

Wärmebedarf gebäudescharf (sl_gebaeude_waermebedarf)

Aktualitätsstand: Datenpaket: 08.05.2025 **DWH-Laden: 08.05.2025 13:00:25** Quelle: Fortführungszyklus:

Eigenberechnung des Nutzenergiebedarfs für Wärme nach der folgenden groben Grundformel je Endenergiesektor, GEMOD-Gebäudetyp:

Wärmebedarf = spezifischer Nutzwärmebedarf X Energiefläche X Anzahl Geschosse

Ermittlung von Hilfsgrößen:

- Ermittlung des Endenergiesektors und GEMOD-Gebäudetyps auf Basis der ALKIS-Gebäudefunktion und der Daten vom Zensus 2022
- Ermittlung der Baualtersklassen auf Basis des Bebauungsplans, Flächenplans und der Baualtersklassen vom Zensus 2022
- Ermittlung der Energiefläche auf Basis der Gebäudegrundfläche und eines spezifischen Flächenkorrekturfaktors je GEMOD-Gebäudetyp (vgl. VDI-Norm)
- Ermittlung der Gebäudehöhe aus LoD2 und LoD1
- Ermittlung der Anzahl der Obergeschosse inkl. einer Korrektur für den Dachausbau bei Wohngebäuden
- Plausibilisierung der Beheizbarkeit von Gebäuden durch Betrachtung von Siedlungsflächen, Zensus 2022-Daten, Verteilung von Gebäudegrundfläche/ Gebäudevolumen/ Gebäudehöhe, etc.
- Übernahme der spezifischen Kennwerte aus den benannten Quellen

zusätzliche Quellen:

- Dochev, Ivan. assigning-energetic-types-to-buildings, URL: https://github.com/ivandochev/assigning-energetic-types-to-buildings/blob/master/IWU_VDI_Types_Assigning_asTool_arcgisread_ExtDaten_v3.py
- Ortnier, Sara; Paar, Angelika; Johannsen, Lea; Wachter, Philipp; Hering, Dominik; Pehnt, Martin et al. (2024): Leitfaden Wärmeplanung. Empfehlungen zur methodischen Vorgehensweise für Kommunen und andere Planungsverantwortliche. Hg. v. ifeu - Institut für Energie- und Umweltforschung Heidelberg gGmbH, Öko-Institut e.V., IER Stuttgart, adelphi consult GmbH, Becker Büttner Held PartGmbH, Prognos AG, et al. Online verfügbar unter <https://www.kww-halle.de/praxis-kommunale-waermewende/bundesgesetz-zur-waermeplanung#c636>, zuletzt geprüft am 02.05.2025.

Hinweise auf:

Aktuelle Änderungen:

Nutzungsberechtigte: Intern

Nutzungsbedingungen:

Quellenvermerk: © Landesenergieagentur Sachsen-Anhalt GmbH, URL: <https://www.sachsen-anhalt-energie.de/de/kwp-st.html>

Dateninhaber: Landesenergieagentur Sachsen-Anhalt (LENA)

Quelle: <https://www.sachsen-anhalt-energie.de/de/kwp-st.html>

Dateiformat:

Dateiname:

EPSG:

Ressource-ID: 520

Layer-Bez.: Absoluter (Nutz-) Wärmebedarf Geb. (Gesamtbedarf aller Wohn- und Nichtwohngebäude)

Abfrage DWH-Laden: -- 1. Schema & Tabelle erstellen
CREATE SCHEMA IF NOT EXISTS lena;

DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_hoehe_01;
DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_hoehe_02;
DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_hoehe_03;
DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_hoehe_04;
DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_bauplaene;
DROP MATERIALIZED VIEW IF EXISTS lena.v_mapping_geb_gitter_id_100m;

DROP TABLE IF EXISTS lena.sl_gebaeude_waermebedarf;

