuePlaceTable VALUES (‘Q1368556’, ‘Parliament Square’, ‘City of
Westminster Greater London London England’);

INSERT INTO StatuePlaceTable VALUES (‘Q7721919’, ‘The Centre Bristol’, ‘City of Bristol
South West England England United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q1011336’, ‘Tranent’, ‘East Lothian Scotland
United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q188313’, ‘Blackburn’, ‘Blackburn with Darwen
Lancashire England United Kingdom’);

INSERT INTO StatuePlaceTable VALUES (‘Q207385’, ‘Kensington Palace’, ‘Royal Borough of
Kensington and Chelsea Greater London London England’);“)
```

CW1

219031729

07/03/2022

Part 2

```
# Load the RPostgreSQL library
library(RPostgreSQL)
```

```
## Loading required package: DBI
```

```
pgsql_drv <- dbDriver("PostgreSQL")

# Connection information
pgsql_user <- "kal41"
pgsql_password <- "219031729"
pgsql_dbname <- "sds27"
pgsql_host <- "pgsql.mcs.le.ac.uk"
pgsql_port <- 5432

# Create the connection
pgsql_conn <- dbConnect(
  pgsql_drv,
  host = pgsql_host, port = pgsql_port,
  user = pgsql_user,
  password = pgsql_password,
  dbname = pgsql_dbname
)

# Remove the connection information
# from the R environment
rm(pgsql_user)
rm(pgsql_password)
rm(pgsql_dbname)
rm(pgsql_host)
rm(pgsql_port)
```

```
#Checking data types for each of the variables in table
#greater_london_osm_point
dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'greater_london_osm_point' ;"
)
```

```
##    column_name         data_type
## 1           id           integer
## 2         geom      USER-DEFINED
## 3       osm_id character varying
## 4         name character varying
## 5      barrier character varying
## 6      highway character varying
## 7          ref character varying
## 8      address character varying
## 9        is_in character varying
## 10       place character varying
## 11    man_made character varying
## 12  other_tags character varying
```

```
#Bicycle parking points in the study area: Havering
Cycle_P_in_Haerving <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT glop.id, glop.osm_id, glop.other_tags,
  gll.oa_code,  gll.lad11cd
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering';"
)

Cycle_P_in_Haerving
```

```
##        id     osm_id
## 1   14299  274870159
## 2  190880 3104472651
## 3  190894 3104472670
## 4  194291 3221558719
## 5  194420 3225682668
## 6  195742 3244383768
## 7  212032 3749182495
## 8  224232 4081920759
## 9  155016 2320751625
## 10  73876 1344561848
##
other_tags
## 1
"amenity"=>"bicycle_parking"
## 2
"amenity"=>"bicycle_parking"
## 3
"amenity"=>"bicycle_parking"
## 4                                               "amenity"=>"bicycle_parking","c
apacity"=>"6","bicycle_parking"=>"stands"
## 5                                                               "amenity"=>"bic
```

```
ycle_parking","bicycle_parking"=>"stands"
## 6  "access"=>"yes","amenity"=>"bicycle_parking","bicycle_parking"=>"stands","ca
pacity"=>"50","covered"=>"no","fee"=>"no"
## 7
"amenity"=>"bicycle_parking"
## 8
"amenity"=>"bicycle_parking"
## 9                           "amenity"=>"bicycle_parking","covered"=>"yes","capacit
y"=>"24","bicycle_parking"=>"high_density"
## 10                               "amenity"=>"bicycle_parking","covered"=>"yes","ca
pacity"=>"14","bicycle_parking"=>"stands"
##      oa_code   lad11cd
## 1  E00011849 E09000016
## 2  E00011452 E09000016
## 3  E00011481 E09000016
## 4  E00011914 E09000016
## 5  E00011914 E09000016
## 6  E00011935 E09000016
## 7  E00011659 E09000016
## 8  E00011935 E09000016
## 9  E00011935 E09000016
## 10 E00011935 E09000016
```

```
#Bicycle parking points in the study area: Havering
#Converted to human readable coordinate format
EWKT_Cycle_P_in_Haerving <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT  glop.other_tags, gll.lad11cd, gll.oa_code, gll.lad11nm,
ST_AsEWKT(glop.geom) geom_as_wkt
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
  st_transform(glop.geom, 27700),
  st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering';"
)

EWKT_Cycle_P_in_Haerving %>% select (lad11cd, oa_code, lad11nm,
                                     geom_as_wkt, other_tags)
```

