1. synth\_ma.county\_stats
2. synth\_ma.cousub\_stats
3. synth\_ma.blk\_stats
4. county\_health.chr (census data – do not replicate)
5. tiger\_cb14\_500k.county (geographic data – do not replicate)
6. tiger\_cb14\_500k.cousub (geographic data – do not replicate)

select cousubfp, countyfp from tiger.cousub where name = <town name> and statefp=’25’

# Counties

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate as chr\_hs\_grad, chr.college as chr\_college, chr.unemployed as chr\_unemployed, " \

"ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM synth\_ma.county\_stats s " \

"JOIN tiger\_cb14\_500k.county g ON g.statefp = '25' AND g.countyfp = s.ct\_fips " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips"

# Counties/list

sql = "SELECT ct\_name, ct\_fips " \

"FROM synth\_ma.county\_stats"

# Counties/geoms

sql = "SELECT countyfp AS ct\_fips, ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM tiger\_cb14\_500k.county WHERE statefp='25'"

# Counties/stats

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate / 100 as chr\_hs\_grad, chr.college / 100 as chr\_college, chr.unemployed / 100 as chr\_unemployed " \

"FROM synth\_ma.county\_stats s " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips"

# Counties/name/<ct\_name>

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate as chr\_hs\_grad, chr.college as chr\_college, chr.unemployed as chr\_unemployed, " \

"ST\_AsGeoJSON(s.ct\_poly) AS geometry " \

"FROM synth\_ma.county\_stats s " \

"JOIN tiger\_cb14\_500k.county g ON g.statefp = '25' AND g.countyfp = s.ct\_fips " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips " \

"WHERE ct\_name=%s"

# Counties/name/<string:ct\_name>/geom

sql = "SELECT countyfp AS ct\_fips, ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM tiger\_cb14\_500k.county " \

"WHERE statefp='25' AND name=%s"

# counties/name/<string:ct\_name>/stats

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate as chr\_hs\_grad, chr.college as chr\_college, chr.unemployed as chr\_unemployed " \

"FROM synth\_ma.county\_stats s " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips " \

"WHERE s.ct\_name=%s"

# counties/id/<string:ct\_fips>

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate as chr\_hs\_grad, chr.college as chr\_college, chr.unemployed as chr\_unemployed, " \

"ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM synth\_ma.county\_stats s " \

"JOIN tiger\_cb14\_500k.county g ON g.statefp = '25' AND g.countyfp = s.ct\_fips " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips " \

"WHERE ct\_fips=%s"

# counties/id/<string:ct\_fips>/geom

sql = "SELECT countyfp AS ct\_fips, ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM tiger\_cb14\_500k.county " \

"WHERE statefp='25' AND countyfp=%s"

# counties/id/<string:ct\_fips>/stats

sql = "SELECT s.ct\_fips, s.ct\_name, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"chr.hs\_graduate as chr\_hs\_grad, chr.college as chr\_college, chr.unemployed as chr\_unemployed " \

"FROM synth\_ma.county\_stats s " \

"JOIN county\_health.chr ON chr.statefp = '25' AND chr.release\_year = 2016 AND chr.countyfp = s.ct\_fips " \

"WHERE ct\_fips=%s"

# Cousubs

sql = "SELECT s.ct\_fips, s.ct\_name, s.cs\_fips, s.cs\_name, s.sq\_mi, s.pop, s.pop\_sm, " \

"CASE WHEN s.pop > 0 THEN s.pop\_male / s.pop ELSE 0 END AS pct\_male, " \

"CASE WHEN s.pop > 0 THEN s.pop\_female / s.pop ELSE 0 END AS pct\_female, " \

"ST\_AsGeoJSON(g.the\_geom) AS geometry " \

"FROM synth\_ma.cousub\_stats s, tiger\_cb14\_500k.cousub g " \

"WHERE g.statefp = '25' AND g.countyfp = s.ct\_fips AND g.cousubfp = s.cs\_fips AND s.cs\_fips != '00000'"

# cousubs/geoms

sql = "SELECT countyfp AS ct\_fips, cousubfp AS cs\_fips, ST\_AsGeoJSON(the\_geom) AS geometry " \

"FROM tiger\_cb14\_500k.cousub " \

"WHERE statefp='25' AND cousubfp != '00000'"

# cousubs/stats

sql = "SELECT s.ct\_fips, s.ct\_name, s.cs\_fips, s.cs\_name, s.sq\_mi, s.pop, s.pop\_sm, " \

"CASE WHEN s.pop > 0 THEN s.pop\_male / s.pop ELSE 0 END AS pct\_male, " \

"CASE WHEN s.pop > 0 THEN s.pop\_female / s.pop ELSE 0 END AS pct\_female " \

"FROM synth\_ma.cousub\_stats s " \

"WHERE s.cs\_fips != '00000'"

# block\_window?minx=-71.26&maxx=-71.22&miny=42.49&maxy=42.51

sql = "SELECT s.block\_id, s.sq\_mi, s.pop, s.pop\_male / s.pop as pct\_male, s.pop\_female / s.pop as pct\_female, s.pop\_sm, " \

"ST\_AsGeoJSON(s.blk\_poly) AS geometry " \

"FROM synth\_ma.blk\_stats s " \

"WHERE s.blk\_poly && ST\_SetSRID(ST\_MakeBox2D(ST\_Point(%s,%s), ST\_Point(%s,%s)), 4269) AND s.pop > 0"

# ResourceTable

ResourceTable extends BaseHasResource

getResource() returns byte array

new String(resourceTable.getResource(), “UTF-8”)

BaseHapiFhirDao.toResource:

**public** <R **extends** IBaseResource> R toResource(Class<R> theResourceType, BaseHasResource theEntity, **boolean** theForHistoryOperation) {

String resourceText = **null**;

**switch** (theEntity.getEncoding()) {

**case** JSON:

**try** {

resourceText = **new** String(theEntity.getResource(), "UTF-8");

} **catch** (UnsupportedEncodingException e) {

**throw** **new** Error("Should not happen", e);

}

**break**;

**case** JSONC:

resourceText = GZipUtil.decompress(theEntity.getResource());

**break**;

}

IParser parser = theEntity.getEncoding().newParser(getContext(theEntity.getFhirVersion()));

R retVal;

**try** {

retVal = parser.parseResource(theResourceType, resourceText);

} **catch** (Exception e) {

StringBuilder b = **new** StringBuilder();

b.append("Failed to parse database resource[");

b.append(theResourceType);

b.append("/");

b.append(theEntity.getIdDt().getIdPart());

b.append(" (pid ");

b.append(theEntity.getId());

b.append(", version ");

b.append(myContext.getVersion().getVersion());

b.append("): ");

b.append(e.getMessage());

String msg = b.toString();

***ourLog***.error(msg, e);

**throw** **new** DataFormatException(msg, e);

}

**if** (retVal **instanceof** IResource) {

IResource res = (IResource) retVal;

retVal = populateResourceMetadataHapi(theResourceType, theEntity, theForHistoryOperation, res);

} **else** {

IAnyResource res = (IAnyResource) retVal;

retVal = populateResourceMetadataRi(theResourceType, theEntity, theForHistoryOperation, res);

}

**return** retVal;

}