Open Street Map Data Wrangling Process

Analysis Process:

Questions:

Given an assumed valid XML file:

What elements exist within that file?

What attributes do those elements have?

What data types are found for those attributes?

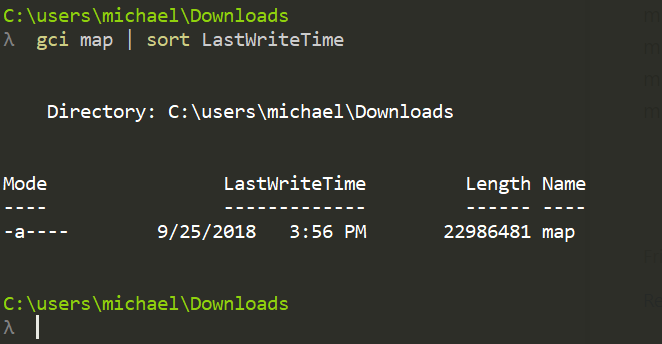
Which elements are nested?

Wrangle

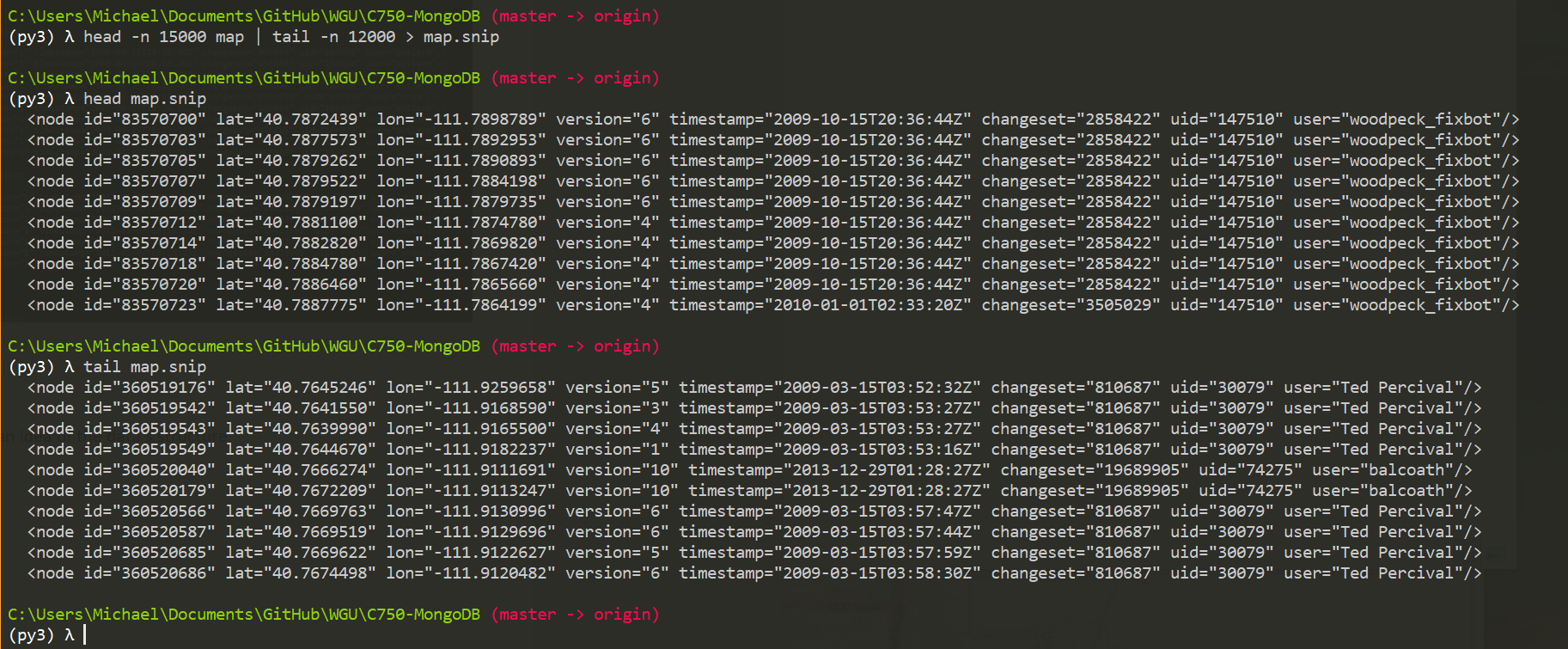
Gather

<https://www.openstreetmap.org/?edit_help=1#map=15/40.7770/-111.8740>

Data source: <https://www.openstreetmap.org/#map=13/40.7478/-111.8452>



How I made the 1.2mb Snip.



Extract

Clean

Store

Explore

Started with a head and tail to get an idea of the data’s structure



This leads me to question what tag structures exist within this data set.

> db.map.find({}, {"\_id":0, "name":1, "nested\_elements":1}).pretty()

