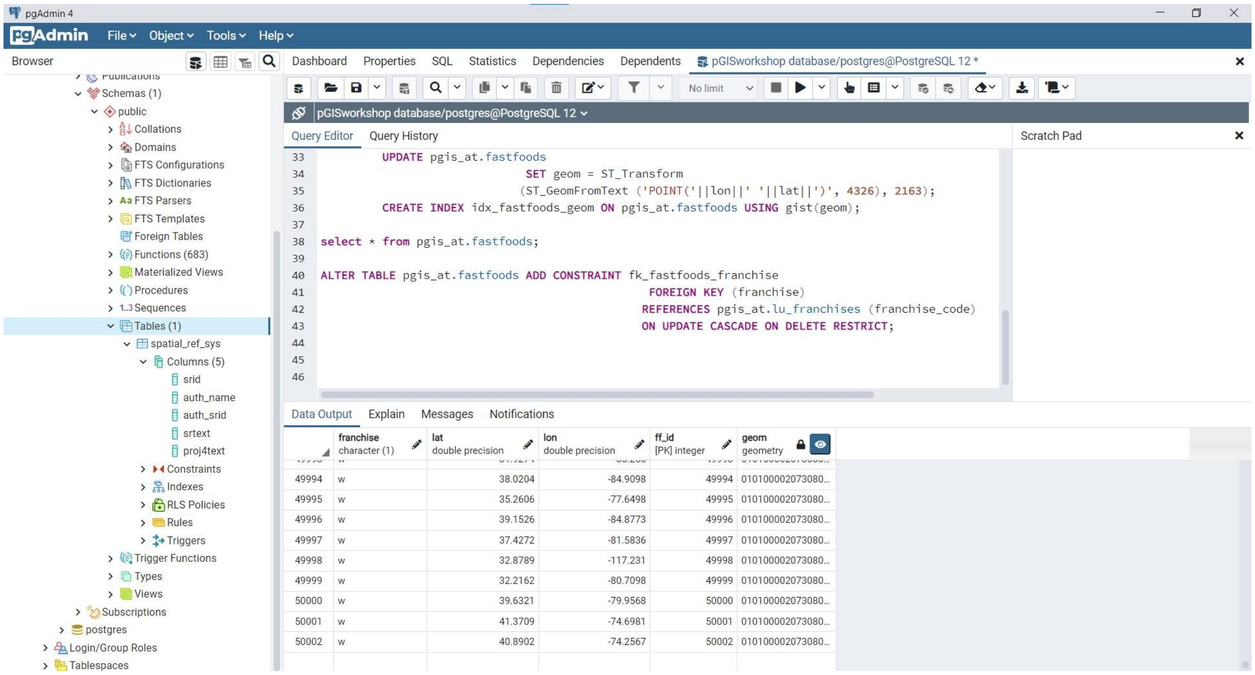
Q1. 50002

Q2.



