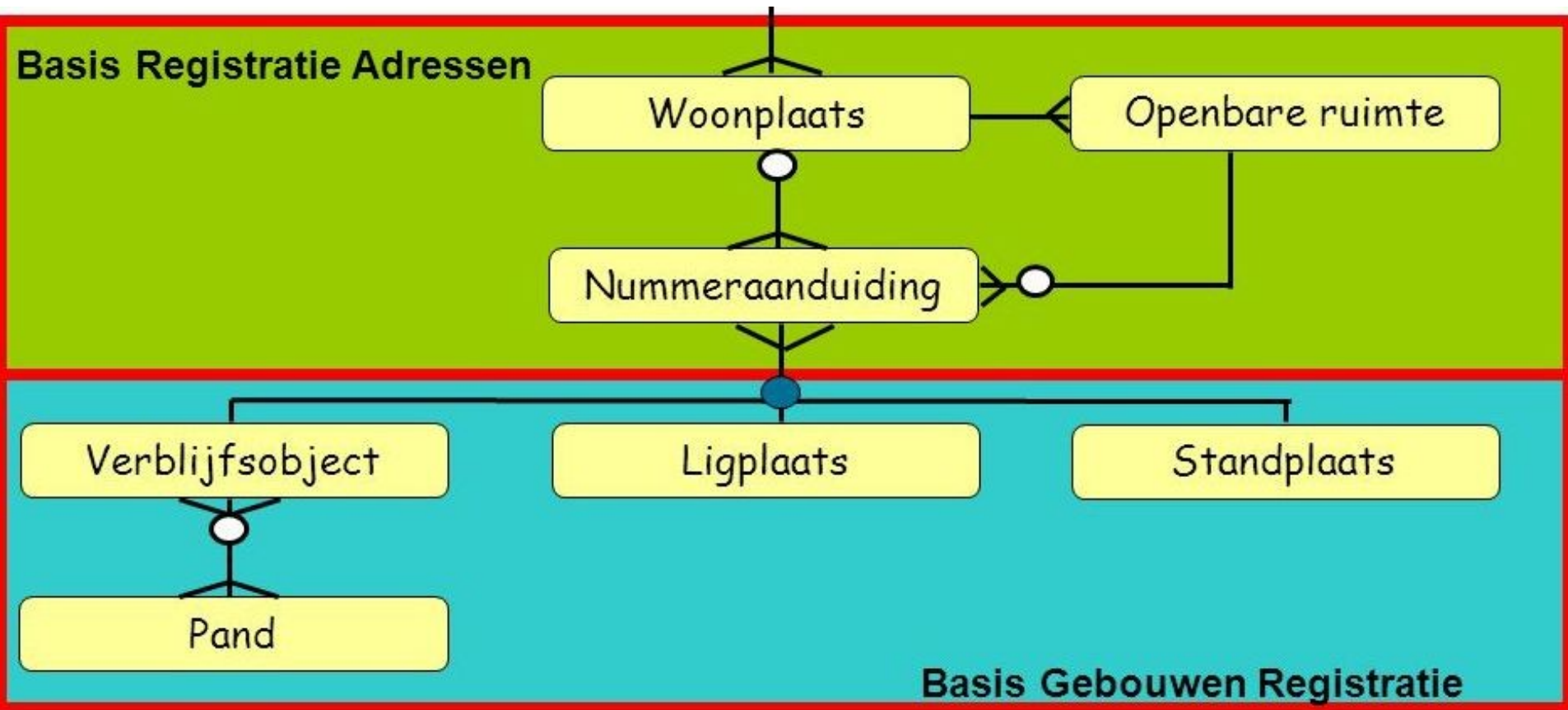


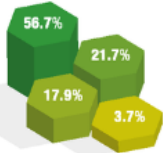
# BAG van complex datamodel tot eindgebruikersdata



ATOS competence  
03 december 2015  
Ronnie Lassche

# 2014 - 2015 GEODIENST

Ruimtelijke data, tools en expertise



Staff Student PhD Overig

Gebruikers Geodienst

25.000

## Datasets

Meer dan 25.00 datasets beschikbaar in het geoportal van de RUG voor onderzoekers en studenten.