```
##      lad11cd   oa_code  lad11nm                           geom_as_wkt
## 1  E09000016 E00011849 Havering SRID=4326;POINT(0.1925964 51.5257979)
## 2  E09000016 E00011452 Havering  SRID=4326;POINT(0.252502 51.5810805)
## 3  E09000016 E00011481 Havering SRID=4326;POINT(0.2356086 51.5944099)
## 4  E09000016 E00011914 Havering  SRID=4326;POINT(0.248445 51.5545826)
## 5  E09000016 E00011914 Havering  SRID=4326;POINT(0.2474866 51.553287)
## 6  E09000016 E00011935 Havering SRID=4326;POINT(0.2502582 51.5584349)
## 7  E09000016 E00011659 Havering SRID=4326;POINT(0.1912415 51.5898794)
## 8  E09000016 E00011935 Havering SRID=4326;POINT(0.2512451 51.5587753)
## 9  E09000016 E00011935 Havering SRID=4326;POINT(0.2520801 51.5585929)
## 10 E09000016 E00011935 Havering SRID=4326;POINT(0.2518227 51.5587505)
##                                                                                  other_tags
## 1                                                          "amenity"=>"bicycle_parking"
## 2                                                          "amenity"=>"bicycle_parking"
## 3                                                          "amenity"=>"bicycle_parking"
## 4                                               "amenity"=>"bicycle_parking","capacity"=>"6","bicycle_parking"=>"stands"
## 5                                                               "amenity"=>"bicycle_parking","bicycle_parking"=>"stands"
## 6  "access"=>"yes","amenity"=>"bicycle_parking","bicycle_parking"=>"stands","capacity"=>"50","covered"=>"no","fee"=>"no"
## 7                                                          "amenity"=>"bicycle_parking"
## 8                                                          "amenity"=>"bicycle_parking"
## 9                       "amenity"=>"bicycle_parking","covered"=>"yes","capacity"=>"24","bicycle_parking"=>"high_density"
## 10                            "amenity"=>"bicycle_parking","covered"=>"yes","capacity"=>"14","bicycle_parking"=>"stands"
```

```
#checking if there is any duplicate point
#Based on id, there are unique bicycle parking
IDGrouped_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT glop.id, glop.geom, count(*)
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
  st_transform(glop.geom, 27700),
  st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
GROUP BY glop.id;"
) %>% knitr::kable()
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warning:
## ((null))
```

```
IDGrouped_Haerving_Cycle_P
```

id	geom	count
14299	0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940	1
73876	0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940	1
155016	0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940	1
190880	0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940	1

190894	0101000020E61000002203D42F6C28CE3F9675A49F15CC4940	1
194291	0101000020E61000007767EDB60BCDCF3F20F70890FCC64940	1
194420	0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940	1
195742	0101000020E6100000AC9223F83A04D03FF73878CB7AC74940	1
212032	0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940	1
224232	0101000020E61000009CEFF1536614D03F03A8F3F285C74940	1

```
#checking if there is any duplicate point
#Based on OA and Supgroup_CD, there are unique bicycle parking
OA_Grouped_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT gll.oa_code, gll.supgrp_cd, count(*)
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
GROUP BY gll.oa_code, gll.supgrp_cd;"
) %>% knitr::kable()

OA_Grouped_Haerving_Cycle_P
```

oa_code	supgrp_cd	count
E00011452	H	1
E00011481	F	1
E00011659	H	1
E00011849	A	1
E00011914	F	2
E00011935	H	4

```
#checking if there is any duplicate point
#Based on OA and grp_cd, there are unique bicycle parking

CD_Grouped_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT gll.grp_cd, count(*)
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
GROUP BY gll.grp_cd;"
) %>% knitr::kable()

CD_Grouped_Haerving_Cycle_P
```

grp_cd	count
A2	1
F2	3
H1	2
H2	4

```
#Of the over 750 multipolygons in the area, only 37 intersects with 100 meter buffer
#around the bicyle parking.
hndBff_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "SELECT gll.id, gll.oa_code
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_buffer(
st_transform(glop.geom, 27700),
100),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering';"
)
```

```
#Roads that intersect bicycle parking points in  Havering
Road_int_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "WITH hr
as(
SELECT glop.geom, glop.osm_id
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_Within(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
)
SELECT hr.*,  glol.name
FROM hr
INNER JOIN greater_london_osm_line glol
ON st_intersects(st_transform(glol.geom, 27700),
st_transform(hr.geom, 27700))"
)
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))
```

```
Road_19m_buff_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "WITH  hr
as(
SELECT glop.geom, glop.osm_id, glop.name
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
)
SELECT hr.*,  glol.name, glol.highway
FROM hr
INNER JOIN greater_london_osm_line glol
ON st_intersects(st_buffer(st_transform(hr.geom, 27700),19
), st_transform(glol.geom, 27700))"
)
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))
```