CREATE MATERIALIZED VIEW lena.v_mapping_geb_bauplaene AS
WITH jahr_quelle AS (
SELECT
alk.gid,
bp.gid AS gid_bp,
fnp.gid AS gid_fnp,
CASE
WHEN bp.status = 'RK' THEN EXTRACT(YEAR FROM bp.rk_dat)
WHEN bp.status = 'GN' THEN EXTRACT(YEAR FROM bp.gen_dat)
ELSE NULL

```

END AS jahr_bp,
ROW_NUMBER() OVER (PARTITION BY alk.gid ORDER BY
CASE
    WHEN bp.status = 'RK' THEN EXTRACT(YEAR FROM bp.rk_dat)
    WHEN bp.status = 'GN' THEN EXTRACT(YEAR FROM bp.gen_dat)
    ELSE NULL
END
) AS row_num_bp,

CASE
    WHEN fnp.status = 'RK' THEN EXTRACT(YEAR FROM fnp.rk_dat)
    WHEN fnp.status = 'GN' THEN EXTRACT(YEAR FROM fnp.gen_dat)
    ELSE NULL
END AS jahr_fnp,
ROW_NUMBER() OVER (PARTITION BY alk.gid ORDER BY
CASE
    WHEN fnp.status = 'RK' THEN EXTRACT(YEAR FROM fnp.rk_dat)
    WHEN fnp.status = 'GN' THEN EXTRACT(YEAR FROM fnp.gen_dat)
    ELSE NULL
END
) AS row_num_fnp
FROM geobasisdaten.gbis_gebaeude alk
LEFT JOIN raumordnung.tab_42001g bp ON ST_Within(alk.the_geom, bp.the_geom)
LEFT JOIN raumordnung.tab_41001g fnp ON ST_Within(alk.the_geom, fnp.the_geom)
)
SELECT
gid,
gid_bp,
gid_fnp,
CASE
    WHEN jahr < 1860 THEN 'A'
    WHEN jahr < 1919 THEN 'B'
    WHEN jahr < 1949 THEN 'C'
    WHEN jahr < 1958 THEN 'D'
    WHEN jahr < 1969 THEN 'E'
    WHEN jahr < 1979 THEN 'F'
    WHEN jahr < 1984 THEN 'G'
    WHEN jahr < 1995 THEN 'H'
    WHEN jahr < 2002 THEN 'I'
    WHEN jahr < 2012 THEN 'J'
    WHEN jahr < 2021 THEN 'K'
    WHEN jahr >= 2021 THEN 'L'
    ELSE NULL
END AS bak_ezfh,

CASE
    WHEN jahr < 1860 THEN 'A'
    WHEN jahr < 1919 THEN 'B'
    WHEN jahr < 1949 THEN 'C'
    WHEN jahr < 1958 THEN 'D'
    WHEN jahr < 1969 THEN 'E'
    WHEN jahr < 1979 THEN 'F'
    WHEN jahr < 1984 THEN 'G'
    WHEN jahr < 1995 THEN 'H'
    WHEN jahr < 2002 THEN 'I'
    WHEN jahr < 2010 THEN 'J2'
    WHEN jahr < 2021 THEN 'K2'
    WHEN jahr >= 2021 THEN 'L'
    ELSE NULL
END AS bak_mfh,

CASE
    WHEN jahr < 1979 THEN '1'
    WHEN jahr < 2010 THEN '2'
    WHEN jahr >= 2010 THEN '3'
    ELSE NULL
END AS bak_nwg

FROM (
    SELECT *, COALESCE(jahr_bp, jahr_fnp) AS jahr
    FROM jahr_quelle
    WHERE row_num_bp <= 1 AND row_num_fnp <= 1
) AS f
;

CREATE MATERIALIZED VIEW lena.v_mapping_geb_gitter_id_100m AS
WITH gitter AS (
    SELECT gitter_id_100m, geom