Internationale appearances

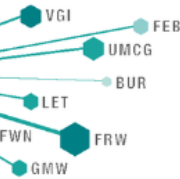


## Movement tracking

Movement tracking voor Sport & Technologie bij de Hanze.

## Centrale Geo-afdeling

De enige universiteit met centrale geo-afdeling.  
rug.nl/geo



## Faculteiten

De meeste faculteiten maken actief gebruik van de Geodienst.



## Het team

Het team bestaat uit 7 man, waarvan 4 studenten.

1

We bestaan 1 jaar!



## Wetenschap versnellen

We hebben een bijdrage geleverd aan onderzoek van meer dan 100 wetenschappers.



## Projecten

Samenwerking met LifeLines voor ruimtelijke analyses.  
Real-time sensor map prototype voor Incas3.



**Merkator**  
GeoSpatial IT-Solutions

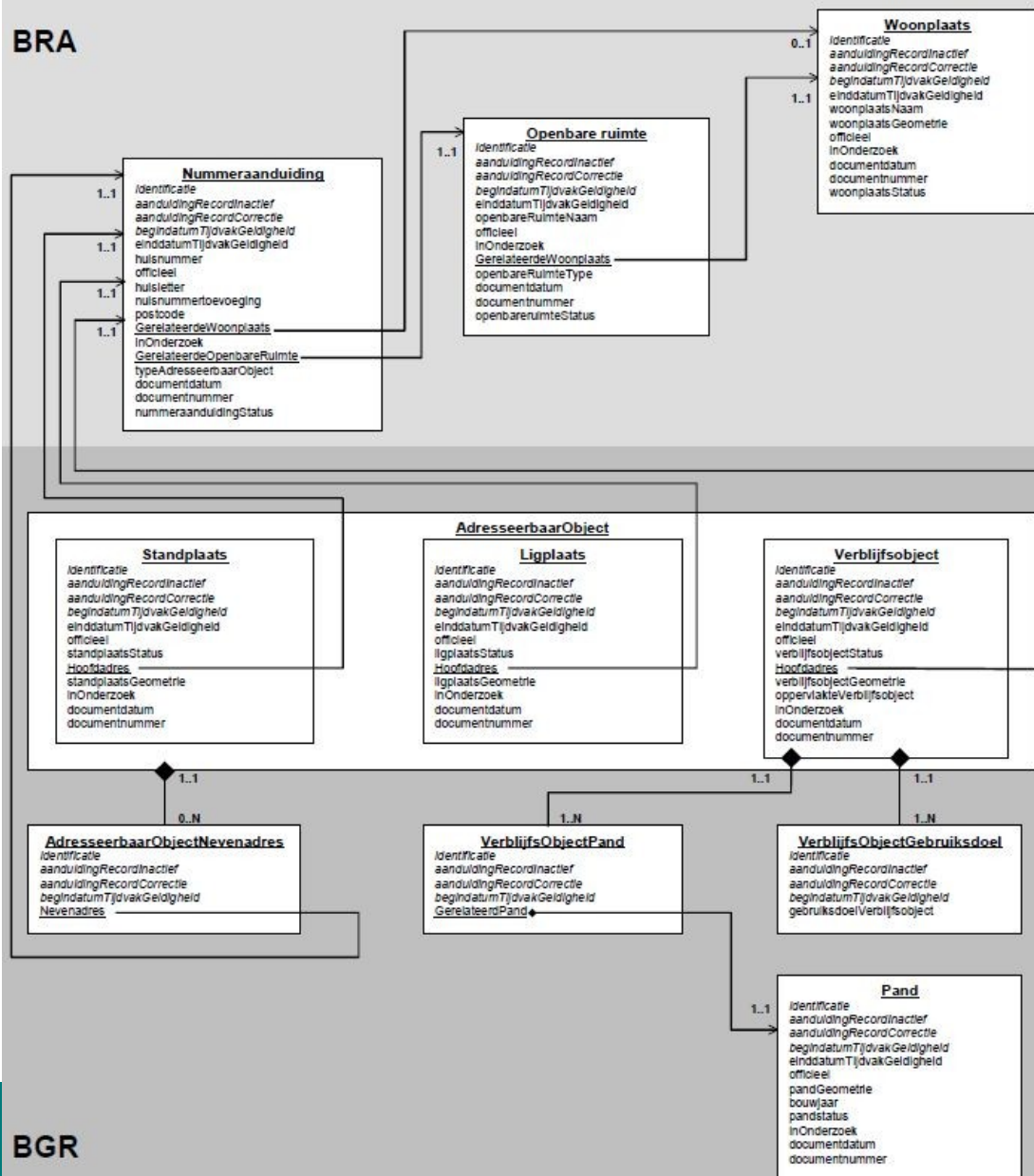
# Veelvoorkomende vragen

Voor Den Bosch wil ik een:

- Lijst met alle adressen van winkels en kantoren
- Selectie van alle panden met winkel- en woonfunctie
- Het zwaartepunt van de bebouwing
- Kaart gemiddelde bouwjaar per buurt

# Data model BAG

BRA



Extra tabellen van NL Extract

- Adres
- Gemeente
- Gemeente\_provincie
- Provincie

BGR

# alle adressen van winkels en kantoren

SELECT

openbareruimtenaam,  
huisnummer, huisletter, huisnummertoevoeging,  
postcode, woonplaatsnaam,  
ARRAY\_TO\_STRING(array\_agg(gebruiksdoelverblijfsobject),';') gebruiksdoel

FROM

bag\_import.verblijfsobjectactueelbestaand vbo  
INNER JOIN bag\_import.verblijfsobjectgebruiksdoelactueel vbo\_gd  
ON vbo.identificatie = vbo\_gd.identificatie  
INNER JOIN bag\_import.adres adr  
ON vbo.identificatie = adr.adresseerbaarobject

WHERE

gebruiksdoelverblijfsobject in ('winkelfunctie','kantoorfunctie') AND  
