Specification Database Basic and Professional

Author: Pim Broekhof

Version: 2.1 (revision 249)

Date: 2024-10-16

Subject: Specification of Postcode.nl national postcode database Basic and Professional products.

Postcode.nl B.V.
Julianastraat 30, 2012 ES Haarlem, The Netherlands
Telephone: (+31) 23 53 25 689
Email: info@postcode.nl

Website: www.postcode.nl
Chamber of Commerce number: 34185878
VAT number: NL8120.08.248.B01
SWIFT: RABONL2U IBAN: NL17RABO0122779924

Table of Contents

1 Version changelog

Version	Datum	Changes
2.1	2024-10-16	Updated Bibliography links
2.0	2024-05-21	Conversion from old specification document

2 Products

Postcode.nl offers two products: the National Postcode Database Basic (referred to as "Basic") and the National Postcode Database Professional (referred to as "Pro"). Each product comes with various format and delivery options.

2.1 Product features

Below is a quick overview of the available options for each product:

Features	Basic	Pro
Updates	Weekly, monthly, quarterly, yearly	Weekly, monthly, quarterly, yearly
Normalization	Single table (merged)	Single table (merged), multiple tables (normalized)
Formats	CSV, MS SQL, ANSI SQL, PCTR	CSV, MS SQL, ANSI SQL, PCTP
Encoding	ASCII and/or UTF-8	ASCII and/or UTF-8
Precision	Postcode ranges	Addresses and postcode ranges
Records	± 650.000	± 9.500.000
Address information	No	Yes (surface area, type, usage)
Unique keys	Postcode.nl IDs	Postcode.nl IDs, Government BAG IDs
Coordinates	No	Yes

Customers can receive multiple variants of the Basic database with a single Basic subscription, provided all deliveries are made to the same organization. Similarly, with a single Pro subscription, customers can receive multiple variants of both Basic and Pro databases, as long as all deliveries go to the same organization.

2.2 Basic product

The Basic product contains addresses organized by postcode ranges. Each postcode range includes a series of house numbers sharing the same postcode on the same street, where the numbers are either all odd or all even. This also includes non-physical PO Boxes ("postbussen") postcode ranges.

Within a postcode range, all addresses will have the same postcode, street, and city. However, there is no guarantee that every house number within the postcode range exists, nor is there information about specific house number additions.

The database includes approximately 650,000 postcode range records, making it simpler to use compared to the larger Pro product. The Basic product is only available as a single table.

Additionally, the Basic product includes a "PCTR" format that is compatible with the "PostNL reeks postcodetabel" format.

2.3 Pro product

The Pro product includes all the information available in the Basic product, with the addition of individual address details. This includes house numbers, number additions, coordinates, surface area, designated usage options, and the type of address. This detailed address-level information is referred to as "perceel" information.

For PO Box ("postbus") postcode ranges, there are no individual addresses since they are not physical locations. Instead, a single placeholder address is provided for each PO Box postcode range.

The database contains approximately 9,500,000 address records. The Pro product is available both as a single table and as a collection of separate normalized tables, including address ("perceel"), postcode range ("pcreeks"), street ("straat"), city ("plaats"), municipality ("gemeente"), and province ("provincies").

Additionally, the Pro product includes a "PCTP" format that is compatible with the "PostNL huisnummer postcodetabel" format.

3 Formats

There are various methods available for importing the initial full dump of the database and applying periodic updates, referred to as mutations. See Section 7 for a description of each field.

3.1 ANSI SQL

SQL commands compatible with ANSI SQL databases such as Oracle, PostgreSQL, Sybase, MySQL, MariaDB, and SQL Server.

Mutation records are provided as a separate table. See Section 8.1 for a description of the steps needed to apply these mutation records.

3.2 ANSI SQL (Applied Mutations)

SQL commands compatible with ANSI SQL databases such as Oracle, PostgreSQL, Sybase, MySQL, MariaDB, and SQL Server.

Mutation records are provided as direct SQL commands on the database tables.

3.3 MS SQL

SQL commands compatible with Microsoft SQL Server, including "GO" statements after every 1000 SQL statements.