{ "name" : "note", "nested\_elements" : null }

{ "name" : "meta", "nested\_elements" : null }

{ "name" : "bounds", "nested\_elements" : null }

{ "name" : "node", "nested\_elements" : [ "tag" ] }

{ "name" : "tag", "nested\_elements" : null }

{ "name" : "nd", "nested\_elements" : null }

{ "name" : "way", "nested\_elements" : [ "nd", "tag" ] }

{ "name" : "member", "nested\_elements" : null }

{ "name" : "relation", "nested\_elements" : [ "tag", "member" ] }

{ "name" : "osm", "nested\_elements" : null }

>

Document Descriptors

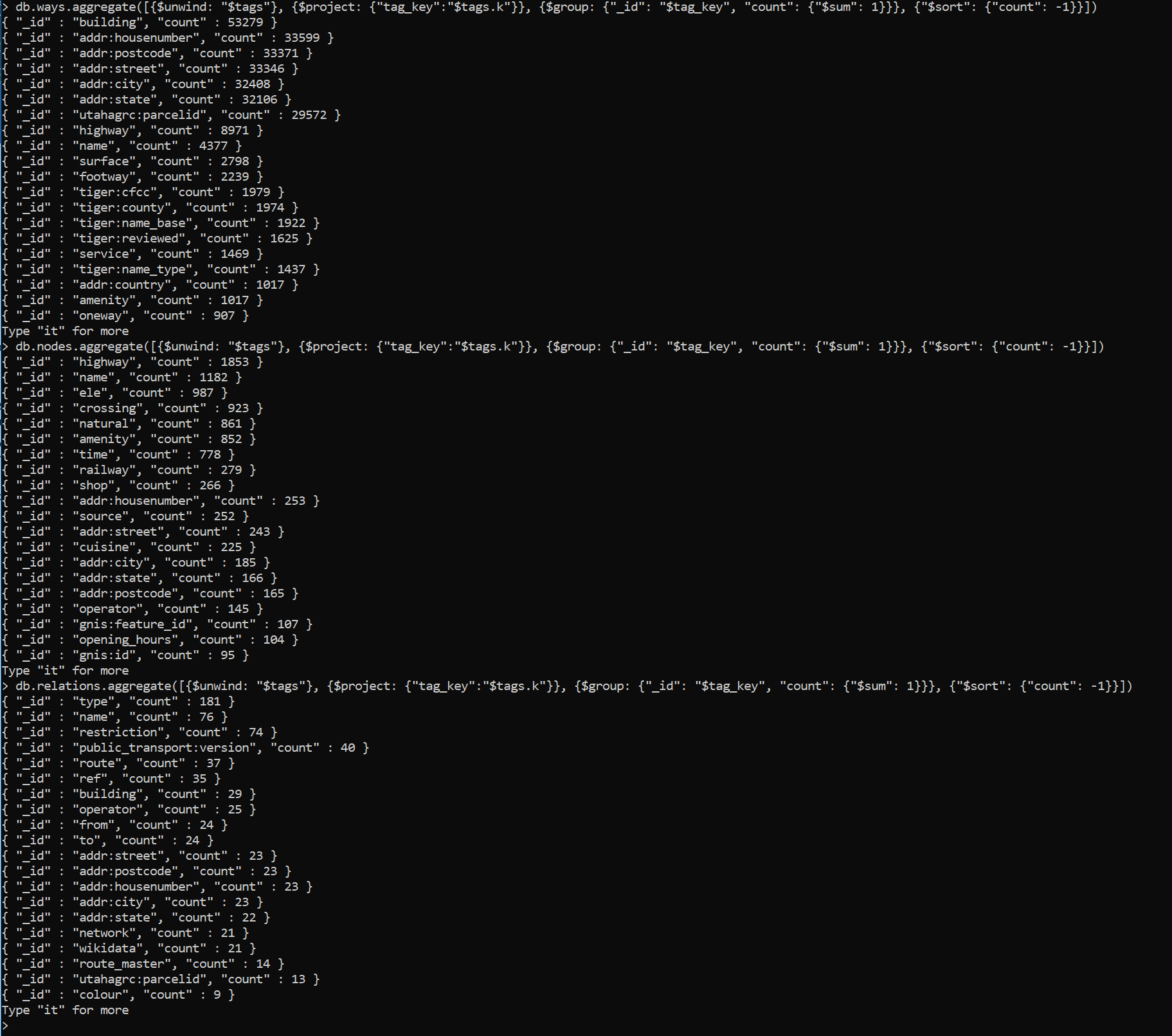
Map Elements

Element Descriptors

Conclude/Predict

