```
-------------------------------------
/* AGGREGATED SHAPE SIMILARITY INDEX (ASI)*/
-------------------------------------

/* 1. Transformation into the same coordinate system */
-------------------------------------------------------

    -- creation of the tables tab_a and tab_b (transformed in the same coordinate system)

    -- tables dubravka_inspire, tab_a  - INSPIRE Buildings - Bratislava/Dúbravka (Slovakia)
    -- tables dubravka_osm, tab_b - OSM Buildings - Bratislava/Dúbravka (Slovakia)

    /*EPSG:8353 - S-JTSK [JTSK03] / Krovak East North*/

DROP TABLE IF EXISTS tab_a;

CREATE TABLE tab_a AS
    SELECT id, ST_Transform(geom,8353) AS geom
    FROM dubravka_inspire;

DROP TABLE IF EXISTS tab_b;

CREATE TABLE tab_b AS
    SELECT id, ST_Transform(geom,8353) AS geom
    FROM dubravka_osm;


/* 2. Creation of buffers with a distance of 0 m */
-------------------------------------------------

    -- creation of the table buf_a from tab_a

DROP TABLE IF EXISTS buf_a;

CREATE TABLE buf_a AS
    SELECT (ST_Dump(ST_Buffer(ST_Union(a.geom),0))).geom AS geom
    FROM tab_a a;

ALTER TABLE buf_a ADD COLUMN id_a SERIAL PRIMARY KEY;

    -- creation of the table buf_b from tab_b

DROP TABLE IF EXISTS buf_b;

CREATE TABLE buf_b AS
    SELECT (ST_Dump(ST_Buffer(ST_Union(b.geom),0))).geom AS geom
    FROM tab_b b;

ALTER TABLE buf_b ADD COLUMN id_b SERIAL PRIMARY KEY;


/* 3. Counting polygons in the buffers  */
-----------------------------------------

    -- counting of polygons in each buffer from buf_a -> attribute num_pol

DROP TABLE IF EXISTS num_pol_a;

CREATE TEMPORARY TABLE num_pol_a AS
    SELECT id_a, COUNT(id) AS num_pol
    FROM (
        SELECT
            id,
            id_a,
            ST_Intersection(tab_a.geom,buf_a.geom) AS geom
        FROM tab_a, buf_a
        WHERE tab_a.geom && buf_a.geom AND ST_Within(tab_a.geom,buf_a.geom)
            ) AS int_a_buf_a
    GROUP BY id_a
    ORDER BY id_a;

```

```
CREATE OR REPLACE VIEW v_buf_a AS
SELECT *
FROM buf_a
INNER JOIN num_pol_a USING (id_a);

    -- counting of polygons in each buffer from buf_B -> attribute num_pol

DROP TABLE IF EXISTS num_pol_b;

CREATE TEMPORARY TABLE num_pol_b AS
    SELECT id_b, COUNT(id) AS num_pol
    FROM (
        SELECT
            id,
            id_b,
            ST_Intersection(tab_b.geom,buf_b.geom) AS geom
        FROM tab_b, buf_b
        WHERE tab_b.geom && buf_b.geom AND ST_Within(tab_b.geom,buf_b.geom)
            ) AS int_b_buf_b
    GROUP BY id_b
    ORDER BY id_b;

CREATE OR REPLACE VIEW v_buf_b AS
    SELECT *
    FROM buf_b
    INNER JOIN num_pol_b USING (id_b);


/* 4. Creation of intersections of buf_a and buf_b
and calculation of their parameters (area, perimeter, number of vertices) */
--------------------------------------------------------------------------------

DROP TABLE IF EXISTS int_a_b;

CREATE TEMPORARY TABLE int_a_b AS
    SELECT
        id_a,
        id_b,
        aa.geom AS geom_a,
        bb.geom AS geom_b,
        ST_Intersection(aa.geom,bb.geom) AS geom_int,
        ST_Area(aa.geom) AS area_a,
        ST_Area(bb.geom) AS area_b,
        ST_Perimeter(aa.geom) AS perim_a,
        ST_Perimeter(bb.geom) AS perim_b,
        ST_NPoints(aa.geom) AS num_vert_a,
        ST_NPoints(bb.geom) AS num_vert_b,
        AA.num_pol AS num_pol_a,
        BB.num_pol AS num_pol_b,
        ST_Area(ST_Intersection(aa.geom,bb.geom)) AS area_int
    FROM v_buf_a AS aa, v_buf_b AS bb
    WHERE aa.geom && bb.geom AND ST_Intersects(aa.geom,bb.geom);


/* 5. Calculation of subindices of similarity
-----------------------------------------------

    -- a. sim_tanimoto, sim_dice

    -- b. sim_hausdorff, sim_frechet

    -- c. sim_area, sim_perimeter, sim_vertices, sim_polygons