woonplaatsnaam = "'s-Hertogenbosch'

GROUP BY

openbareruimtenaam,  
huisnummer,  
huisletter,  
huisnummertoevoeging,  
postcode,  
woonplaatsnaam

# Adres en verblijfsobject

## adres

adres\_id

objecttype

openbareruimtenaam

huisnummer

huisletter

huisnummertoevoeging

postcode

woonplaatsnaam

the\_geom

## verblijfsobject

identificatie

adres\_id

oppervlakteverblijfsobject

the\_geom

aantal\_pand

gerelateerdpand

winkelfunctie

logiesfunctie

gezondheidszorgfunctie

kantoorfunctie

bijeenkomstfunctie

woonfunctie

onderwijsfunctie

industriefunctie

overige\_gebruiksfunctie

sportfunctie

celfunctie

# Tabel: Verblijfsobject

```
CREATE TEMPORARY VIEW temp_vw_vbo_gebruiksdoel_aggregated
SELECT
    identificatie,
    ARRAY_TO_STRING(array_agg(gebruiksdoelverblijfsobject),';') doel
FROM
    bag_import.verblijfsobjectgebruiksdoelpeildatum
GROUP BY
    identificatie;
```

# Tabel: Verblijfsobject

SELECT

identificatie,  
strpos(doel, 'winkelfunctie') > 0 winkelfunctie,  
strpos(doel, 'logiesfunctie') > 0 logiesfunctie,  
strpos(doel, 'gezondheidszorgfunctie') > 0 gezondheidszorgfunctie,  
strpos(doel, 'kantoorfunctie') > 0 kantoorfunctie,  
strpos(doel, 'bijeenkomstfunctie') > 0 bijeenkomstfunctie,  
strpos(doel, 'woonfunctie') > 0 woonfunctie,  
strpos(doel, 'onderwijsfunctie') > 0 onderwijsfunctie,  
strpos(doel, 'industriefunctie') > 0 industriefunctie,  
strpos(doel, 'overige gebruiksfunctie') > 0 overige\_gebruiksfunctie,  
strpos(doel, 'sportfunctie') > 0 sportfunctie,  
strpos(doel, 'celfunctie') > 0 celfunctie