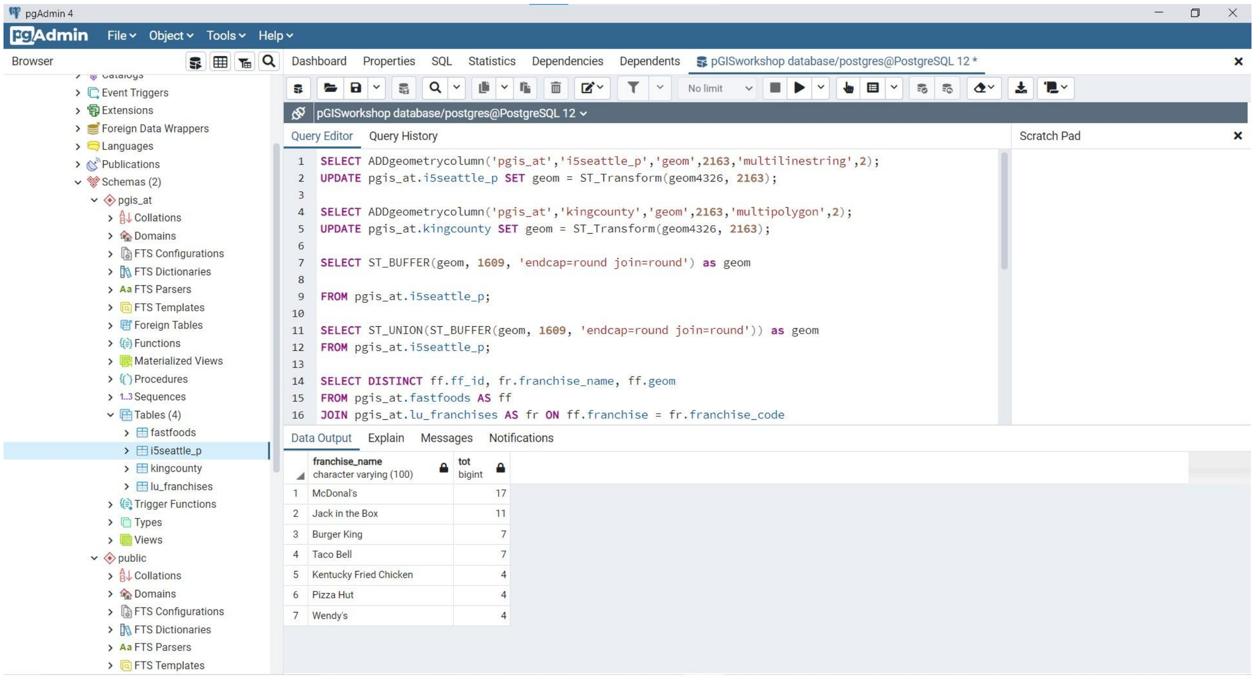
```sql
         UPDATE pgis_at.fastfoods
                              SET geom = ST_Transform
                             (ST_GeomFromText ('POINT('||lon||' '||lat||')', 4326), 2163);
         CREATE INDEX idx_fastfoods_geom ON pgis_at.fastfoods USING gist(geom);

select * from pgis_at.fastfoods;

ALTER TABLE pgis_at.fastfoods ADD CONSTRAINT fk_fastfoods_franchise
                                                FOREIGN KEY (franchise)
                                               REFERENCES pgis_at.lu_franchises (franchise_code)
                                               ON UPDATE CASCADE ON DELETE RESTRICT;
```

	franchise character (1)	lat double precision	lon double precision	ff_id [PK] integer	geom geometry
49994	w	38.0204	-84.9098	49994	010100002073080...
49995	w	35.2606	-77.6498	49995	010100002073080...
49996	w	39.1526	-84.8773	49996	010100002073080...
49997	w	37.4272	-81.5836	49997	010100002073080...
49998	w	32.8789	-117.231	49998	010100002073080...
49999	w	32.2162	-80.7098	49999	010100002073080...
50000	w	39.6321	-79.9568	50000	010100002073080...
50001	w	41.3709	-74.6981	50001	010100002073080...
50002	w	40.8902	-74.2567	50002	010100002073080...

Q3.



```sql
SELECT ADDgeometrycolumn('pgis_at','i5seattle_p','geom',2163,'multilinestring',2);
UPDATE pgis_at.i5seattle_p SET geom = ST_Transform(geom4326, 2163);

SELECT ADDgeometrycolumn('pgis_at','kingcounty','geom',2163,'multipolygon',2);
UPDATE pgis_at.kingcounty SET geom = ST_Transform(geom4326, 2163);

SELECT ST_BUFFER(geom, 1609, 'endcap=round join=round') as geom

FROM pgis_at.i5seattle_p;

SELECT ST_UNION(ST_BUFFER(geom, 1609, 'endcap=round join=round')) as geom
FROM pgis_at.i5seattle_p;

SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
```

	franchise_name character varying (100)	tot bigint
1	McDonal's	17
2	Jack in the Box	11
3	Burger King	7
4	Taco Bell	7
5	Kentucky Fried Chicken	4
6	Pizza Hut	4
7	Wendy's	4

The following two lines create a buffer around the I-5 highway. ST_BUFFER is the command for creating buffers around features. The feature in question can be a point, line or polygon. The buffer command denotes the specified geometry column, buffer distance as measured in meters, and buffer style. This code ends up creating a buffer around the highway that spans 1069 meters.

```
SELECT ST_BUFFER(geom, 1609, 'endcap=round join=round') as geom
FROM pgis_at.i5seattle_p;
```