```

```

FROM (
  SELECT gitter_id_100m, geom FROM zensus2022.sl_gebaeude_baujahr_100m
  UNION
  SELECT gitter_id_100m, geom FROM zensus2022.sl_gebaeude_gebaeudetyp_100m
  UNION
  SELECT gitter_id_100m, geom FROM zensus2022.energetraeger_geb100m
  UNION
  SELECT gitter_id_100m, geom FROM zensus2022.heizungsart_geb100m
) a
GROUP BY gitter_id_100m, geom
),
gebaeude_mit_gitter AS (
  SELECT
    geb.gid,
    geb.gml_id,
    gitter.gitter_id_100m,
    ST_Centroid(geb.the_geom) AS geb_centroid,
    geb.the_geom
  FROM geobasisdaten.gbis_gebaeude geb
  JOIN gitter ON ST_Contains(gitter.geom, geb.the_geom)
),
top_gebaeudetyp AS (
  SELECT
    t2.gitter_id_100m,
    t2.gebaeudetyp,
    100 * t2.anz / ges.anz_insg AS gebaeudetyp_anteil
  FROM (
    SELECT
      gitter_id_100m,
      gebaeudetyp,
      anz,
      ROW_NUMBER() OVER (PARTITION BY gitter_id_100m ORDER BY anz DESC) AS row_num
    FROM (
      SELECT gitter_id_100m, 'EFH' AS gebaeudetyp, freiefh + efh_dhh + freist_zfh + zfh_dhh AS anz
      FROM zensus2022.sl_gebaeude_gebaeudetyp_100m
      UNION ALL
      SELECT gitter_id_100m, 'RH', efh_reihenhaus + zfh_reihenhaus FROM
      zensus2022.sl_gebaeude_gebaeudetyp_100m
      UNION ALL
      SELECT gitter_id_100m, 'MFH', mfh_3bis6wohnungen + mfh_7bis12wohnungen FROM
      zensus2022.sl_gebaeude_gebaeudetyp_100m
      UNION ALL
      SELECT gitter_id_100m, 'GMH', mfh_13undmehrwohnungen FROM
      zensus2022.sl_gebaeude_gebaeudetyp_100m
      UNION ALL
      SELECT gitter_id_100m, 'SONST', anderergebaeudetyp FROM
      zensus2022.sl_gebaeude_gebaeudetyp_100m
    ) t
  ) t2
  JOIN (
    SELECT gitter_id_100m,
      freiefh + efh_dhh + efh_reihenhaus + freist_zfh + zfh_dhh + zfh_reihenhaus +
      mfh_3bis6wohnungen + mfh_7bis12wohnungen + mfh_13undmehrwohnungen + anderergebaeudetyp
    AS anz_insg
    FROM zensus2022.sl_gebaeude_gebaeudetyp_100m
  ) ges ON t2.gitter_id_100m = ges.gitter_id_100m
  WHERE t2.row_num = 1 AND t2.anz > 0
),
top_baujahr AS (
  SELECT
    t2.gitter_id_100m,
    t2.geb_baujahr,
    100 * t2.anz / ges.anz_insg AS geb_baujahr_anteil
  FROM (
    SELECT
      gitter_id_100m,
      geb_baujahr,
      anz,
      ROW_NUMBER() OVER (PARTITION BY gitter_id_100m ORDER BY anz DESC) AS row_num
    FROM (
      SELECT gitter_id_100m, 'vor1919', vor1919 FROM zensus2022.sl_gebaeude_baujahr_100m
      UNION ALL SELECT gitter_id_100m, 'a1919bis1948', a1919bis1948 FROM
      zensus2022.sl_gebaeude_baujahr_100m
      UNION ALL SELECT gitter_id_100m, 'a1949bis1978', a1949bis1978 FROM
      zensus2022.sl_gebaeude_baujahr_100m
      UNION ALL SELECT gitter_id_100m, 'a1979bis1990', a1979bis1990 FROM
      zensus2022.sl_gebaeude_baujahr_100m
      UNION ALL SELECT gitter_id_100m, 'a1991bis2000', a1991bis2000 FROM

```

```

zensus2022.sl_gebaeude_baujahr_100m
        UNION ALL SELECT gitter_id_100m, 'a2001bis2010', a2001bis2010 FROM
zensus2022.sl_gebaeude_baujahr_100m
        UNION ALL SELECT gitter_id_100m, 'a2011bis2019', a2011bis2019 FROM
zensus2022.sl_gebaeude_baujahr_100m
        UNION ALL SELECT gitter_id_100m, 'a2020undspaeter', a2020undspaeter FROM
zensus2022.sl_gebaeude_baujahr_100m
    ) t2( gitter_id_100m, geb_baujahr, anz )
    ) t2
JOIN (
    SELECT gitter_id_100m,
        vor1919 + a1919bis1948 + a1949bis1978 + a1979bis1990 + a1991bis2000 +
        a2001bis2010 + a2011bis2019 + a2020undspaeter AS anz_insg
    FROM zensus2022.sl_gebaeude_baujahr_100m
    ) ges ON t2.gitter_id_100m = ges.gitter_id_100m
WHERE t2.row_num = 1 AND t2.anz > 0
)
SELECT
    gmg.gid,
    gmg.gml_id,
    gmg.gitter_id_100m,
    gt.gebaeudetyp AS gebaeudetyp_z,
    gt.gebaeudetyp_anteil AS gebaeudetyp_z_p,
    bak.geb_baujahr AS bak_z,
    bak.geb_baujahr_anteil AS bak_z_p,
    gmg.the_geom
FROM gebaeude_mit_gitter gmg
LEFT JOIN top_gebaeudetyp gt ON gmg.gitter_id_100m = gt.gitter_id_100m
LEFT JOIN top_baujahr bak ON gmg.gitter_id_100m = bak.gitter_id_100m
;