```
Road_19m_buff_Haerving_Cycle_P
```

```
##                                                    geom     osm_id
## 1   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 2   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 3   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 4   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 5   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 6   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 7   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 8   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 9   0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 10  0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 11  0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 12  0101000020E610000018E6A9B3FFA6C83F1A6778584DC34940  274870159
## 13  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 14  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 15  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 16  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 17  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 18  0101000020E6100000842C0B26FE28D03FF3EB87D860CA4940 3104472651
## 19  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 20  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 21  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 22  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 23  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 24  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 25  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 26  0101000020E61000002203D42F6C28CE3F9675A49F15CC4940 3104472670
## 27  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 28  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 29  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 30  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 31  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 32  0101000020E61000007767EDB60BCDCF3F20F70890FCC64940 3221558719
## 33  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 34  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 35  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 36  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 37  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 38  0101000020E6100000B85F9912A4ADCF3FA726C11BD2C64940 3225682668
## 39  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 40  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 41  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 42  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 43  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 44  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 45  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 46  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 47  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 48  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 49  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 50  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 51  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 52  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 53  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 54  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 55  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 56  0101000020E6100000AC9223F83A04D03FF73878CB7AC74940 3244383768
## 57  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 58  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 59  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 60  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 61  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 62  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 63  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 64  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 65  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 66  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 67  0101000020E610000076A911FA997AC83FC4CA0D2B81CB4940 3749182495
## 68  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 69  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 70  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 71  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 72  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 73  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 74  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 75  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 76  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 77  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 78  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 79  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 80  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 81  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 82  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 83  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 84  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 85  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 86  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 87  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 88  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 89  0101000020E61000009CEFF1536614D03F03A8F3F285C74940 4081920759
## 90  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 91  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 92  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 93  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 94  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 95  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 96  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 97  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 98  0101000020E6100000F53B5E921422D03F8FA3DEF87FC74940 2320751625
## 99  0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 100 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 101 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 102 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 103 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 104 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 105 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 106 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 107 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 108 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 109 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 110 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 111 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
## 112 0101000020E6100000A1FC38F5DC1DD03FD40FEA2285C74940 1344561848
##                                 name
## 1                               <NA>
## 2                               <NA>
## 3                               <NA>
## 4                               <NA>
## 5                               <NA>
## 6                               <NA>
## 7                               <NA>
## 8                               <NA>
## 9                               <NA>
## 10                              <NA>
## 11                              <NA>
## 12                              <NA>
## 13                              <NA>
## 14                              <NA>
## 15                              <NA>
## 16                              <NA>
## 17                              <NA>
## 18                              <NA>
## 19                              <NA>
## 20                              <NA>
## 21                              <NA>
## 22                              <NA>
## 23                              <NA>
## 24                              <NA>
## 25                              <NA>
## 26                              <NA>
## 27                              <NA>
## 28                              <NA>
## 29                              <NA>
## 30                              <NA>
## 31                              <NA>
## 32                              <NA>
## 33                              <NA>
## 34                              <NA>
## 35                              <NA>
## 36                              <NA>
## 37                              <NA>
## 38                              <NA>
## 39                              <NA>
## 40                              <NA>
## 41                              <NA>
## 42                              <NA>
## 43                              <NA>
## 44                              <NA>
## 45                              <NA>
## 46                              <NA>
## 47                              <NA>
## 48                              <NA>
## 49                              <NA>
## 50                              <NA>
## 51                              <NA>
## 52                              <NA>
## 53                              <NA>
## 54                              <NA>
## 55                              <NA>
## 56                              <NA>
## 57                              <NA>
## 58                              <NA>
## 59                              <NA>
## 60                              <NA>
## 61                              <NA>
## 62                              <NA>
## 63                              <NA>
## 64                              <NA>
## 65                              <NA>
## 66                              <NA>
## 67                              <NA>
## 68  Upminster Station Bike Compound
## 69  Upminster Station Bike Compound
## 70  Upminster Station Bike Compound
## 71  Upminster Station Bike Compound
## 72  Upminster Station Bike Compound
## 