create table DATA\_SET\_T

(

-- 01-01. 기본 변수

spo\_cd character varying(50), -- cell 코드

grs\_x character varying(50), -- GRS80 UTM-K 좌표 x

grs\_y character varying(50), -- GRS80 UTM-K 좌표 y

geom geometry(MultiPolygon, 5179),

-- 01-02. 와이파이 변수

w\_pnt numeric, -- 와이파이 Cell 위치 여부(1 혹은 0)

w\_pnt\_150m numeric, -- 와이파이 반경 150m 위치 여부

-- 01-03. 인구 변수

liv\_pp numeric, -- 생산가능 인구

-- 01-04. 거리 변수

-- + 요소

cal\_dist numeric, -- Cell별 500m 문화재 이내 거리

tou\_dist numeric, -- Cell별 500m 이내 관광지의 거리

pub\_dist numeric, -- Cell별 집합건물 500m 이내 거리

hot\_dist numeric, -- Cell별 500m 숙박업소 이내 거리

mul\_dist numeric, -- Cell별 500m 다중이용시설 이내 거리

bus\_dist numeric, -- Cell별 500m 버스정류장 이내 거리

food\_dist numeric, -- Cell별 500m 음식점 이내 거리

caf\_dist numeric, -- Cell별 500m 집단급식소 이내 거리

-- - 요소

sch\_dist numeric, -- Cell별 500m 학교 이내 거리

-- 01-05. 지수 변수

liv\_w numeric, -- 거주인구(생산가능인구) 지수

cal\_w numeric, -- 문화재 지역 지수

tou\_w numeric, -- 관광지 지역 지수

pub\_w numeric, -- 집합건물 지수

hot\_w numeric, -- 숙박업소 지수

mul\_w numeric, -- 다중이용시설 지수

bus\_w numeric, -- 버스정류장 지수

food\_w numeric, -- 음식점 지수

caf\_w numeric, -- 집단급식소 지수

sch\_w numeric, -- 학교 지수

pnt\_w numeric, -- 와이파이 지수

pnt\_150m\_w numeric, --와이파이 반경 150m 지수

tt\_w numeric -- 총 지수값

);

CREATE INDEX DATA\_SET\_geom\_idx ON public.DATA\_SET\_T USING gist (geom);

with kb\_GRID\_01\_1 as

(

select

gid as spo\_cd,

ST\_X(ST\_CENTROID(geom)) as grs\_x,

ST\_Y(ST\_CENTROID(geom)) as grs\_y,

geom

from v\_liv\_pop

)

insert into DATA\_SET\_T select \* from kb\_GRID\_01\_1;

-- 와이파이 설치지역

with kb\_GRID\_02\_1 as

(

select b.spo\_cd

from B\_WIFI\_POINT a, DATA\_SET\_T b

where ST\_INTERSECTS(a.geom, b.geom)

)

update DATA\_SET\_T as grid set w\_pnt = 1 from kb\_GRID\_02\_1 as kb where grid.spo\_cd = kb.spo\_cd;

update DATA\_SET\_T as grid set w\_pnt = 0 where w\_pnt is null;

--select count(\*) from DATA\_\_SET\_T where w\_pnt = 1;

-- 반경 150m내 와이파이 설치지역

with kb\_GRID\_02\_2 as

(

select b.spo\_cd

from B\_WIFI\_POINT a, DATA\_SET\_T b

where ST\_INTERSECTS(st\_buffer(a.geom,150), b.geom)

)

update DATA\_SET\_T as grid set w\_pnt\_150m = 1 from kb\_GRID\_02\_2 as kb where grid.spo\_cd = kb.spo\_cd;

update DATA\_SET\_T as grid set w\_pnt\_150m = 0 where w\_pnt\_150m is null;

--select count(\*) from DATA\_SET\_T where w\_pnt\_150m = 1;

-- 생산가능인구

with kb\_GRID\_03\_2 as

(

select b.spo\_cd, COALESCE(round(a.val,2),0) as val

from v\_liv\_pop a, DATA\_SET\_T b

where ST\_INTERSECTS(ST\_CENTROID(a.geom), b.geom)

)

update DATA\_SET\_T as grid

set liv\_pp = pop.val

from kb\_GRID\_03\_2 as pop

where grid.spo\_cd = pop.spo\_cd;

update DATA\_SET\_T set liv\_pp = 0.00 where liv\_pp is null;

-- 문화재와 500m 이내의 거리 계산

-- 문화재에 포함되는 격자 데이터를 5km 격자만 추출하여 거리계산 값을 생성하고, 그 이외 격자는 -1 데이터로 업데이트 한다.

with kb\_GRID\_04\_1 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_heritage) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set cal\_dist = kb.meter from kb\_GRID\_04\_1 as kb where grid.spo\_cd = kb.spo\_cd;

-- 관광지와 500m 이내의 거리 계산

with kb\_GRID\_04\_2 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_tour\_zone) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set tou\_dist = kb.meter from kb\_GRID\_04\_2 as kb where grid.spo\_cd = kb.spo\_cd;

-- 집합건물과 500m 이내의 거리 계산

with kb\_GRID\_04\_3 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_pub\_buld) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set pub\_dist = kb.meter from kb\_GRID\_04\_3 as kb where grid.spo\_cd = kb.spo\_cd;

-- 숙박업소와 500m 이내의 거리 계산

with kb\_GRID\_04\_4 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_hotel) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set hot\_dist = kb.meter from kb\_GRID\_04\_4 as kb where grid.spo\_cd = kb.spo\_cd;

-- select hot\_dist from DATA\_SET\_T limit 100;

-- 다중이용시설과 500m 이내의 거리 계산

with kb\_GRID\_04\_5 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_multiuse) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set mul\_dist = kb.meter from kb\_GRID\_04\_5 as kb where grid.spo\_cd = kb.spo\_cd;