```

```

CREATE TABLE lena.sl_gebaeude_waermebedarf(

```

```

    gid integer PRIMARY KEY,
    gml_id varchar,
    waerme_kwh double precision,
    sektor varchar,
    g_gemod varchar,
    g_gemod_tx varchar,
    code_g varchar,
    code_g_txt varchar,
    code_f varchar,
    code_f_txt varchar,
    gflache_m2 double precision,
    eflache_m2 double precision,
    geb_hoehe double precision,
    g_hoehe_2 double precision,
    g_hoehe_1 double precision,
    g_volumen double precision,
    gsshoehe double precision,
    anz_og double precision,
    fk_dg_ausb integer,
    fk_wg_nwg integer,
    beheizt integer,
    rellage varchar,
    dachform varchar,
    geb_typ_z varchar,
    geb_typ_p integer,
    bak_ezfh varchar,
    bak_mfh varchar,
    bak_nwg varchar,
    bak_z varchar,
    bak_z_p integer,
    gmdschl integer,
    strssname varchar,
    gid_tn integer,
    gid_str integer,
    objid_str varchar,
    gitter_id varchar,
    gid_bp varchar,
    gid_fnp varchar,
    the_geom geometry(MultiPolygon,25832)
);

```

```

INSERT INTO lena.sl_gebaeude_waermebedarf (
    gid, gml_id, waerme_kwh, sektor, g_gemod, g_gemod_tx, code_g, code_g_txt, code_f,
    code_f_txt, gflache_m2, eflache_m2, geb_hoehe, g_hoehe_2, g_hoehe_1, g_volumen, gsshoehe,
    anz_og, fk_dg_ausb, fk_wg_nwg, beheizt, rellage, dachform, geb_typ_z, geb_typ_p, bak_ezfh, bak_mfh,
    bak_nwg, bak_z, bak_z_p, gmdschl, strssname, gid_tn, gid_str, objid_str, gitter_id, gid_bp, gid_fnp,