73  Upminster Station Bike Compound
## 74  Upminster Station Bike Compound
## 75  Upminster Station Bike Compound
## 76  Upminster Station Bike Compound
## 77  Upminster Station Bike Compound
## 78  Upminster Station Bike Compound
## 79  Upminster Station Bike Compound
## 80  Upminster Station Bike Compound
## 81  Upminster Station Bike Compound
## 82  Upminster Station Bike Compound
## 83  Upminster Station Bike Compound
## 84  Upminster Station Bike Compound
## 85  Upminster Station Bike Compound
## 86  Upminster Station Bike Compound
## 87  Upminster Station Bike Compound
## 88  Upminster Station Bike Compound
## 89  Upminster Station Bike Compound
## 90                             <NA>
## 91                             <NA>
## 92                             <NA>
## 93                             <NA>
## 94                             <NA>
## 95                             <NA>
## 96                             <NA>
## 97                             <NA>
## 98                             <NA>
## 99                             <NA>
## 100                            <NA>
## 101                            <NA>
## 102                            <NA>
## 103                            <NA>
## 104                            <NA>
## 105                            <NA>
## 106                            <NA>
## 107                            <NA>
## 108                            <NA>
## 109                            <NA>
## 110                            <NA>
## 111                            <NA>
## 112                            <NA>
##                                                    name      highway
## 1                                      Knightswood Road  residential
## 2                                                  <NA>      service
## 3                                       Ingrebourne Way     cycleway
## 4                                          Rainham Road      primary
## 5                                                  <NA>      service
## 6                                          Rainham Road      primary
## 7                                                  <NA>      footway
## 8                                                  <NA>     cycleway
## 9                  London Buses route 103 → Chase Cross         <NA>
## 10                               Ingreborne Valley Way         <NA>
## 11                            London LOOP (Section 23)         <NA>
## 12        London Buses route 103 → Rainham Interchange         <NA>
## 13                                                <NA>      service
## 14                                                <NA>     cycleway
## 15                                                <NA>      footway
## 16                                                <NA>      footway
## 17                               Ingreborne Valley Way         <NA>
## 18                            London LOOP (Section 22)         <NA>
## 19                                        Arundel Road  residential
## 20                                         Avenue Road     tertiary
## 21                                    Queens Park Road  residential
## 22                                         Avenue Road     tertiary
## 23                                        Station Road     tertiary
## 24                                                <NA>      footway
## 25  London Buses route 496 → Romford, Queen's Hospital         <NA>
## 26                London Buses route 496 → Harold Wood         <NA>
## 27                                                <NA>         <NA>
## 28                                    Corbets Tey Road    secondary
## 29                   London Buses route 370 → Lakeside         <NA>
## 30             London Buses route 370 → Romford Market         <NA>
## 31             London Buses route 370 → Romford Market         <NA>
## 32                London Buses route 370 → Corbets Tey         <NA>
## 33                                      Stewart Avenue  residential
## 34                                    Corbets Tey Road    secondary
## 35                   London Buses route 370 → Lakeside         <NA>
## 36             London Buses route 370 → Romford Market         <NA>
## 37             London Buses route 370 → Romford Market         <NA>
## 38                London Buses route 370 → Corbets Tey         <NA>
## 39                                        Station Road     tertiary
## 40                                        Station Road     tertiary
## 41                                                <NA>      footway
## 42             London Buses route 248 → Romford Market         <NA>
## 43                   London Buses route 370 → Lakeside         <NA>
## 44                               Ingreborne Valley Way         <NA>
## 45                  London Buses route 346 → Upminster         <NA>
## 46                  London Buses route 652 → Upminster         <NA>
## 47                    London Buses route 646 → Cranham         <NA>
## 48                  London Buses route 646 → Noak Hill         <NA>
## 49                    London Buses route 248 → Cranham         <NA>
## 50                    London Buses route 648 → Cranham         <NA>
## 51             London Buses route 648 → Romford Market         <NA>
## 52                   London Buses route 347 → Ockendon         <NA>
## 53            London Buses route 347 → Romford Station         <NA>
## 54             London Buses route 370 → Romford Market         <NA>
## 55             London Buses route 370 → Romford Market         <NA>
## 56                London Buses route 370 → Corbets Tey         <NA>
## 57                                                <NA>         <NA>
## 58                                                <NA>      footway
## 59                                                <NA>        steps
## 60                                                <NA>         <NA>
## 61                                                <NA>      footway
## 62                                                <NA>      footway
## 63                                                <NA>      footway
## 64                                                <NA>      footway
## 65                                                 <NA>        steps
## 66                                                 <NA>     cycleway
## 67                                                 <NA>         <NA>
## 68                                     Station Approach unclassified
## 69                                                 <NA>         <NA>
## 70                                                 <NA>         <NA>
## 71                      London, Tilbury & Southend Line         <NA>
## 72                    London Buses route 370 → Lakeside         <NA>
## 73                   London Buses route 346 → Upminster         <NA>
## 74                     London Buses route 646 → Cranham         <NA>
## 75                   London Buses route 646 → Noak Hill         <NA>
## 76                     London Buses route 648 → Cranham         <NA>
## 77              London Buses route 648 → Romford Market         <NA>
## 78              London Buses route 370 → Romford Market         <NA>
## 79            C2C: London - Chafford Hundred - Southend         <NA>
## 80               C2C: Shoeburyness - London (semi-fast)         <NA>
## 81               C2C: London - Shoeburyness (semi-fast)         <NA>
## 82                C2C: Shoeburyness - London (stopping)         <NA>
## 83                C2C: London - Shoeburyness (stopping)         <NA>
## 84                           C2C: Leigh-on-Sea - London         <NA>
## 85                           C2C: London - Leigh-on-Sea         <NA>
## 86            C2C: Southend - Chafford Hundred - London         <NA>
## 87              London Buses route 370 → Romford Market         <NA>
## 88                 London Buses route 370 → Corbets Tey         <NA>
## 89                                                 <NA>         <NA>
## 90                           London, Tilbury & Southend         <NA>
## 91                                                 <NA>      service
## 92                                                 <NA>         <NA>
## 93                           London, TIlbury & Southend         <NA>
## 94                      London, Tilbury & Southend Line         <NA>
## 95               C2C: Shoeburyness - London (semi-fast)         <NA>
## 96                C2C: Shoeburyness - London (stopping)         <NA>
## 97                           C2C: Leigh-on-Sea - London         <NA>
## 98            C2C: Southend - Chafford Hundred - London         <NA>
## 99                                                 <NA>         <NA>
## 100                          London, TIlbury & Southend         <NA>
## 101                                                <NA>         <NA>
## 102                                                <NA>      footway
## 103                     London, Tilbury & Southend Line         <NA>
## 104           C2C: London - Chafford Hundred - Southend         <NA>
## 105              C2C: Shoeburyness - London (semi-fast)         <NA>
## 106              C2C: London - Shoeburyness (semi-fast)         <NA>
## 107               C2C: Shoeburyness - London (stopping)         <NA>
## 108               C2C: London - Shoeburyness (stopping)         <NA>
## 109                          C2C: Leigh-on-Sea - London         <NA>
## 110                          C2C: London - Leigh-on-Sea         <NA>
## 111           C2C: Southend - Chafford Hundred - London         <NA>
## 112                                                <NA>         <NA>
```

