Making Routes and Visualizing in QGIS Using SQL

Select the CTA Stop:

```
SELECT * FROM cta_railstations
WHERE gid=2;
```

With your CTA stop buffering to speed up the process make edges with source and target geometries:

```
WITH cta AS
(SELECT * FROM cta_railstations
WHERE gid=2)
SELECT row_number() OVER (ORDER BY chicagostreets.gid)::integer AS gid,
chicagostreets.gid AS id, chicagostreets."STREET_NAM" AS name, chicagostreets."LENGTH" AS cost,
chicagostreets.geom,
st_startpoint(ST_LineMerge(chicagostreets.geom)) as source,
st_endpoint(ST_LineMerge(chicagostreets.geom)) as target
FROM cta LEFT JOIN chicagostreets
ON st_within(chicagostreets.geom, st_setsrid(st_buffer(cta.geom, 5280), 3435));
```

Make each node distinct so there are no repeats and you can identify them as start and end points:

```
WITH cta AS
        (SELECT * FROM cta_railstations
        WHERE gid=2),
edge AS
        (SELECT row_number() OVER (ORDER BY chicagostreets.gid)::integer AS gid,
        chicagostreets.gid AS id, chicagostreets."STREET_NAM" AS name, chicagostreets."LENGTH" AS cost,
chicagostreets.geom,
        st_startpoint(ST_LineMerge(chicagostreets.geom)) as source,
        st_endpoint(ST_LineMerge(chicagostreets.geom)) as target
        FROM cta LEFT JOIN chicagostreets
        ON st_within(chicagostreets.geom, st_setsrid(st_buffer(cta.geom, 5280), 3435)))
SELECT row_number() OVER (ORDER BY a.gid)::integer AS gid, a.gid AS geom
        FROM    (SELECT DISTINCT edge.source AS gid FROM edge
                UNION
                SELECT DISTINCT edge.target AS gid FROM edge) AS a
                GROUP BY a.gid;
```

Now Join the edges and nodes to make a network:

```
WITH cta AS
        (SELECT * FROM cta_railstations
        WHERE gid=2),
edge AS
        (SELECT row_number() OVER (ORDER BY chicagostreets.gid)::integer AS gid,
        chicagostreets.gid AS id, chicagostreets."STREET_NAM" AS name, chicagostreets."LENGTH" AS cost,
chicagostreets.geom,
        st_startpoint(ST_LineMerge(chicagostreets.geom)) as source,
        st_endpoint(ST_LineMerge(chicagostreets.geom)) as target
        FROM cta LEFT JOIN chicagostreets
        ON st_within(chicagostreets.geom, st_setsrid(st_buffer(cta.geom, 5280), 3435))),
node AS
        (SELECT row_number() OVER (ORDER BY a.gid)::integer AS gid, a.gid AS geom
                FROM  (SELECT DISTINCT edge.source AS gid FROM edge
                        UNION
                        SELECT DISTINCT edge.target AS gid FROM edge) AS a
                GROUP BY a.gid)
SELECT edge.gid, edge.id, edge.name, edge.cost, edge.geom, source.gid as source, target.gid as target
FROM edge
        JOIN node AS source ON edge.source = source.geom
        JOIN node AS target ON edge.target = target.geom;
```

You can route from one point to another:

(XX = Network Name; ### = Source ID # or Target ID #)

```
SELECT seq,id1 as node, id2 as edge, route.cost, XX.name as streetname, XX.geom
FROM pgr_dijkstra('
 SELECT gid AS id, source::integer, target::integer, cost::double precision AS cost
 FROM XX', ###, ###, false, false )
AS route
LEFT JOIN XX
ON route.id2 = XX.gid;
```

You can route from one point to many points:

(XX = Network Name; ### = Source ID # or Target ID #)

```
SELECT seq,id1 as path, id2 as node, id3 as edge, route.cost, XX.name, XX.geom
FROM pgr_kdijkstraPath('
 SELECT gid AS id,
 source::integer,
 target::integer,
 cost::double precision AS cost
 FROM XX', ###, array[###,###,###,###],
 false, false )
AS route
LEFT JOIN XX
ON route.id3 = XX.gid;
```