```

```

the_geom
)
SELECT
  g.gid,
  g.gml_id,
  null as waerme_kwh,
  r.endenergiesektor as sektor,
  r.gebtyp_gemod as g_gemod,
  r.gebtyp_gemod_txt as g_gemod_tx,
  g.gfkzshh as code_g,
  g.funktion as code_g_txt,
  ntz.gfkzshh as code_f,
  ntz.gfkzshh_txt as code_f_txt,
  round(ST_Area(g.the_geom)) as gflache_m2,
  ST_Area(g.the_geom) * r.flaechenkorrektur_faktor / 100 as eflache_m2,
  coalesce(l2.measured_height,l1.measured_height) as geb_hoehe,
  l2.measured_height as g_hoehe_2,
  l1.measured_height as g_hoehe_1,
  null as g_volumen,
  r.geschosshoehe as gsshoehe,
  null as anz_og,
  case when l2.rooftype in ('3100','3200','3300','3400') then 75 else 0 end as fk_dg_ausb,
  100 as fk_wg_nwg,
  null as beheizt,
  g.rellage,
  l2.rooftype as dachform,
  z.gebaeudetyp_z as geb_typ_z,
  z.gebaeudetyp_z_p as geb_typ_p,
  p.bak_ezfh,
  p.bak_mfh,
  p.bak_nwg,
  z.bak_z,
  z.bak_z_p,
  g.gmdschl,
  g.strassenname as strssname,
  ntz.gid as gid_tn,
  s.gid_str,
  s.objid_str,
  z.gitter_id_100m as gitter_id,
  p.gid_bp,
  p.gid_fnp,
  g.the_geom
FROM geobasisdaten.gbise_gbaeude g
LEFT JOIN (select gml_id, rooftype, measured_height from geobasisdaten.gebaeude_lod2) l2 on
g.gml_id=l2.gml_id /* LoD2: Gebäudehöhe, Dachform */
LEFT JOIN (select gml_id, measured_height from geobasisdaten.gebaeude_lod1) l1 on g.gml_id=l1.gml_id
/* LoD1: Gebäudehöhe */
LEFT JOIN lena.v_mapping_gebaeude_bauplaene p ON g.gid=p.gid /* Baujahresklassen aus B-Plan und
FNP */
LEFT JOIN geobasisdaten.nutzungsgebiete_dlm ntz on ST_Within(g.the_geom, ntz.geom) /*
Siedlungsblock */
LEFT JOIN lena.mapping_gebaeude_strassenabschnitt as s on g.gid=s.gid /* Straßenabschnitt */
LEFT JOIN lena.v_mapping_geb_gitter_id_100m z on g.gid=z.gid /* Zensus 2022*/
LEFT JOIN lena.ref_gebfkt_gebtyp_gemod r on r.gfkzshh=g.gfkzshh /* Referenzwerte */
;

-- Berechnung: beheizt
UPDATE lena.sl_gebaeude_waermebedarf
SET
beheizt = case
  when gflache_m2 < 50 then 0
  when sektor = 'WG' and code_f in
('41002_1790','41002_2520','41002_2530','41002_2540','41002_2570','41002_2610','41002_2620','41003'
,'41005','41008_4100','41008_4200','41008_4230','41008_4320','41008_4330','41008_4330','41008_4400',
'41008_4420','41008_4440','41009') then 0
  when sektor = 'WG' and code_f = '41008_4310' then 39
  when geb_hoehe < 2.8 then 0
  when gflache_m2 * geb_hoehe < 90 then 0
  when code_g_txt in ('Keller','Tiefgarage') then 0
  when sektor='kein_bedarf' then 0
  else 100 end
;

-- Imputation: Gebäudehöhe
CREATE MATERIALIZED VIEW lena.v_mapping_geb_hoehe_01 AS
SELECT
  sektor, g_gemod, geb_typ_z, gid_tn,
  PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY geb_hoehe) AS geb_hoehe_median

```

```

FROM lena.sl_gebaeude_waermebedarf
WHERE geb_hoehe is not null and beheizt > 0
GROUP BY
    sektor, g_gemod, geb_typ_z, gid_tn
;
CREATE MATERIALIZED VIEW lena.v_mapping_geb_hoehe_02 AS
SELECT
    sektor, g_gemod, geb_typ_z, code_f,
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY geb_hoehe) AS geb_hoehe_median
FROM lena.sl_gebaeude_waermebedarf
WHERE geb_hoehe is not null and beheizt > 0
GROUP BY
    sektor, g_gemod, geb_typ_z, code_f
;
CREATE MATERIALIZED VIEW lena.v_mapping_geb_hoehe_03 AS
SELECT
    sektor, g_gemod, geb_typ_z,
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY geb_hoehe) AS geb_hoehe_median
FROM lena.sl_gebaeude_waermebedarf
WHERE geb_hoehe is not null and beheizt > 0
GROUP BY
    sektor, g_gemod, geb_typ_z
;
CREATE MATERIALIZED VIEW lena.v_mapping_geb_hoehe_04 AS
SELECT
    sektor,
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY geb_hoehe) AS geb_hoehe_median
FROM lena.sl_gebaeude_waermebedarf
WHERE geb_hoehe is not null and beheizt > 0
GROUP BY
    sektor
;

```

```

UPDATE lena.sl_gebaeude_waermebedarf f
SET geb_hoehe = med.geb_hoehe_median
FROM lena.v_mapping_geb_hoehe_01 med
WHERE f.geb_hoehe is null
    and f.sektor = med.sektor
    and f.g_gemod = med.g_gemod
    and f.geb_typ_z = med.geb_typ_z
    and f.gid_tn = med.gid_tn
;

```

```

UPDATE lena.sl_gebaeude_waermebedarf f
SET geb_hoehe = med.geb_hoehe_median
FROM lena.v_mapping_geb_hoehe_02 med
WHERE f.geb_hoehe is null
    and f.sektor = med.sektor
    and f.g_gemod = med.g_gemod
    and f.geb_typ_z = med.geb_typ_z
    and f.code_f = med.code_f
;