FROM

temp\_vw\_vbo\_gebruiksdoel\_aggregated;



# View: vw\_verblijfsobject

```
CREATE OR REPLACE VIEW bag.vw_verblijfsobject AS  
SELECT
```

```
--adres kolommen
```

```
(((openbareruimtenaam || ' ') ||  
  adres.huisnummer::TEXT) || ' ') ||  
  BTRIM((COALESCE(adres.huisletter, '') || ' ') ||  
COALESCE(adres.huisnummertoevoeging, '')) AS adres,  
(adres.postcode || ' ') || woonplaatsnaam AS plaats,  
oppervlakteverblijfsobject AS oppervlakte_ruw,  
oppervlakteverblijfsobject::TEXT || ' m2' AS oppervlakte,  
verblijfsobject.winkelfunctie,
```

```
--overige functies
```

```
FROM
```

```
    bag.verblijfsobject
```

```
    INNER JOIN bag.adres ON verblijfsobject.adres_id = adres.adres_id;
```

# alle adressen van winkels en kantoren

SELECT

openbareruimtenaam,  
huisnummer,  
huisletter,  
huisnummertoevoeging,  
postcode,  
woonplaatsnaam,  
adres,  
plaats

FROM

bag.vw\_verblijfsobject

WHERE

(winkelfunctie OR kantoorfunctie) AND  
woonplaatsnaam = "'s-Hertogenbosch'

# panden met winkel- en woonfunctie

```
WITH kandidaat_panden AS (  
  SELECT  
    vbo_p.gerelateerdpand,  
    COUNT(DISTINCT gebruiksdoelverblijfsobject)  
  
  FROM  
    bag_import.verblijfsobjectpand vbo_p  
    INNER JOIN bag_import.verblijfsobjectgebruiksdoelactueel vbo_gd  
      ON vbo_p.identificatie = vbo_gd.identificatie  
    INNER JOIN bag_import.adres adr  
      ON vbo_p.identificatie = adr.adresseerbaarobject  
  
  WHERE  
    woonplaatsnaam = "'s-Hertogenbosch' AND  
    gebruiksdoelverblijfsobject IN ('woonfunctie', 'winkelfunctie')  
  
  GROUP BY gerelateerdpand  
  HAVING COUNT(DISTINCT gebruiksdoelverblijfsobject) > 1)  
  SELECT  
    p.*  
  
  FROM  
    kandidaat_panden kp  
    INNER JOIN bag_import.pandactueelbestaand p  
      ON kp.gerelateerdpand = p.identificatie
```