```
#Buildings that intersect bicycle parking in Havering
Building_int_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "WITH  hm
as(
SELECT glop.geom, glop.osm_id, glop.name
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
)
SELECT hm.geom, hm.osm_id, hm.name,  glop2.building, glop2.tourism, glop2.sport, g
lop2.office, glop2.geom
FROM hm
INNER JOIN greater_london_osm_polygon glop2
ON st_intersects(st_transform( hm.geom, 27700)
, st_transform(glop2.geom, 27700))
WHERE glop2.building IS NOT NULL"
)
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))

## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))
```

```
Building_int_Haerving_Cycle_P
```

```
## [1] geom     osm_id   name     building tourism  sport    office   geom    
## <0 rows> (or 0-length row.names)
```

```
#no building intersect bicycle parkings
```

```
#Buildings within 50m distance from the bicycle parkings
Building_50m_away_Haerving_Cycle_P <- dbGetQuery(
  conn = pgsql_conn,
  statement = "WITH  hm
as(
SELECT glop.geom, glop.osm_id, glop.name, glop.other_tags
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
)
SELECT hm.geom, hm.osm_id, hm.name,  glop2.building, glop2.tourism, glop2.sport, g
lop2.office, glop2.geom, hm.other_tags
FROM hm
INNER JOIN greater_london_osm_polygon glop2
ON st_dwithin(st_transform( hm.geom, 27700)
, st_transform(glop2.geom, 27700), 50)
WHERE glop2.building IS NOT NULL"
)
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))

## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warnin
g:
## ((null))
```

```
class(Building_50m_away_Haerving_Cycle_P)
```

```
## [1] "data.frame"
```

```
All_road <- st_read(
  pgsql_conn,
  query = "SELECT * FROM greater_london_osm_line;"
)
```

```
## Warning: RS-DBI driver warning: ((null))
```

```
Hovering <- st_read(
  pgsql_conn,
  query = "  SELECT * FROM greater_london_loac gll
WHERE lad11nm = 'Havering';"
)
```

```
## Warning: RS-DBI driver warning: ((null))
```

```
all_building <- st_read(
  pgsql_conn,
  query = "SELECT * FROM greater_london_osm_polygon as glop2
  WHERE glop2.building IS NOT NULL;"
)
```

```
## Warning: RS-DBI driver warning: ((null))
```

```
cycle <- st_read(
  pgsql_conn,
  query = "SELECT *
      FROM greater_london_osm_point as glop
      WHERE glop.other_tags like '%bicycle_parking%'"
)
```

```
## Warning: RS-DBI driver warning: ((null))
```

```
Cycle_Pk_in_Haerving <- st_read(
  pgsql_conn,
  query = "
    SELECT glop.geom, gll.supgrp_cd, gll.grp_cd
      FROM greater_london_osm_point as glop
INNER JOIN greater_london_loac as gll
        ON ST_within(
                 ST_Transform(glop.geom, 27700),
                 ST_Transform(gll.geom, 27700)
             )
             WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering';"
)
```