    -- d. sim_d (distance), sim_s (set), sim_sh (shape)*/

ALTER TABLE int_a_b
    ADD sim_tanimoto numeric GENERATED ALWAYS AS (area_int/(area_a+area_b-area_int))
    STORED,
    ADD sim_dice numeric GENERATED ALWAYS AS (2*area_int/(area_a+area_b)) STORED,
        ADD sim_hausdorff numeric GENERATED ALWAYS AS (
        1- LEAST(1, ST_HausdorffDistance(geom_a, geom_b)/15)) STORED,
    ADD sim_frechet numeric GENERATED ALWAYS AS (
        1-LEAST(1, ST_FrechetDistance(geom_a, geom_b)/15)) STORED,
```

```
    ADD sim_area numeric GENERATED ALWAYS AS (
        GREATEST(1-ABS((area_a-area_b)/GREATEST(area_a,area_b)),0)) STORED,
    ADD sim_perimeter numeric GENERATED ALWAYS AS (
        GREATEST((1-ABS((perim_a-perim_b)/GREATEST(perim_a,perim_b))),0)) STORED,
    ADD sim_vertices numeric GENERATED ALWAYS AS (
        GREATEST(1 - ABS((num_vert_a-num_vert_b)::float/GREATEST(num_vert_a,num_vert_b
        )::float),0)) STORED,
    ADD sim_polygons numeric GENERATED ALWAYS AS (
        LEAST(((num_pol_a)::float/(num_pol_b)::float),((num_pol_b)::float/(num_pol_a
        )::float))) STORED;


DROP TABLE IF EXISTS sim_a_b;

CREATE TABLE sim_a_b AS
    SELECT
        id_a,
        id_b,
        --geom_a,
        --geom_b,
        geom_int,
        sim_dice,
        sim_tanimoto,
        sim_hausdorff,
        sim_frechet,
        sim_area,
        sim_perimeter,
        sim_vertices,
        sim_polygons,
        GREATEST (sim_hausdorff,sim_frechet) AS sim_d,
        sim_tanimoto AS sim_s,
        LEAST(sim_area,sim_perimeter) AS sim_sh
    FROM int_a_b;


/* 6. Calculation of aggreagted similarity indices
-----------------------------------------------------

    -- sim_min - using the MIN aggregation function

    -- sim_max - using the MAX aggregation function

    -- sim_avg - using the AVG aggtegation

    -- sim_agr - using new proposed aggregation (aggregated shape similarity index
    ASI)*/

ALTER TABLE sim_a_b
    ADD sim_min numeric GENERATED ALWAYS AS (LEAST(sim_d,sim_s,sim_sh)) STORED,
    ADD sim_max numeric GENERATED ALWAYS AS (GREATEST(sim_d,sim_s,sim_sh)) STORED,
    ADD sim_avg numeric GENERATED ALWAYS AS ((sim_d+sim_s+sim_sh)/3) STORED,
    ADD sim_agr numeric GENERATED ALWAYS AS (LEAST(GREATEST(sim_d,sim_s),sim_sh))
    STORED;


/* 7. Deciding on category of similarity or change
-----------------------------------------------------

    -- sim_cat: Identical, Moved or rotated, Generalised or slightly changed,
    Different (Different number of objects)*/

DROP TABLE IF EXISTS sim_cat_a_b;

CREATE TABLE sim_cat_a_b AS SELECT DISTINCT id_A, id_B, geom_int as geom, sim_d, sim_s
, sim_sh, sim_agr, sim_vertices,
        CASE WHEN sim_agr > 0.75 THEN 'Identical'
          ELSE
            CASE
                WHEN sim_sh > 0.75 AND sim_vertices = 1 AND sim_s < 0.75 AND sim_d <
                0.75 THEN 'Moved or rotated'
                WHEN sim_agr > 0.5 OR sim_d > 0.75 OR sim_s > 0.75  OR sim_area > 0.75
                 THEN 'Generalised or slightly changed'
                --WHEN sim_polygons < 1 THEN 'Different number of objects'
```

```
                ELSE 'Different'
            END
        END AS sim_cat FROM sim_a_b
        ORDER BY sim_cat;


/* 8. Calculation of the basic statistical characteristics of results*/
-------------------------------------------------------------------------

    -- average values of aggregated similarity indices

SELECT AVG(sim_min) AS min, AVG(sim_max) As max, AVG(sim_avg) AS avg, AVG(sim_agr) AS
agr
FROM sim_a_b;

    -- number of objects in all categories

SELECT sim_cat, COUNT(sim_cat)
FROM sim_cat_a_b
GROUP BY sim_cat;

-------------------------------------------------------------------
--END

```