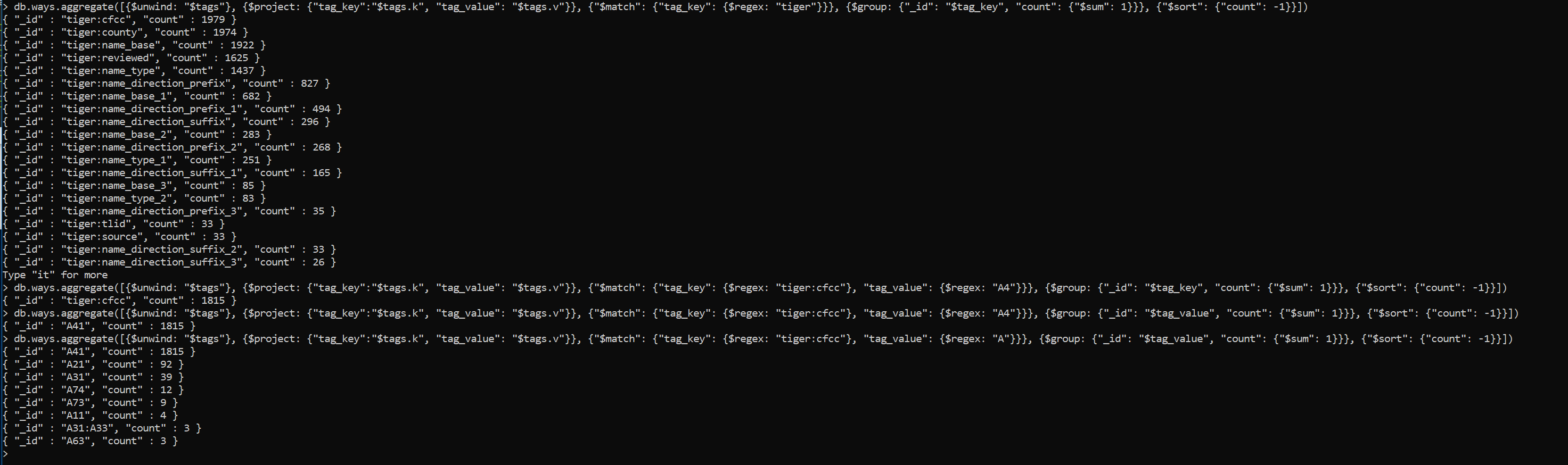
Communicate

Determining volume of tag keys

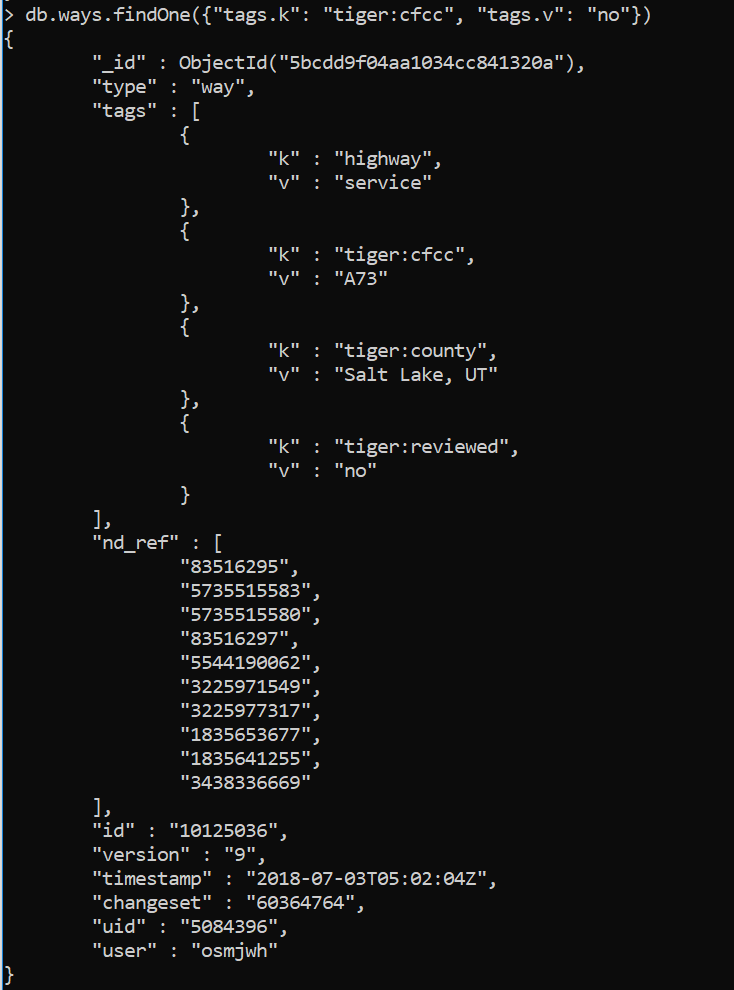


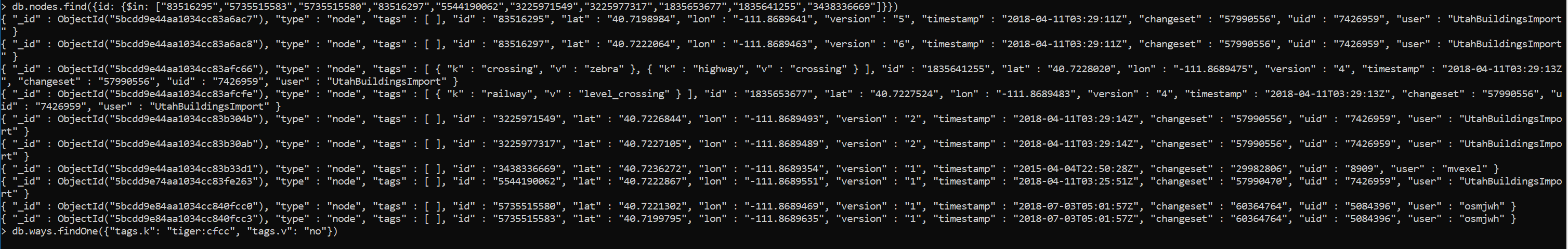
Tiger data:

<https://wiki.openstreetmap.org/wiki/TIGER_to_OSM_Attribute_Map#TIGER_CFCC_to_OSM_Attribute_Pair>



Investigating unreviewed tiger data:





> db.ways.aggregate([{$match: {"tags.k":{$regex: ":"}}}, {$sample : {size: 1}}])

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841bc30"), "type" : "way", "tags" : [ { "k" : "addr:city", "v" : "Salt Lake City" }, { "k" : "addr:housenumber", "v" : "1969" }, { "k" : "addr:postcode", "v" : "84108" }, { "k" : "addr:state", "v" : "UT" }, { "k" : "addr:street", "v" : "Scenic Drive" }, { "k" : "building", "v" : "house" }, { "k" : "utahagrc:parcelid", "v" : "16143530050000" } ], "nd\_ref" : [ "5341987664", "5341986271", "5341974743", "5341973158", "5341973145", "5341981915", "5341980899", "5341972624", "5341972360", "5341984221", "5341986221", "5341987664" ], "id" : "553403504", "version" : "1", "timestamp" : "2018-01-16T04:02:41Z", "changeset" : "55480909", "uid" : "7426959", "user" : "UtahBuildingsImport" }

> db.ways.aggregate([{$match: {"tags.k":{$regex: ":"}}}, {$sample : {size: 1}}])

Leaving Tiger data.

Looking at addr data:

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {"tags.k": {$regex: "^addr:"}}}, {$project: {"tag\_key":"$tags.k"}}, {$group: {"\_id": "$tag\_key", "count": {"$sum": 1}}}, {$sort: {"count": -1}}])

{ "\_id" : "addr:housenumber", "count" : 33599 }

{ "\_id" : "addr:postcode", "count" : 33371 }

{ "\_id" : "addr:street", "count" : 33346 }

{ "\_id" : "addr:city", "count" : 32408 }

{ "\_id" : "addr:state", "count" : 32106 }

{ "\_id" : "addr:country", "count" : 1017 }

{ "\_id" : "addr:housename", "count" : 14 }

{ "\_id" : "addr:place", "count" : 1 }

{ "\_id" : "addr:county", "count" : 1 }

{ "\_id" : "addr:unit", "count" : 1 }

{ "\_id" : "addr:full", "count" : 1 }

>

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {"tags.k": {$regex: "^addr:housename"}}}, {$project: {"housename":"$tags.v"}}])

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84138ab"), "housename" : "Scott M. Matheson Courthouse" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc8413943"), "housename" : "Williams Building" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc8413fa0"), "housename" : "585 Komas" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc841451e"), "housename" : "Mountain America Credit Union" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84147f8"), "housename" : "Hightower Apartments" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84150a6"), "housename" : "633" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84150aa"), "housename" : "The Sundown Apartments" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84150ab"), "housename" : "The Sundown Apartments" }

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84151b6"), "housename" : "Roland Hall" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841d855"), "housename" : "The Drayton" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841d9d3"), "housename" : "Emma" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841eb96"), "housename" : "633" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841ec3a"), "housename" : "First Fleet Condos" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc8422615"), "housename" : "Chateau Normandie" }

>

> db.ways.find({"tags.v": "633", "tags.k": "addr:housename"})

{ "\_id" : ObjectId("5bcdd9f04aa1034cc84150a6"), "type" : "way", "tags" : [ { "k" : "addr:city", "v" : "Salt Lake City" }, { "k" : "addr:housename", "v" : "633" }, { "k" : "addr:housenumber", "v" : "75" }, { "k" : "addr:postcode", "v" : "84102" }, { "k" : "addr:state", "v" : "UT" }, { "k" : "addr:street", "v" : "900 East" }, { "k" : "building", "v" : "retail" }, { "k" : "building:levels", "v" : "1" }, { "k" : "name", "v" : "Tesoro Quick Stop" }, { "k" : "shop", "v" : "convenience" }, { "k" : "source", "v" : "survey" }, { "k" : "utahagrc:parcelid", "v" : "16051270330000" } ], "nd\_ref" : [ "5504527754", "4519715233", "4519715234", "4519715231", "5504527753", "5504527754" ], "id" : "455580054", "version" : "2", "timestamp" : "2018-03-26T06:04:39Z", "changeset" : "57523498", "uid" : "7426959", "user" : "UtahBuildingsImport" }