```
## Warning: RS-DBI driver warning: ((null))
```

```
buildingwithin50m <- st_read(
  pgsql_conn,
  query = "WITH  hm
as(
  SELECT glop.geom, glop.osm_id, glop.name, glop.other_tags
  FROM greater_london_osm_point glop
  INNER JOIN greater_london_loac gll
  ON st_intersects(
    st_transform(glop.geom, 27700),
    st_transform(gll.geom, 27700)
  )
  WHERE glop.other_tags like '%bicycle_parking%' AND
  gll.lad11nm = 'Havering'
)
SELECT hm.geom, hm.osm_id, hm.name,  glop2.building, glop2.tourism, glop2.sport, glop2.office, glop2.geom, hm.other_tags
FROM hm
INNER JOIN greater_london_osm_polygon glop2
ON st_dwithin(st_transform( hm.geom, 27700)
              , st_transform(glop2.geom, 27700), 50)
WHERE glop2.building IS NOT NULL;"
)
```

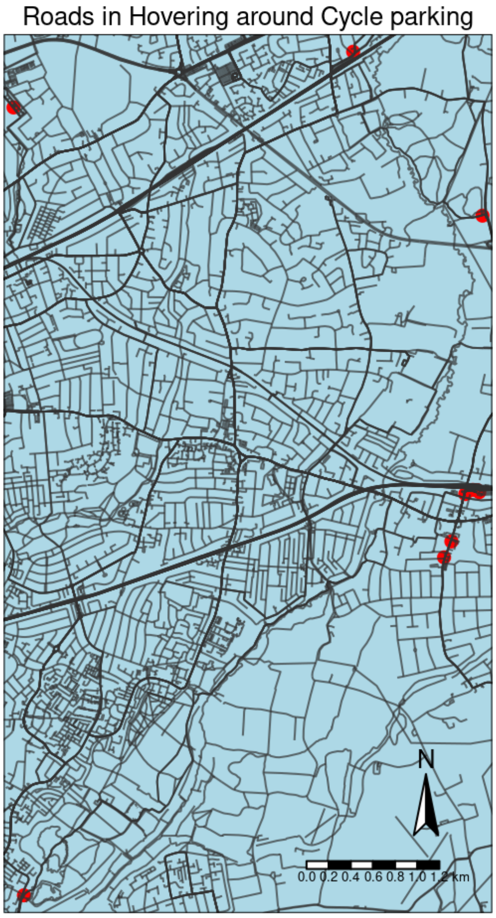
```
## Warning: RS-DBI driver warning: ((null))

## Warning: RS-DBI driver warning: ((null))
```

```
roads19maway <- st_read(
  pgsql_conn,
  query = "
  WITH  hr
as(
SELECT glop.geom, glop.osm_id, glop.name
FROM greater_london_osm_point glop
INNER JOIN greater_london_loac gll
ON st_intersects(
st_transform(glop.geom, 27700),
st_transform(gll.geom, 27700)
)
WHERE glop.other_tags like '%bicycle_parking%' AND
gll.lad11nm = 'Havering'
)
SELECT hr.*, glol.geom , glol.name, glol.highway
FROM hr
INNER JOIN greater_london_osm_line glol
ON st_intersects(st_buffer(st_transform(hr.geom, 27700),19
), st_transform(glol.geom, 27700))
"
)
```

```
## Warning: RS-DBI driver warning: ((null))

## Warning: RS-DBI driver warning: ((null))
```

```
pall <- c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3", "#FF7F00", "#FFFF33", "#A656
28", "#F781BF",'#01BEFE')
```

```
#Bicycle Parkings in Havering
tm_shape(Cycle_Pk_in_Haerving) +
  # Represent them as filled polygons
  tm_dots(col = 'red', size = 0.18) +
  # Add the line shapes
  tm_shape(All_road) +
  # Represent them as lines
  tm_lines(col = "#333333", alpha = 0.7) +
  tm_layout(
    main.title = 'Roads in Hovering around Cycle parking', bg.color="lightblue",
    main.title.size = 0.8, main.title.position="center") +
  tm_compass() + tm_scale_bar() +
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```

