
```
-- Welche Daten stehen wie zur Verfügung?

-- Straßenbaumkataster
SELECT * FROM hamburg.strassenbaumkataster_original LIMIT 100;
-- Die Tabelle entstand aus einer WFS-Abfrage in QGIS.
-- https://geodienste.hamburg.de/HH\_WFS\_Strassenbaumkataster
-- Die Daten wurden anschließend in eine PostgreSQL-DB übertragen.

-- Ortsteile, Stadtteile, Bezirke
SELECT * FROM hamburg.ortsteile LIMIT 100;
SELECT * FROM hamburg.stadtteile LIMIT 100;
SELECT * FROM hamburg.bezirke LIMIT 100;

/*
Was stellen wir fest bei der Betrachtung der Tabelle
'strassenbaumkataster_original'?
1. Der PK 'id' ist eine fortlaufende Nummer mit 1 beginnend.
   Wahrscheinlich künstlich erzeugt beim Umgang der Daten von QGIS zu
   PostgreSQL.
2. Könnte 'baumid' mglw. als PK einsetzbar sein?
3. 'gattung', 'art' und 'sorte' sind voneinander anhängig und redundant.
   'sorte_latein' sollte eindeutig sein. Alles andere hängt davon ab.
4. Unterscheiden sich 'pflanzjahr' und 'pflanzjahr_portal'?
5. 'kronendurchmesser_z' = 'kronendurchmesser' + m
   'stammumfang_z' = 'stammumfang' + cm
6. 'strasse' und 'hausnummer' stimmen nicht zwingend mit der Geometrie überein.
   Kann z.B. auch mal gegenüber sein.
7. 'ortsteil_nr', 'stadtteil' und 'bezirk' sind voneinander abhängig.
*/

-- Festlegen welche Attribute tatsächlich gebraucht und
-- in eine neue Tabelle übertragen werden müssen.
SELECT * FROM hamburg.strassenbaumkataster_original LIMIT 100;

-- zu 2. Könnte 'baumid' mglw. als PK einsetzbar sein?
SELECT COUNT(*) FROM hamburg.strassenbaumkataster_original;

SELECT COUNT(DISTINCT baumid) FROM hamburg.strassenbaumkataster_original;

-- zu 4. Unterscheiden sich 'pflanzjahr' und 'pflanzjahr_portal'?
SELECT * FROM hamburg.strassenbaumkataster_original
WHERE pflanzjahr != pflanzjahr_portal;

DROP TABLE IF EXISTS hamburg.strassenbaumkataster;

-- Einfacher als mit einer Unterabfrage.
-- Die neue Tabelle wird automatisch erzeugt.
SELECT
    baumid,
    baumnummer,
    sorte_latein, -- wird später durch sorte_nr ersetzt
    pflanzjahr,
    kronendurchmesser,
    stammumfang,
    strasse,
    hausnummer,
    ortsteil_nr,
    geom
INTO hamburg.strassenbaumkataster
FROM hamburg.strassenbaumkataster_original;

-- 'baumid' wird zum PK
ALTER TABLE hamburg.strassenbaumkataster
ADD CONSTRAINT baumpk PRIMARY KEY (baumid);
```

```
SELECT * FROM hamburg.strassenbaumkataster LIMIT 100;

-- zu 3.
-- Wie viele und welche Datensätze würde eine neue Tabelle beinhalten?
SELECT DISTINCT sorte_latein, sorte_deutsch,
                art_latein, art_deutsch,
                gattung_latein, gattung_deutsch
FROM hamburg.strassenbaumkataster_original;

DROP TABLE IF EXISTS hamburg.sorte;

-- Tabelle 'sorte' erzeugen
CREATE TABLE hamburg.sorte (
    id SERIAL PRIMARY KEY,
    sorte_latein VARCHAR,
    sorte_deutsch VARCHAR,
    art_latein VARCHAR,
    art_deutsch VARCHAR,
    gattung_latein VARCHAR,
    gattung_deutsch VARCHAR
);

-- Tabelle 'sorte' mit einer Unterabfrage befüllen.
INSERT INTO hamburg.sorte (sorte_latein, sorte_deutsch,
                          art_latein, art_deutsch,
                          gattung_latein, gattung_deutsch)
SELECT DISTINCT sorte_latein, sorte_deutsch,
                art_latein, art_deutsch,
                gattung_latein, gattung_deutsch
FROM hamburg.strassenbaumkataster_original
WHERE sorte_latein IS NOT null;

SELECT * FROM hamburg.sorte;

-- Wie vielen Bäume wurde keine Sorte zugeordnet?
SELECT COUNT(*) FROM hamburg.strassenbaumkataster
WHERE sorte_latein IS NULL;

-- In 'strassenbaumkataster' wird 'sorte_latein' mit der
-- 'id' (PK) aus 'sorte' ersetzt.
WITH subquery AS (
    SELECT id, sorte_latein FROM hamburg.sorte
)
UPDATE hamburg.strassenbaumkataster
SET sorte_latein = subquery.id::text
FROM subquery
WHERE hamburg.strassenbaumkataster.sorte_latein = subquery.sorte_latein;

-- Die Spalte 'sorte_latein' wird in 'sorte_nr' geändert.
ALTER TABLE hamburg.strassenbaumkataster
    RENAME COLUMN sorte_latein TO sorte_nr;

-- Der Datentyp von 'sorte_nr' wird in INTEGER geändert.
ALTER TABLE hamburg.strassenbaumkataster
    ALTER COLUMN sorte_nr TYPE INTEGER USING sorte_nr::integer;

-- Überprüfen, ob die Anzahl mit der obigen übereinstimmt.
SELECT COUNT(*) FROM hamburg.strassenbaumkataster
WHERE sorte_nr IS NULL;

-- Überprüfen, ob in 'sorte_nr' die richtigen IDs eingetragen wurden.
SELECT ka.baumid, so.sorte_deutsch, so.id, ka.sorte_nr
FROM hamburg.strassenbaumkataster ka, hamburg.sorte so
WHERE so.id = ka.sorte_nr
LIMIT 100;
```

```
-- Informationen zu den Tabellen einer DB abfragen.
-- db_gp_2022/Catalogs/PostgreSQL Catalog/Views/pg_tables
SELECT * FROM pg_catalog.pg_tables
    WHERE schemaname != 'pg_catalog'
    AND schemaname != 'information_schema'
    AND schemaname = 'grp0'
    ORDER BY schemaname, tablename;

-- Informationen zu den Attributen der Tabelle abfragen.
SELECT * FROM information_schema.columns
    WHERE table_schema='grp0' AND table_name='strassenbaumkataster';

-- Verfügbare Erweiterungen
SELECT * FROM pg_catalog.pg_available_extensions;

-- Weitere nützliche Statements
-- https://www.postgresql.org/docs/current/information-schema.html
SELECT * FROM information_schema.schemata;
SELECT * FROM information_schema.tables;
SELECT * FROM information_schema.key_column_usage;

SELECT * FROM information_schema.views;
SELECT * FROM information_schema.triggers;
SELECT * FROM information_schema.routines where routine_type='FUNCTION';

SELECT * FROM pg_catalog.pg_indexes;

-- PostGIS
SELECT UpdateGeometrySRID('grp0', 'stadtteile', 'geom', 25832);
SELECT public.ST_AsText(public.ST_Transform(geom, 3857)) FROM bezirke;
```