```

```

UPDATE lena.sl_gebaeude_waermebedarf f
SET geb_hoehe = med.geb_hoehe_median
FROM lena.v_mapping_geb_hoehe_03 med
WHERE f.geb_hoehe is null
    and f.sektor = med.sektor
    and f.g_gemod = med.g_gemod
    and f.geb_typ_z = med.geb_typ_z
;

```

```

UPDATE lena.sl_gebaeude_waermebedarf f
SET geb_hoehe = med.geb_hoehe_median
FROM lena.v_mapping_geb_hoehe_03 med
WHERE f.geb_hoehe is null
    and f.sektor = med.sektor
;

```

```

-- Berechnung: Anzahl Obergeschosse, Gebäudevolumen
UPDATE lena.sl_gebaeude_waermebedarf
SET
    anz_og = round(10*geb_hoehe / gsshoehe)/10,
    g_volumen = round(case when geb_hoehe > 0 then gflache_m2 * geb_hoehe else null end)
;

```

```

-- Imputation: Baualtersklasse
UPDATE lena.sl_gebaeude_waermebedarf f
SET
    bak_nwg = CASE
        WHEN f.bak_z in ('vor1919','a1919bis1948','a1949bis1978') THEN 1
    END
;

```

```

    WHEN f.bak_z in ('a1979bis1990','a1991bis2000','a2001bis2010') THEN 2
    WHEN f.bak_z in ('a2011bis2019','a2020undspaeter') THEN 3
    WHEN f.bak_z is null THEN 0
    ELSE null END
WHERE f.bak_nwg is null
;
UPDATE lena.sl_gebaeude_waermebedarf f
SET bak_ezfh = CASE
    --WHEN f.bak_z in (") THEN 'A'
    WHEN f.bak_z in ('vor1919') THEN 'B'
    WHEN f.bak_z in ('a1919bis1948') THEN 'C'
    WHEN f.bak_z in ('a1949bis1978') THEN 'D'
    --WHEN f.bak_z in (") THEN 'E'
    --WHEN f.bak_z in (") THEN 'F'
    WHEN f.bak_z in ('a1979bis1990') THEN 'G'
    --WHEN f.bak_z in (") THEN 'H'
    WHEN f.bak_z in ('a1991bis2000') THEN 'I'
    WHEN f.bak_z in ('a2001bis2010') THEN 'J'
    WHEN f.bak_z in ('a2011bis2019') THEN 'K'
    WHEN f.bak_z in ('a2020undspaeter') THEN 'L'
    ELSE 'Z'
    END
WHERE f.bak_ezfh is null
;
UPDATE lena.sl_gebaeude_waermebedarf f
SET bak_mfh = CASE
    --WHEN f.bak_z in (") THEN 'A'
    WHEN f.bak_z in ('vor1919') THEN 'B'
    WHEN f.bak_z in ('a1919bis1948') THEN 'C'
    WHEN f.bak_z in ('a1949bis1978') THEN 'D'
    --WHEN f.bak_z in (") THEN 'E'
    --WHEN f.bak_z in (") THEN 'F'
    WHEN f.bak_z in ('a1979bis1990') THEN 'G'
    --WHEN f.bak_z in (") THEN 'H'
    WHEN f.bak_z in ('a1991bis2000') THEN 'I'
    WHEN f.bak_z in ('a2001bis2010') THEN 'J2'
    WHEN f.bak_z in ('a2011bis2019') THEN 'K2'
    WHEN f.bak_z in ('a2020undspaeter') THEN 'L'
    ELSE 'Z'
    END
WHERE f.bak_mfh is null
;

-- Berechnung: waerme_kwh
UPDATE lena.sl_gebaeude_waermebedarf f
SET waerme_kwh = f.eflache_m2 * f.beheizt * 0.01 *
    CASE
        WHEN f.sektor = 'kein_bedarf' THEN 0
        WHEN f.sektor is null and f.code_g_txt in ('Tiefgarage','Keller') THEN 0 /* führt er nicht aus */
        WHEN f.sektor = 'NWG' THEN f.anz_og * kz.q_d_spez_kwh_m2_ebf_a
            WHEN f.sektor = 'WG' THEN (f.anz_og + COALESCE(f.fk_dg_ausb * 0.01, 0)) *
kz.q_d_spez_kwh_m2_ebf_a
        ELSE NULL
    END
FROM lena.ref_gebfkt_nutzenergiebedarf kz
WHERE
    f.sektor = 'kein_bedarf' OR
    f.sektor is null OR
    (f.sektor = kz.endenergiesektor
    AND ((f.sektor = 'NWG' AND f.g_gemod = kz.gebtyp_gemod AND f.bak_nwg = kz.bak)
    OR (f.sektor = 'WG'
    AND (kz.gebtyp_gemod_txt = case when f.geb_typ_z='SONST' or f.geb_typ_z is null then 'Gesamt'
else f.geb_typ_z end)
    AND (((f.geb_typ_z IN ('EFH','RH','SONST') or f.geb_typ_z is null) AND f.bak_ezfh = kz.bak)
    OR (f.geb_typ_z IN ('MFH','GMH') AND f.bak_mfh = kz.bak))
    )
    )
    )
;
UPDATE lena.sl_gebaeude_waermebedarf f
SET waerme_kwh = 0
WHERE f.sektor is null and f.code_g_txt in ('Tiefgarage','Keller')
;

-- Berechnung: fk_wg_nwg
UPDATE lena.sl_gebaeude_waermebedarf
SET fk_wg_nwg =
CASE