# Pand, pand\_woonplaats,

## **pand**

identificatie

pandstatus

bouwjaar

aantal\_vbo

aantal\_winkelfunctie

aantal\_logiesfunctie

aantal\_gezondheidszorgfunctie

aantal\_kantoorfunctie

aantal\_bijeenkomstfunctie

aantal\_woonfunctie

aantal\_onderwijsfunctie

aantal\_industriefunctie

aantal\_overige\_gebruiksfunctie

aantal\_sportfunctie

aantal\_celfunctie

the\_geom

## **pand\_woonplaats\_2014**

identificatie

woonplaats\_id

## **woonplaats\_gemeente\_2014**

id

woonplaatsnaam

geom

gm\_code

gm\_naam

pr\_code

pr\_naam

## **pand\_cbs\_2014**

identificatie

gm\_code

wk\_code

bu\_code

## **adresseerbaarobject\_woonplaats\_2014**

## **adresseerbaarobject\_cbs\_2014**

# Tabel: pand

```
CREATE TEMPORARY TABLE temp_pand_vbo
SELECT
    gerelateerdpand,
    count(*) aantal_vbo,
    sum(winkelfunctie::INTEGER) aantal_winkelfunctie ,
    sum(logiesfunctie::INTEGER) aantal_logiesfunctie ,
    sum(gezondheidszorgfunctie::INTEGER) aantal_gezondheidszorgfunctie ,
    sum(kantoorfunctie::INTEGER) aantal_kantoorfunctie ,
    sum(bijeenkomstfunctie::INTEGER) aantal_bijeenkomstfunctie ,
    sum(woonfunctie::INTEGER) aantal_woonfunctie ,
    sum(onderwijsfunctie::INTEGER) aantal_onderwijsfunctie ,
    sum(industriefunctie::INTEGER) aantal_industriefunctie ,
    sum(overige_gebruiksfunctie::INTEGER) aantal_overige_gebruiksfunctie ,
    sum(sportfunctie::INTEGER) aantal_sportfunctie ,
    sum(celfunctie::INTEGER) aantal_celfunctie
FROM
    bag.verblijfsobject
GROUP BY
    gerelateerdpand;
```

# Tabel: pand

```
INSERT INTO bag.pand
SELECT DISTINCT
    identificatie,
    pandstatus ,
    bouwjaar::INT,
    COALESCE(aantal_vbo,0),
    COALESCE(aantal_winkelfunctie,0),
    COALESCE(aantal_logiesfunctie,0),
    COALESCE(aantal_gezondheidszorgfunctie,0),
    COALESCE(aantal_kantoorfunctie,0),
    COALESCE(aantal_bijeenkomstfunctie,0),
    COALESCE(aantal_woonfunctie,0),
    COALESCE(aantal_onderwijsfunctie,0),
    COALESCE(aantal_industriefunctie,0),
    COALESCE(aantal_overige_gebruiksfunctie,0),
    COALESCE(aantal_sportfunctie,0),
    COALESCE(aantal_celfunctie,0),
    geovlak
FROM
    bag_import.pandpeildatumbestaand p
    LEFT JOIN temp_pand_vbo pv
        ON p.identificatie = pv.gerelateerdpand;
```

# Tabel: pand\_woonplaats

```
INSERT INTO temp_pand
SELECT
    identificatie,
    ST_CENTROID(the_geom)
FROM
    bag.pand;
```

```
CREATE INDEX temp_pand_gist
ON temp_pand
USING GIST(centroid);
```

## Loop door woonplaatsen

```
INSERT INTO bag.pand_woonplaats_2014
SELECT
    identificatie,
    recWoonplaats.id
FROM
    temp_pand
WHERE
    ST_DWITHIN(recWoonplaats.geom,centroid,0);
```

# Panden met winkel- en woonfunctie

```
SELECT
    p.*
FROM
    bag.pand p
    INNER JOIN bag.pand_woonplaats_2014 pw
        ON p.identificatie = pw.identificatie
    INNER JOIN bag.woonplaats_gemeente_2014 wg
        ON pw.woonplaats_id = wg.id
WHERE
    p.aantal_winkelfunctie > 0 AND
    p.aantal_woonfunctie > 0 AND
    woonplaatsnaam = "'s-Hertogenbosch'
```



# Zwaartepunt van de bebouwing

```
SELECT woonplaatsnaam,  
       ST_SetSRID(ST_MAKEPOINT(avg(x),avg(y)),28992)  
FROM ( SELECT  
       woonplaatsnaam, st_x(vbo.geopunt) x, st_y(vbo.geopunt) y  
FROM  
       bag_import.verblijfsobjectactueelbestaand vbo  
       INNER JOIN bag_import.adres adr  
       ON vbo.identificatie = adr.adresseerbaarobject  
UNION ALL  
SELECT  
       woonplaatsnaam, st_x(ST_CENTROID(geovlak)) , st_y(ST_CENTROID(geovlak))  
FROM  
       bag_import.standplaatsactueelbestaand vbo  
       INNER JOIN bag_import.adres adr  
       ON vbo.identificatie = adr.adresseerbaarobject  
UNION ALL  
SELECT  
       woonplaatsnaam, st_x(ST_CENTROID(geovlak)), st_y(ST_CENTROID(geovlak))  
FROM  
       bag_import.ligplaatsactueelbestaand vbo  
       INNER JOIN bag_import.adres adr  
       ON vbo.identificatie = adr.adresseerbaarobject  
  ) as all_adres  
WHERE woonplaatsnaam = "'s-Hertogenbosch'  
GROUP BY woonplaatsnaam
```