-- select mul\_dist from DATA\_SET\_T limit 100;

-- 버스정류장과 500m 이내의 거리 계산

with kb\_GRID\_04\_6 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_bus) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set bus\_dist = kb.meter from kb\_GRID\_04\_6 as kb where grid.spo\_cd = kb.spo\_cd;

-- select bus\_dist from DATA\_SET\_T limit 100;

-- 음식점과 500m 이내의 거리 계산

with kb\_GRID\_04\_7 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_food) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set food\_dist = kb.meter from kb\_GRID\_04\_7 as kb where grid.spo\_cd = kb.spo\_cd;

-- select food\_dist from DATA\_SET\_T limit 100;

-- 집단습식소와 500m 이내의 거리 계산

with kb\_GRID\_04\_8 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_cafeteria) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set caf\_dist = kb.meter from kb\_GRID\_04\_8 as kb where grid.spo\_cd = kb.spo\_cd;

-- select caf\_dist from DATA\_SET\_T limit 100;

-- 학교와 500m 이내의 거리 계산

-- 학교에 포함되는 격자 데이터를 500m 격자만 추출하여 거리계산 값을 생성하고, 그 의외의 격자는 -1 데이터로 업데이트 한다

with kb\_GRID\_04\_9 as

(

select a.spo\_cd, COALESCE(min(round(cast(ST\_Distance(a.geom, b.geom) as numeric),2)),-1) as meter

from (select spo\_cd, geom from DATA\_SET\_T) a

LEFT JOIN

(select geom from v\_school) b

ON ST\_DWithin(a.geom, b.geom, 500)

group by a.spo\_cd

)

update DATA\_SET\_T as grid set sch\_dist = kb.meter from kb\_GRID\_04\_9 as kb where grid.spo\_cd = kb.spo\_cd;

-- select sch\_dist from DATA\_SET\_T limit 100;

with kb\_GRID\_05\_1 as

(

select spo\_cd,

-- 생산가능인구

CASE when liv\_pp > 0 and liv\_pp <= 50 then '2'

when liv\_pp > 50 and liv\_pp <= 148 then '3'

when liv\_pp > 148 and liv\_pp <= 276 then '4'

when liv\_pp > 457 then '5' else '1'

END AS weight\_liv\_pp,

-- 와이파이 지수(설치지역, 150m반경)

CASE when w\_pnt = '0' then '5' else '1'

END AS weight\_w\_pnt,

CASE when w\_pnt\_150m = '0' then '5' else '1'

END AS weight\_w\_pnt\_150m,

-- 학교가 위치한 격자의 경우 계산에서 제외하기 위해 음수값 입력

CASE when sch\_dist = 0 then '0'

when sch\_dist > 0 then round(((500-sch\_dist)/125+1), 4) else '1'

END AS weight\_sch\_dist,

-- 관광지 지역과의 거리에 반비례한 지수값 입력

CASE when tou\_dist >= 0 then round(((500-tou\_dist)/125+1), 4) else '1'

END AS weight\_tou\_dist,

-- 문화재 지역과의 거리에 반비례한 지수값 입력

CASE when cal\_dist >= 0 then round(((500-cal\_dist)/125+1), 4) else '1'

END AS weight\_cal\_dist,

-- 집합건물과의 거리에 반비례한 지수값 입력

CASE when pub\_dist >= 0 then round(((500-pub\_dist)/125+1), 4) else '1'

END AS weight\_pub\_dist,

-- 숙박업소와의 거리에 반비례한 지수값 입력

CASE when hot\_dist >= 0 then round(((500-hot\_dist)/125+1), 4) else '1'

END AS weight\_hot\_dist,

-- 다중이용시설과의 거리에 반비례한 지수값 입력

CASE when mul\_dist >= 0 then round(((500-mul\_dist)/125+1), 4) else '1'

END AS weight\_mul\_dist,

-- 버스정류장과의 거리에 반비례한 지수값 입력

CASE when bus\_dist >= 0 then round(((500-bus\_dist)/125+1), 4) else '1'

END AS weight\_bus\_dist,

-- 음식점과의 거리에 반비례한 지수값 입력

CASE when food\_dist >= 0 then round(((500-food\_dist)/125+1), 4) else '1'

END AS weight\_food\_dist,

-- 집단급식소와의 거리에 반비례한 지수값 입력

CASE when caf\_dist >= 0 then round(((500-caf\_dist)/125+1), 4) else '1'

END AS weight\_caf\_dist

from DATA\_SET\_T

)

update DATA\_SET\_T as grid

set

cal\_w = weight\_cal\_dist::numeric,

tou\_w = weight\_tou\_dist::numeric,

pub\_w = weight\_pub\_dist::numeric,

hot\_w = weight\_hot\_dist::numeric,

mul\_w = weight\_mul\_dist::numeric,

bus\_w = weight\_bus\_dist::numeric,

food\_w = weight\_food\_dist::numeric,

caf\_w = weight\_caf\_dist::numeric,

sch\_w = weight\_sch\_dist::numeric,

pnt\_w = weight\_w\_pnt::numeric,

pnt\_150m\_w = weight\_w\_pnt\_150m::numeric,

liv\_w = weight\_liv\_pp::numeric

from kb\_GRID\_05\_1 as kb

where grid.spo\_cd = kb.spo\_cd;

update DATA\_SET\_T

set tt\_w =

case

when sch\_w < 0 then ‘0’

when sch\_w >= 0 then round((cal\_w + tou\_w + pub\_w + hot\_w + mul\_w + bus\_w + food\_w + caf\_w + sch\_w + pnt\_w + pnt\_150m\_w +liv\_pp), 4);