{ "\_id" : ObjectId("5bcdd9f14aa1034cc841eb96"), "type" : "way", "tags" : [ { "k" : "addr:city", "v" : "Salt Lake City" }, { "k" : "addr:housename", "v" : "633" }, { "k" : "addr:postcode", "v" : "84102" }, { "k" : "addr:state", "v" : "UT" }, { "k" : "addr:street", "v" : "900 East" }, { "k" : "building", "v" : "retail" }, { "k" : "building:levels", "v" : "1" }, { "k" : "source", "v" : "survey" } ], "nd\_ref" : [ "5504527753", "4519715232", "5504527755", "5504527754", "5504527753" ], "id" : "573000859", "version" : "1", "timestamp" : "2018-03-26T06:02:37Z", "changeset" : "57523498", "uid" : "7426959", "user" : "UtahBuildingsImport" }

>

> db.ways.find({"tags.v": "633", "tags.k": "addr:housename"}).pretty()

{

"\_id" : ObjectId("5bcdd9f04aa1034cc84150a6"),

"type" : "way",

"tags" : [

{

"k" : "addr:city",

"v" : "Salt Lake City"

},

{

"k" : "addr:housename",

"v" : "633"

},

{

"k" : "addr:housenumber",

"v" : "75"

},

{

"k" : "addr:postcode",

"v" : "84102"

},

{

"k" : "addr:state",

"v" : "UT"

},

{

"k" : "addr:street",

"v" : "900 East"

},

{

"k" : "building",

"v" : "retail"

},

{

"k" : "building:levels",

"v" : "1"

},

{

"k" : "name",

"v" : "Tesoro Quick Stop"

},

{

"k" : "shop",

"v" : "convenience"

},

{

"k" : "source",

"v" : "survey"

},

{

"k" : "utahagrc:parcelid",

"v" : "16051270330000"

}

],

"nd\_ref" : [

"5504527754",

"4519715233",

"4519715234",

"4519715231",

"5504527753",

"5504527754"

],

"id" : "455580054",

"version" : "2",

"timestamp" : "2018-03-26T06:04:39Z",

"changeset" : "57523498",

"uid" : "7426959",

"user" : "UtahBuildingsImport"

}

{

"\_id" : ObjectId("5bcdd9f14aa1034cc841eb96"),

"type" : "way",

"tags" : [

{

"k" : "addr:city",

"v" : "Salt Lake City"

},

{

"k" : "addr:housename",

"v" : "633"

},

{

"k" : "addr:postcode",

"v" : "84102"

},

{

"k" : "addr:state",

"v" : "UT"

},

{

"k" : "addr:street",

"v" : "900 East"

},

{

"k" : "building",

"v" : "retail"

},

{

"k" : "building:levels",

"v" : "1"

},

{

"k" : "source",

"v" : "survey"

}

],

"nd\_ref" : [

"5504527753",

"4519715232",

"5504527755",

"5504527754",

"5504527753"