# Woonplaats\_points\_2014

## woonplaats\_points\_2014

woonplaats\_id

x\_extent

y\_extent

aantal

aantal\_ligplaats

aantal\_standplaats

aantal\_vbo

aantal\_bijeenkomstfunctie

aantal\_celfunctie

aantal\_gezondheidszorgfunctie

aantal\_industriefunctie

aantal\_kantoorfunctie

aantal\_logiesfunctie

aantal\_onderwijsfunctie

aantal\_overige\_gebruiksfunctie

aantal\_sportfunctie

aantal\_winkelfunctie

aantal\_woonfunctie

the\_geom

# Tabel:woonplaats\_points\_2014

```
CREATE TEMPORARY TABLE temp_objecten AS
SELECT
    woonplaats_id,
    0 ligplaats,
    0 standplaats,
    1 vbo,
    bijeenkomstfunctie::INT,
    celfunctie::INT,
    gezondheidszorgfunctie::INT,
    --overige functies
    woonfunctie::INT,
    ST_X(v.the_geom) x,
    ST_Y(v.the_geom) y
FROM
    bag.adresseerbaarobject_woonplaats_2014 a
    INNER JOIN bag.verblijfsobject v
        ON v.identificatie = a.identificatie
```

# Tabel:woonplaats\_points\_2014

```
INSERT INTO bag.woonplaats_points_2014
SELECT
    woonplaats_id,
    (max(x) - MIN(x))::INT x_extent,
    (max(y) - MIN(y))::INT y_extent,
    COUNT(*) aantal,
    SUM(ligplaats) aantal_ligplaats,
    SUM(standplaats) aantal_standplaats,
    SUM(vbo) aantal_vbo,
    SUM(bijeenkomstfunctie) aantal_bijeenkomstfunctie,
    SUM(celfunctie) aantal_celfunctie,
    SUM(gezondheidszorgfunctie) aantal_gezondheidszorgfunctie,
    --overige functies
    SUM(woonfunctie) aantal_woonfunctie,
    ST_SetSRID(ST_MAKEPOINT(avg(x),avg(y)),28992)
FROM
    temp_objecten
GROUP BY
    woonplaats_id;
```

# Zwaartepunt van de bebouwing

```
SELECT
    *
FROM
    bag.woonplaats_points_2014 wp
    INNER JOIN bag.woonplaats_gemeente_2014 wg
        ON wp.woonplaats_id = wg.id
WHERE
    woonplaatsnaam = "'s-Hertogenbosch'
```

# Overige punt tabellen

## **Tabellen:**

cbs\_2014\_gemeente\_points  
cbs\_2014\_wijk\_points  
cbs\_2014\_buurt\_points  
pc4\_points  
pc5\_points  
pc6\_points

## **Views:**

vw\_cbs\_2014\_gemeente  
vw\_cbs\_2014\_wijk  
vw\_cbs\_2014\_buurt

# View: vw\_cbs\_2014\_buurt

```
SELECT
  cbs.buurtcode,
  cbs.buurt,
  cbs.wijk,
  cbs.gemeente,
  cbs.aantal_inwoners,
  cbs.mannen,
  cbs.vrouwen,
  cp.aantal,
  cp.aantal_ligplaats,
  cp.aantal_standplaats,
  cp.aantal_vbo,
  cp.aantal_bijeenkomstfunctie,
  cp.aantal_celfunctie,
  cp.aantal_gezondheidszorgfunctie,
  --overige functies
  cp.aantal_woonfunctie,
  cbs.geom
FROM adm.vw_cbs_buurt_2014 cbs
  INNER JOIN bag.cbs_2014_buurt_points cp
    ON cbs.buurtcode = cp.bu_code;
```