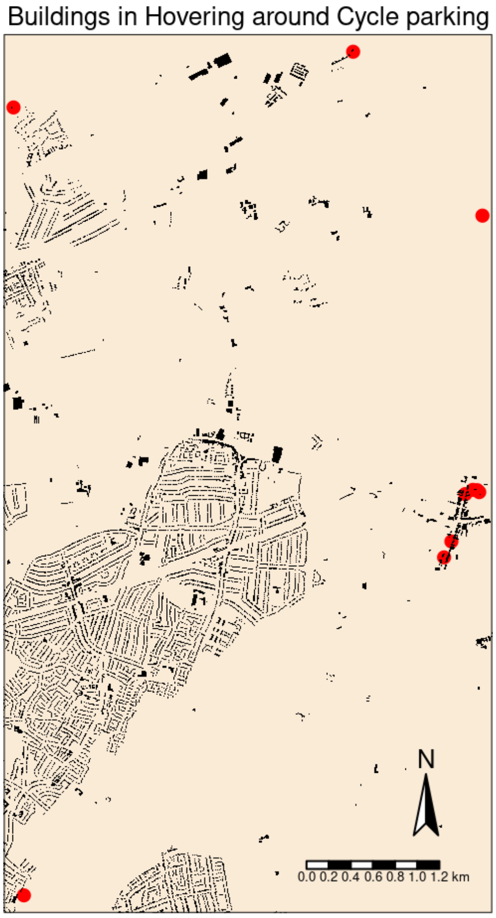


Roads in Hovering around Cycle parking

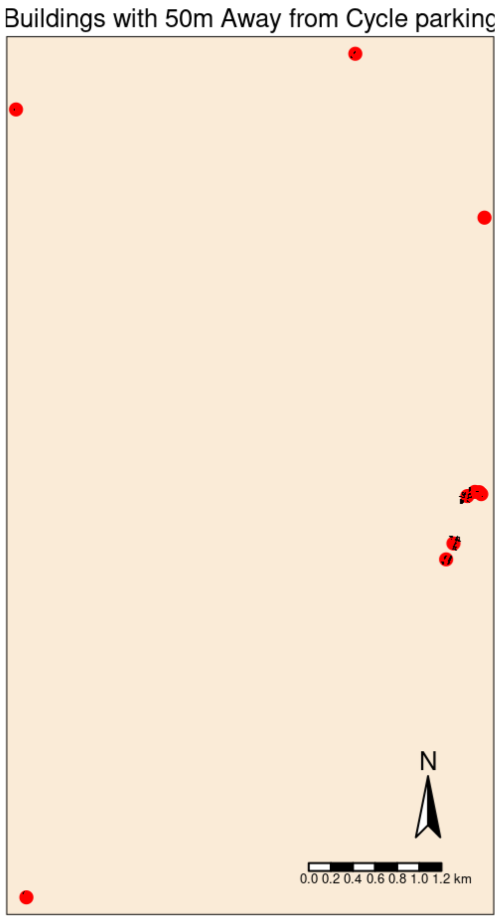
```
tm_shape(Cycle_Pk_in_Haerving) +
  # Represent them as filled polygons
  tm_dots(col = 'red', size = 0.18) +
  # Add the line shapes
  tm_shape(all_building,) +
  # Represent them as lines
  tm_fill( col = 'black', lwd = 3)   +
  tmap_options(check.and.fix = TRUE) +
  tm_layout(
    main.title = 'Buildings in Hovering around Cycle parking', bg.color="antiquewh
ite",
    main.title.size = 0.8,main.title.position="center") +
  tm_compass() + tm_scale_bar() +
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```

```
## Warning: The shape all_building is invalid. See sf::st_is_valid
```

```
## Warning: The shape all_building contains empty units.
```

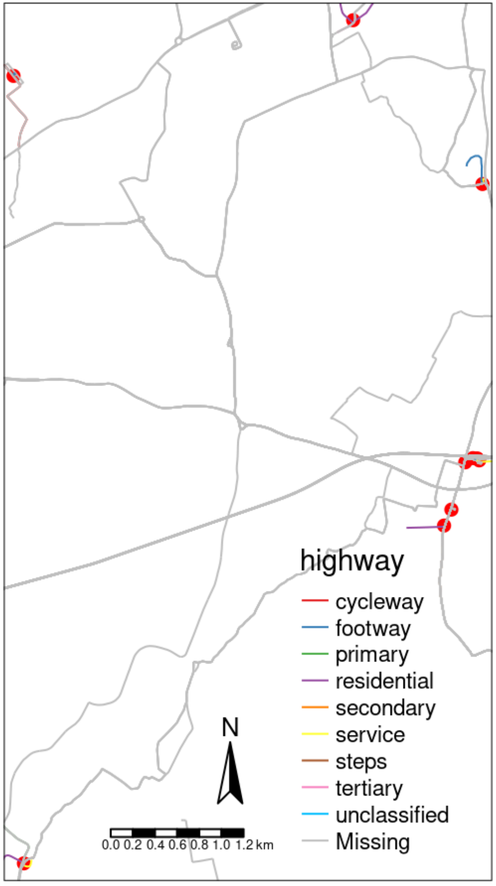

Buildings in Hovering around Cycle parking
N
0.0 0.2 0.4 0.6 0.8 1.0 1.2 km

```
tm_shape(Cycle_Pk_in_Haerving) +
  # Represent them as filled polygons
  tm_dots(col = 'red', size = 0.18) +
  # Add the line shapes
  tm_shape(buildingwithin50m) +
  # Represent them as lines
  tm_fill(col = 'black', lwd = 3)   +
  tmap_options(check.and.fix = TRUE) +
  tm_layout(
    main.title = 'Buildings with 50m Away from Cycle parking', bg.color="antiquewh
ite",
    main.title.size = 0.8,main.title.position="center") +
  tm_compass() + tm_scale_bar() +
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```