],

"id" : "573000859",

"version" : "1",

"timestamp" : "2018-03-26T06:02:37Z",

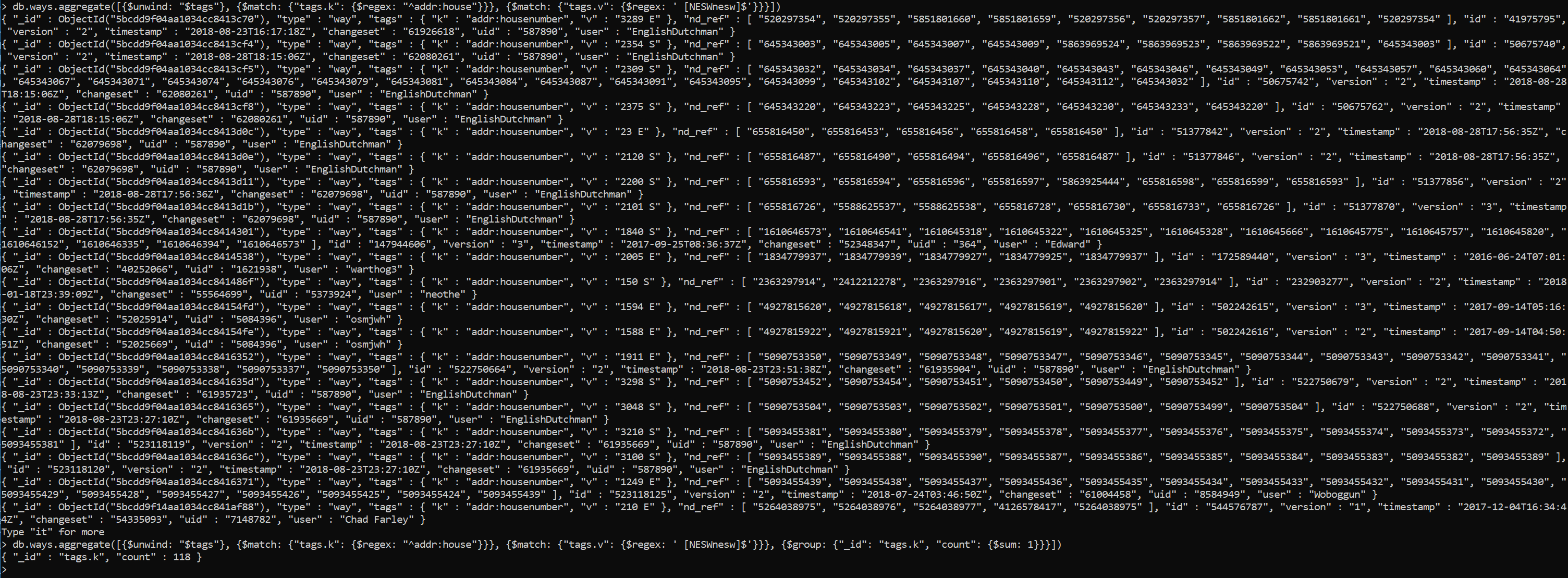
"changeset" : "57523498",

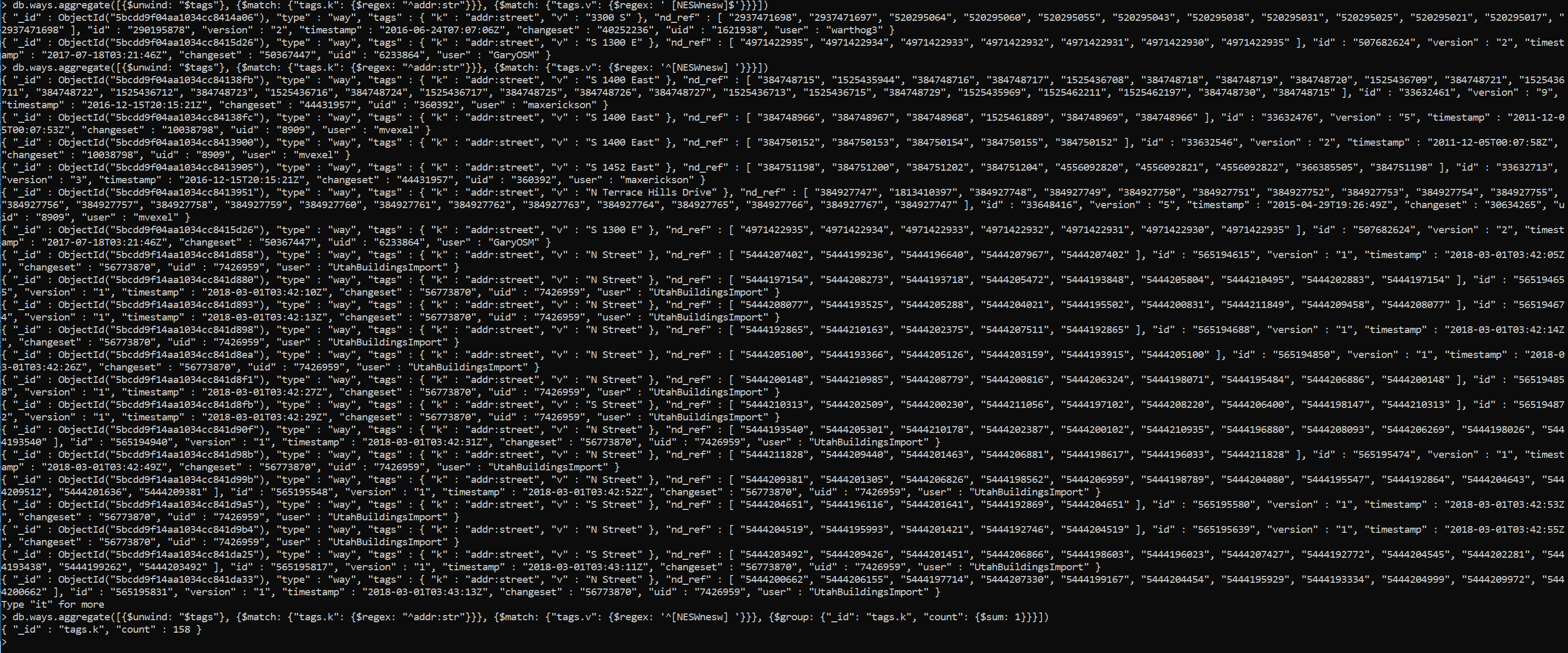
"uid" : "7426959",

"user" : "UtahBuildingsImport"

}

>





Street clean up.

Targeting North, south, east, west

Want to expand single letter to full name

In some cases there are single letters followed by a period.

Need to ignore ‘E Street’, ‘N Street’, ‘S Street’ as these are actual streets.

Also need to ignore “Saint Mary's Drive” and “Saint Mary’s Way”

> db.nodes.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\b[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 51 }

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\b[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 126 }

>

Is it worth it to worry about the possessive false hits?

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {"tags.v": /'[NESWnesw]/}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 112 }

112 are identified with the possessive false hit regex

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\b[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 238 }

There are 238 hits including the possessive hits

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\b[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 126 }

>

Another way of presenting street directions is an abutment to the street/house number.