The following code uses the ST_UNION command to create a dissolved buffer and clean up the output.

```
SELECT ST_UNION(ST_BUFFER(geom, 1609, 'endcap=round join=round')) as geom
FROM pgis_at.i5seattle_p;
```

The following line of code uses the command ST_WITHIN and spatial join to find points within a specific distance of a particular feature. This line finds restaurant locations that are within a mile from the I-5 highway. We also need to use a spatial join because we are comparing two geometries of fastfood and I-5. We use the command ST_DWITHIN to perform this spatial join. We include the fastfoods geometry, I-5 geometry and distance in this operation, and this helps us find which fastfoods are within one mile from the I-5 highway.

```
SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 1609*1)
ORDER BY ff.ff_id;
```

The following line helps us create a view that we can use later, this will come in handy because we want to include these points on a map

```
CREATE VIEW pGIS_at.ffwithin1 AS
SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 1609*1)
ORDER BY ff.ff_id;
```

These last few lines of code help us aggregate spatial data with a specific guideline or criteria. In order to aggregate our data, we use the command GROUP BY. This code helps us find how many fastfood restaurants in every chain are within a specific distance of the I-5 highway.

```
SELECT fr.franchise_name, COUNT (DISTINCT ff.ff_id) AS tot
```

```
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 1609*1)
GROUP BY fr.franchise_name
ORDER BY tot DESC;
```

Q4. (-1612831.62950272 498633.490371454)

Q5.

```
SELECT franchise_name

FROM fastfoods

Join pgis_at.fastfoods ff ON kingcounty = lu_franchise

JOIN pgis_at.kingcounty kc ON ST_CONTAINS (kingcounty.geometry, table.geometry)
```

Q6. McDonalds, Burger King, Kentucky Fried Chicken are the top 3 closest and then Taco Bell is extremely close behind

Q7 and onwards

```
        CREATE EXTENSION postgis;
CREATE SCHEMA pGIS_at;
CREATE TABLE pGIS_at.lu_franchises(
                                franchise_code char(1) PRIMARY KEY,
                                franchise_name varchar(100));

INSERT INTO pGIS_at.lu_franchises (franchise_code, franchise_name)
VALUES ('b', 'Wendy'),
                  ('c', 'Red Robins'),
                  ('d', 'Qdoba'),
                  ('e', 'Wienerschnitzel'),
                  ('f', 'Taco Bell'),
                  ('g', 'Mod'),
                  ('h', 'Sonic'),
                  ('i', 'McDonalds'),
                  ('j', 'Dairy Queen'),
                                                  ('k', 'A&W Queen');

CREATE TABLE pGIS_at.fastfoods (
                                franchise char(1) NOT NULL,
                                lat double precision,
                            lon double precision);

COPY pGIS_at.fastfoods FROM 'C:\Users\Aditi\Desktop\GEOG 465 Lab 4\Final
Part\newfastfoodmaps_locations - Sheet1.csv' DELIMITER ',';

select * from pgis_at.fastfoods;

ALTER TABLE pgis_at.fastfoods ADD COLUMN ff_id SERIAL PRIMARY KEY;

   SELECT AddGeometryColumn ('pgis_at','fastfoods','geom', 2163,'POINT',2);
       UPDATE pGIS_at.fastfoods
                       SET geom = ST_Transform
                      (ST_GeomFromText ('POINT('||lon||' '||lat||')', 4326), 2163);
       CREATE INDEX idx_fastfoods_geom ON pGIS_at.fastfoods USING gist(geom);

           ALTER TABLE pgis_at.fastfoods ADD CONSTRAINT fk_fastfoods_franchise
                                     FOREIGN KEY (franchise)
                                    REFERENCES pgis_at.lu_franchises (franchise_code)
                                    ON UPDATE CASCADE ON DELETE RESTRICT;

SELECT ADDgeometrycolumn('pgis_at','i5seattle_p','geom',2163,'multilinestring',2);
UPDATE pgis_at.i5seattle_p SET geom = ST_Transform(geom4326, 2163);

SELECT ADDgeometrycolumn('pgis_at','kingcounty','geom',2163,'multipolygon',2);
UPDATE pgis_at.kingcounty SET geom = ST_Transform(geom4326, 2163);


SELECT ST_BUFFER(geom, 3218, 'endcap=round join=round') as geom

FROM pgis_at.i5seattle_p;

SELECT ST_UNION(ST_BUFFER(geom, 3218, 'endcap=round join=round')) as geom
FROM pgis_at.i5seattle_p;

SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 3218*1)
ORDER BY ff.ff_id;
```

```sql
CREATE VIEW pGIS_at.ffwithin1 AS
SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 3218*1)
ORDER BY ff.ff_id;

SELECT fr.franchise_name, COUNT (DISTINCT ff.ff_id) AS tot
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 3218*1)
GROUP BY fr.franchise_name
ORDER BY tot DESC;

SELECT county, ST_CENTROID(geom) AS centroid

                    FROM pgis_at.kingcounty;

SELECT county, ST_AsText(ST_CENTROID(geom)) AS centroid

                      FROM pgis_at.kingcounty;

SELECT franchise_name

FROM fastfoods

Join pgis_at.fastfoods ff ON kingcounty = lu_franchise

JOIN pgis_at.kingcounty kc ON ST_CONTAINS (kingcounty.geometry, table.geometry)


CREATE TABLE pgis_at.dbuffer1 AS

SELECT st_union( ST_Buffer(geom, 3218, 'endcap=round join=round')) as geom

FROM pgis_at.i5seattle_p;
```