# Kaart gemiddelde bouwjaar per buurt

```
SELECT
    buurt,
    geom,
    AVG(p.bouwjaar)::INTEGER gemiddeld_bouwjaar
FROM
    bag_import.pandactueelbestand p
    INNER JOIN adm.vw_cbs_buurt_2014
        ON ST_DWITHIN(geovlak,geom,0)
WHERE
    gemeente = "'s-Hertogenbosch'
GROUP BY
    buurt,
    geom
```



# Bouwjaar tabellen

## **bouwjaar\_buurt\_2014**

bu\_code,  
gem\_bouwjaar  
stddev\_bouwjaar  
perc\_bj\_1600  
perc\_bj\_1699  
perc\_bj\_1799  
perc\_bj\_1899  
perc\_bj\_1944  
perc\_bj\_1959  
perc\_bj\_1969  
perc\_bj\_1979  
perc\_bj\_1989  
perc\_bj\_1999  
perc\_bj\_2009  
perc\_bj\_2019  
perc\_bj\_onbekend  
aantal\_panden

## **Tabellen:**

bouwjaar\_wijk\_2014  
bouwjaar\_gemeente\_2014  
bouwjaar\_woonplaats\_2014  
bouwjaar\_pc4  
bouwjaar\_pc5  
bouwjaar\_pc6

## **Views:**

vw\_bouwjaar\_buurt\_2014  
vw\_bouwjaar\_wijk\_2014  
vw\_bouwjaar\_gemeente\_2014

# Tabel: bouwjaar\_buurt\_2014

gem\_bouwjaar

ROUND(avg(CASE WHEN (bouwjaar BETWEEN 1100 AND 2019) THEN bouwjaar END)) ,

stddev\_bouwjaar

ROUND(stddev\_pop(CASE WHEN (bouwjaar BETWEEN 1100 AND 2019) THEN bouwjaar END),1) ,

perc\_bj\_1600

COALESCE(  
 ROUND(SUM(CASE WHEN (bouwjaar BETWEEN 1100 AND 1599) THEN 1 END)/(COUNT(\*))::NUMERIC,2)  
,0.00) ,

perc\_bj\_1699

COALESCE(  
 ROUND(SUM(CASE WHEN (bouwjaar BETWEEN 1600 AND 1699) THEN 1 END)/(COUNT(\*))::NUMERIC,2)  
,0.00) ,

perc\_bj\_onbekend

COALESCE(  
 ROUND(SUM(CASE WHEN (bouwjaar < 1100 OR bouwjaar > 2019) THEN 1 AND)/(COUNT(\*))::NUMERIC,2),  
 0),

# View: vw\_bouwjaar\_buurt\_2014

```
SELECT
  gemeentecode, gemeente,
  aantal_inwoners, mannen, vrouwen,
  gem_bouwjaar as "Gemiddeld_bouwjaar",
  stddev_bouwjaar as "standaard_deviatie_bouwjaar",
  (perc_bj_1600 + perc_bj_1699 + perc_bj_1799) * 100 as perc_bj_1799,
  --bouwjaren
  perc_bj_onbekend * 100 as perc_bj_onbekend,
  aantal_panden,
  CASE GREATEST ((perc_bj_1600 + perc_bj_1699 + Perc_bj_1799), bouwjaren, perc_bj_2019)
    WHEN (perc_bj_1600 + perc_bj_1699 + perc_bj_1799) THEN 'voor 1800'
    WHEN perc_bj_1899 THEN '1800 - 1899'
    --bouwjaren
    WHEN perc_bj_2019 THEN '2010 - 2019'
    ELSE 'Onbekend'
  END AS "Meest_voorkomend_bouwjaar",
  geom
FROM
  adm.vw_cbs_gemeente_2014 cbs
  INNER JOIN bag.bouwjaar_gemeente_2014 bj
    ON cbs.gemeentecode = gm_code;
```

# Kaart gemiddelde bouwjaar per buurt

```
SELECT  
  buurt,  
  gemiddeld_bouwjaar,  
  geom  
FROM  
  bag.vw_bouwjaar_buurt_2014  
WHERE  
  gemeente = "'s-Hertogenbosch'
```