Based on our investigation, it’s not common in our data set but does exist.

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\d[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

{ "\_id" : "tags.k", "count" : 2 }

> db.node.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\d[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}, {$group: {"\_id": "tags.k", "count": {$sum: 1}}}])

> db.ways.aggregate([{$unwind: "$tags"}, {$match: {$or: [{"tags.k": /^addr:str/},{"tags.k": /^addr:hou/}]}}, {$match: {$and: [{"tags.v": /\d[NESWnesw]\b/}, {"tags.v": {$not: /[NESWnesw] Street$/}}, {"tags.v": {$not: /'[NESWnesw]/}}]}}])

{ "\_id" : ObjectId("5bd4e7084aa1030f38c298dd"), "type" : "way", "tags" : { "k" : "addr:housenumber", "v" : "356s" }, "nd\_ref" : [ "4519759043", "4519759044", "4519759045", "4519759046", "4519759043" ], "id" : "455584436", "version" : "1", "timestamp" : "2016-11-24T23:44:05Z", "changeset" : "43934060", "uid" : "1518893", "user" : "morpher" }

{ "\_id" : ObjectId("5bd4e7084aa1030f38c29929"), "type" : "way", "tags" : { "k" : "addr:housenumber", "v" : "509E" }, "nd\_ref" : [ "4582257998", "4582258001", "4582258002", "4582258003", "4582258004", "4582258005", "4584903669", "4582258000", "4582257999", "4582257998" ], "id" : "462897281", "version" : "6", "timestamp" : "2018-09-28T00:45:51Z", "changeset" : "62993723", "uid" : "6618675", "user" : "keithalleman" }

>