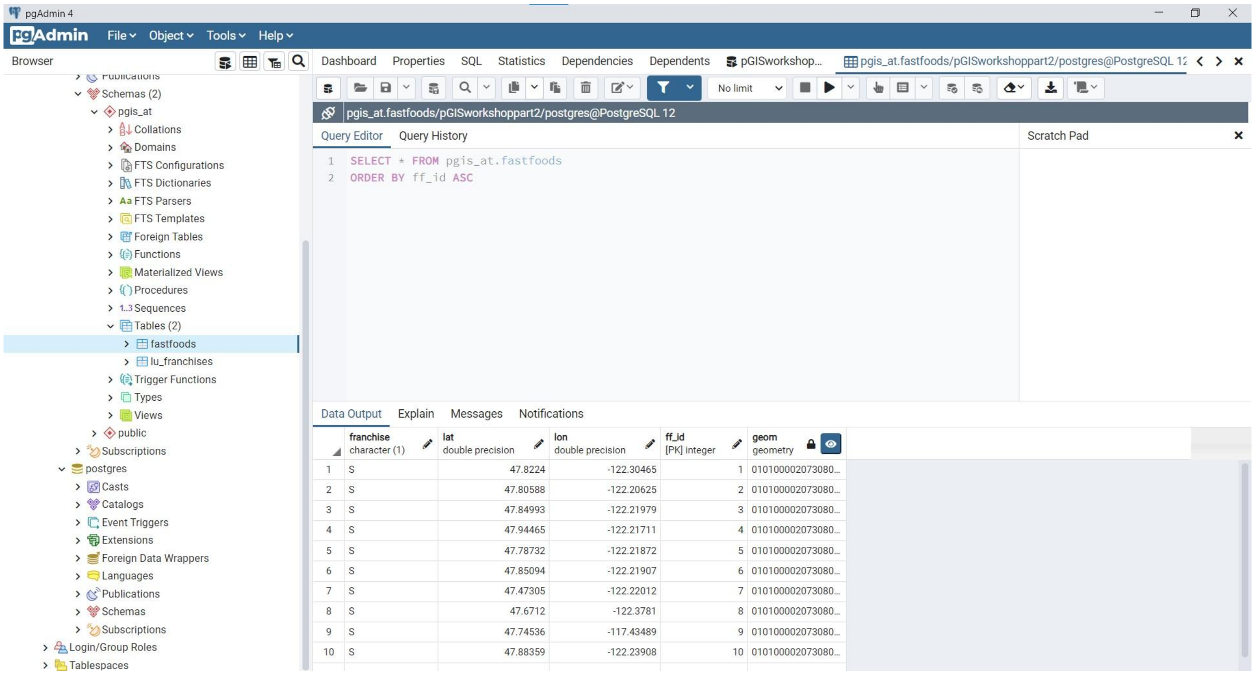
Q7.

```
SELECT DISTINCT ff.ff_id, fr.franchise_name, ff.geom
FROM pgis_at.fastfoods AS ff
JOIN pgis_at.lu_franchises AS fr ON ff.franchise = fr.franchise_code
JOIN pgis_at.i5seattle_p AS i5 ON ST_DWITHIN(ff.geom, i5.geom, 3218*1)
ORDER BY ff.ff_id;
```