Buildings with 50m Away from Cycle parkinc
N
0.0 0.2 0.4 0.6 0.8 1.0 1.2 km

```
tm_shape(Cycle_Pk_in_Haerving) +
  # Represent them as filled polygons
  tm_dots(col = 'red', size = 0.18) +
  # Add the line shapes
  tm_shape(roads19maway) +
  # Represent them as lines
  tm_lines('highway', palette = pall)  +
  tmap_options(check.and.fix = TRUE) +
  tm_layout(
            main.title = 'Roads Within 19m from Cycle parking',
            main.title.size = 0.8,main.title.position="center") +
  tm_compass() + tm_scale_bar() +
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```

Roads Within 19m from Cycle parking



Part 1

Just like many databases around, the dataset “GY7708_2021-22_Assignment_1--Statues-Wikidata.csv” given for this course work contains anomalies. These anomalies affect the structure of the data, making it redundant, contain duplicate and lack integrity among others. The effect of these problems is that it would likely lead to deletion, insertion, and update anomaly, making it difficult to work with our data. Normalization is thus introduced to help solve these problems. As taught in class, the first, second and third normal form are important for database to be in the right structure.

As seen in the GY7708_2021-22_Assignment_1--Statues-Wikidata.csv, it does not comply with the first normal form rule which opines that every column must be in atomic format and should not contain list of duplicated information. This anomaly can be found in the DepictedAltLabel column of the dataset. In order to ensure that the dataset set comply with the first normal form, we need to break the list variables in the rows of DepictedAltLabel column into new ones, allowing the column dataset to be in atomic form. Also, to ensure that the dataset can be identified and traced easily, I created a column “ID” which was used as the primary key for the table.

While I depended on the ID and the Statue column as composite primary keys, I noticed that the table does not conform with the second normal rule as some of the columns have partial dependence. Hence, I partitioned the table into two, keeping the columns inception which completely dependents of ID and statue. The other table contained other columns that are partially dependent. This made the table conform with the second normal form rule, however, it contradicts the third normal form rule.

While the third normal form states that no non-primary variable should depend on another non-primary variable i.e., there should be no transitive dependency for non-prime attributes. To ensure that this principle is met, I divided the table into five, allowing columns that are dependent on other columns stay with their dependee.

For each of the five tables created, a column is selected as the primary key, this allows identification and integrity of the dataset. Also, to link the tables together, I created foreign key.

The foreign key is regarded as a primary key in one table link to another table that contains the similar column but not as its primary key.

To ensure that all the tables are well linked with their dependent and dependee, I confirmed the integrity of the table in r with the function dm_examine_constraints(). This showed that the tables are well linked.

Part 2

The Points of Interest (POIs) assigned to me is Bicycle_parking in the Havering region of London. Using the ST_within function in PostGIS, I was able to identify the POIs assigned to me in the study area. I found out that there are ten Bicycle_parking points in Havering. The Bicycle_parking are distributed across the area, however, there clusters in some part of the study area. When counted and grouped by the ID of the POI, it was observed that there are 10 unique bicycle ID. However, four bicycle_parkings were in the output area ‘E00011935’ and super group H. While 4 other bicycle_parking units were distributed among four output areas, output area E00011914 also had two bicycle_parking.

To further enhance the spatial analysis functions ST_buffer and ST_intersects were used. With the help of the ST_intersect, I was able to check the number of roads that intersect the bicycle_parking points, and the result indicates that three roads intersected the bicycle_parking, and one of them is the ‘cycleway.

According to the Cambridge Cycle Parking Guide 2008, bicycle parking should be situated atleast 20 meters away from the highway. This made me conduct an analysis on the type of roads that intersects a 19-meter buffer around the bicycle_parking. The spatial analysis revealed that 112 different roads intersected the 19-meter buffer around the points. However, 5 of them were tertiary i.e highway. Other road types include footway, service, secondary, residential and primary roads.

Also, according to the Standard for Public parking, bicycle parking can only service building 50 meters around it adequately

These queries were also executed in R to give images of the spatial analysis performed. After connecting the R studio to the database, I performed the same query done in PostGreSQL. I also used tmap to print out the images of the indicating the spatial relationship between POI and other features in the study are.

Reference

1. Camcycle.org.uk. 2008. *How to provide Cycle Parking: a step-by-step guide for planners and providers*. [online] Available at: <https://www.camcycle.org.uk/files/resources/cycleparking/guide/cycleparkingguide.pdf> [Accessed 7 March 2022].

2. Bicycleassociation.org.uk. 2021. *STANDARDS FOR PUBLIC CYCLE PARKING*. [online] Available at: <https://www.bicycleassociation.org.uk/wp-content/uploads/2021/06/05132-Cycle-Parking-and-Security-Standards-June-2021-REV-5.pdf> [Accessed 7 March 2022].