```

```

        WHEN beheizt = 0 THEN 0
        WHEN anz_og > 1 THEN ((anz_og -1)/anz_og) * 100
        ELSE 100
    END
WHERE sektor = 'gemischt'
;

```

```

-- Berechnung waerme_kwh (gemischt genutzte Wohngebäude)

```

```

WITH
    kw_wg AS (select gebtyp_gemod_txt, bak, q_d_spez_kwh_m2_ebf_a from
lena.ref_gebfkt_nutzenergiebedarf where endenergiesektor = 'WG'),
    kw_nwg AS (select gebtyp_gemod_txt, bak, q_d_spez_kwh_m2_ebf_a from
lena.ref_gebfkt_nutzenergiebedarf where endenergiesektor = 'NWG' and gebtyp_gemod_txt = 'Gesamt')
UPDATE lena.sl_gebaeude_waermebedarf f
SET waerme_kwh = f.eflache_m2 * f.beheizt * 0.01
    * (kw_wg.q_d_spez_kwh_m2_ebf_a * (f.fk_wg_nwg * 0.01
    * (f.anz_og +
COALESCE(f.fk_dg_ausb * 0.01, 0)))
    +(kw_nwg.q_d_spez_kwh_m2_ebf_a * (100 - f.fk_wg_nwg) * 0.01 * f.anz_og))
FROM kw_wg, kw_nwg
WHERE
    f.sektor = 'gemischt'
    and kw_nwg.bak = f.bak_nwg
    and kw_wg.gebtyp_gemod_txt = case when (f.geb_typ_z is null) or (f.geb_typ_z = 'SONST') then
'Gesamt' else f.geb_typ_z end
    and ((kw_wg.gebtyp_gemod_txt in ('EFH','RH','Gesamt') and kw_wg.bak = f.bak_ezfh)
    or (kw_wg.gebtyp_gemod_txt in ('MFH','GMH') and kw_wg.bak = f.bak_mfh))
;

```

```

-- Index anlegen

```

```

CREATE INDEX IF NOT EXISTS idx_gid ON lena.sl_gebaeude_waermebedarf(gid);

```

```

-- Statistik aktualisieren

```

```

ANALYZE lena.sl_gebaeude_waermebedarf;

```

Abfrage Datenpaket:

```

SELECT gid, gml_id, waerme_kwh, sektor, g_gemod, g_gemod_tx, code_g, code_g_txt, code_f,
code_f_txt, gflache_m2, eflache_m2, geb_hoehe, g_hoehe_2, g_hoehe_1, g_volumen, gsshoehe, anz_og,
fk_dg_ausb, fk_wg_nwg, beheizt, rellage, dachform, geb_typ_z, geb_typ_p, bak_ezfh, bak_mfh, bak_nwg,
bak_z, bak_z_p, gmdschl, strssname, gid_tn, gid_str, objid_str, gitter_id, gid_bp, gid_fnp, the_geom FROM
lena.sl_gebaeude_waermebedarf

```

Abfrage Karte:

```

the_geom from (SELECT gid, beheizt, coalesce(waerme_kwh,-1) as ne_waerme_kwh, the_geom FROM
lena.sl_gebaeude_waermebedarf ) AS foo using unique gid using srid=25832

```