Q8.

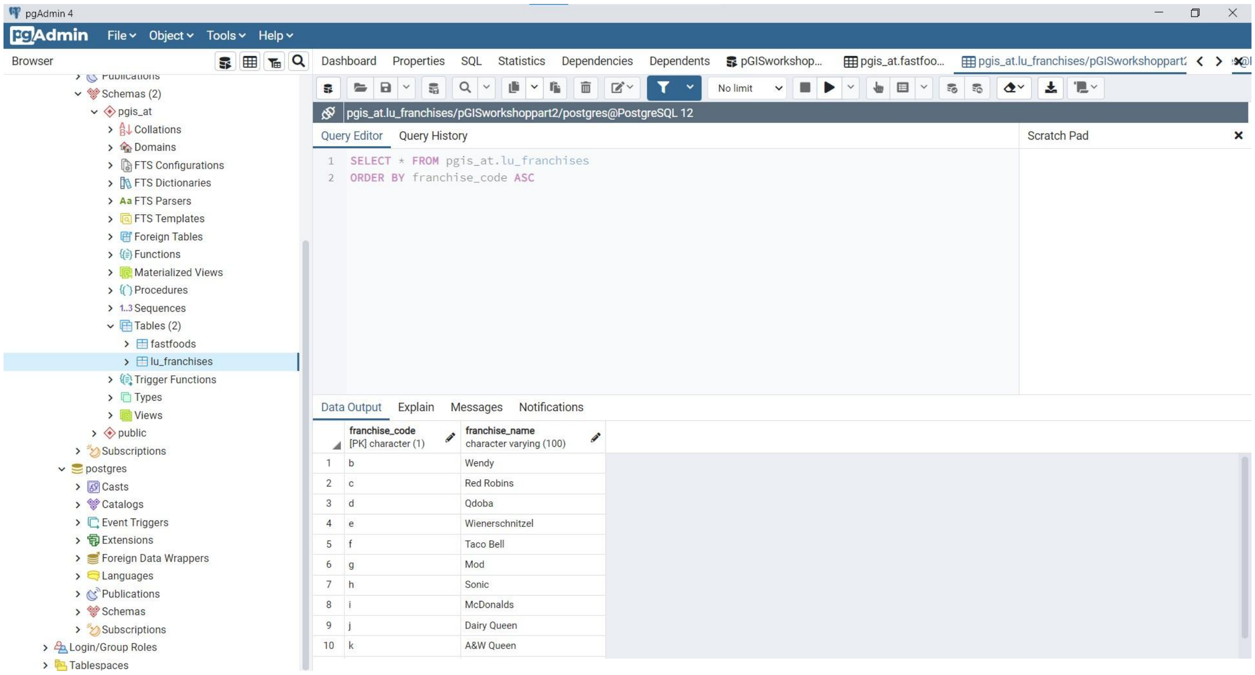
```
SELECT fr.franchise_name,
ST_distance(ff.geom, ST_centroid(pgis_at.kingcounty.geom)) AS Distance
FROM pgis_at.lu_franchises fr
JOIN pgis_at.fastfoods ff
ON ff.franchise = fr.franchise_code,
pgis_at.kingcounty
ORDER BY Distance ASC;
```

Q9.



```sql
SELECT * FROM pgis_at.fastfoods
ORDER BY ff_id ASC
```

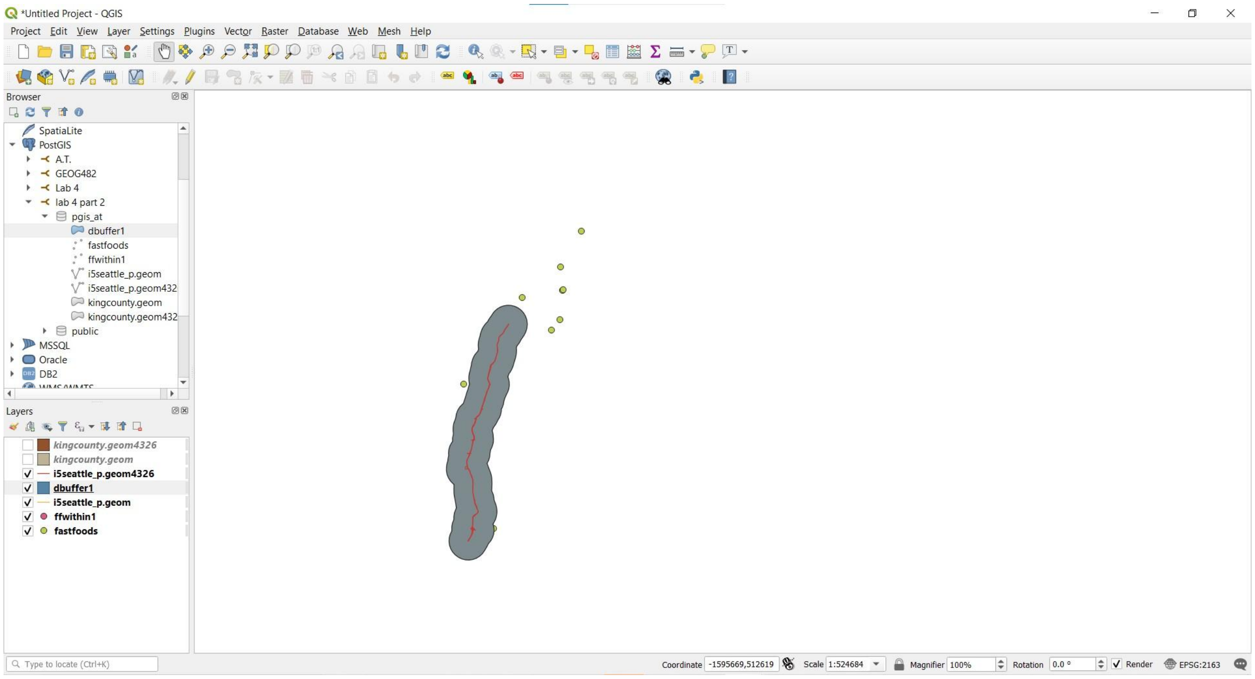
	franchise character (1)	lat double precision	lon double precision	ff_id [PK] integer	geom geometry
1	S	47.8224	-122.30465	1	010100002073080...
2	S	47.80588	-122.20625	2	010100002073080...
3	S	47.84993	-122.21979	3	010100002073080...
4	S	47.94465	-122.21711	4	010100002073080...
5	S	47.78732	-122.21872	5	010100002073080...
6	S	47.85094	-122.21907	6	010100002073080...
7	S	47.47305	-122.22012	7	010100002073080...
8	S	47.6712	-122.3781	8	010100002073080...
9	S	47.74536	-117.43489	9	010100002073080...
10	S	47.88359	-122.23908	10	010100002073080...



```sql
SELECT * FROM pgis_at.lu_franchises
ORDER BY franchise_code ASC
```

	franchise_code [PK] character (1)	franchise_name character varying (100)
1	b	Wendy
2	c	Red Robins
3	d	Qdoba
4	e	Wienerschnitzel
5	f	Taco Bell
6	g	Mod
7	h	Sonic
8	i	McDonalds
9	j	Dairy Queen
10	k	A&W Queen

Q10.



☐ kingcounty.geom4326
☐ kingcounty.geom
☑ i5seattle_p.geom4326
☑ dbuffer1
☑ i5seattle_p.geom
☑ ffwithin1
☑ fastfoods

Coordinate -1595669,512619 Scale 1:524684 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:2163