Mutation records are provided as a separate table. See section 8.1 for a description of the steps needed to apply these mutation records.

3.4 MS SQL (Applied Nutations)

SQL commands compatible with Microsoft SQL Server, including "GO" statements after every 1000 SQL statements.

Mutation records are provided as direct SQL commands on the database tables.

3.5 CSV

CSV records in the RFC-4180 format.

Mutation records are provided as "insert", "delete", "update", and "downdate" records in separate mutation tables. Applying these mutation records to the actual database tables must be done by the application.

3.6 PCT-BAG

Records using a "PostNL postcodetabel" compatible format, also called PCTR for the Basic database product and PCTP for the Pro database product. This format is commonly used in CRM systems like AX Dynamics.

4 Delivery

Delivery of both the initial full database dump and mutations can be received in several ways:

- An email is sent with a download service link.
- The Data API is used to check the availability of deliveries with download service links.

The download service allows at least 50 downloads for each link over a period of at least 6 months. Beyond this limit, downloads may be denied, and after this period, the download link may fail.

Therefore, it is recommended not to make repetitive downloading of the same delivery part of client functionality, including unit and integration testing.

For more information, refer to the Data API specification document [Data-API].

4.1 Frequency

Depending on the chosen delivery frequency, there is an option to either send a full database dump each time or a mutation delivery.

The following delivery frequencies are available:

Frequency	When
Weekly	Monday of each week
Monthly	First Monday of each month
Quarterly	First Monday of January, April, July and October
Yearly	First Monday of January

5 SQL and CSV formats

For each of the main products (Basic and Pro), three encoding variants are available:

Encoding	Description
UTF-8	Only UTF-8 variants of fields are included
ASCII	Only ASCII variants of fields are included
UTF-8 and ASCII	Both UTF-8 and ASCII variants of fields are included

For the Pro product, there is also an option to deliver the database in a normalized format, with separate tables.

Normalization	Description
Merged	All fields are in a single table "pcdata"
Normalized	Fields are split into separate tables. (Pro only)

5.1 Basic Merged, Fields per Encoding

All fields are contained in a single merged "pcdata" table. See section 7 for field definitions.

Table	Field	UTF-8	ASCII	UTF-8 and ASCII
pcdata	reeksid	Yes	Yes	Yes
pcdata	wijkcode	Yes	Yes	Yes
pcdata	lettercombinatie	Yes	Yes	Yes
pcdata	huisnr_van	Yes	Yes	Yes
pcdata	huisnr_tm	Yes	Yes	Yes
pcdata	reeksindicatie	Yes	Yes	Yes
pcdata	straatnaam		Yes	Yes
pcdata	straatnaam_nen		Yes	Yes
pcdata	straatnaam_utf8	Yes		Yes
pcdata	straatnaam_utf8_nen	Yes		Yes
pcdata	plaatsnaam		Yes	Yes
pcdata	plaatsnaam_utf8	Yes		Yes
pcdata	plaatsnaam_utf8_nen	Yes		Yes
pcdata	gemeentenaam		Yes	Yes
pcdata	gemeentenaam_utf8	Yes		Yes
pcdata	gemeentenaam_utf8_nen	Yes		Yes
pcdata	provincienaam		Yes	Yes

5.2 Pro Normalized, Fields per Encoding

When using merged normalization, all fields are contained in a single "pcdata" table instead of their own separate tables. See section 7 for field definitions.

Table	Field	UTF-8	ASCII	UTF-8 and ASCII
perceel	perceelid	Yes	Yes	Yes
perceel	perceeltype	Yes	Yes	Yes
perceel	huisnr	Yes	Yes	Yes
perceel	huisnr_bag_letter	Yes	Yes	Yes
perceel	huisnr_bag_toevoeging	Yes	Yes	Yes

Table	Field	UTF-8	ASCII	UTF-8 and ASCII
perceel	bag_nummeraanduidingid	Yes	Yes	Yes
perceel	bag_adresseerbaarobjectid	Yes	Yes	Yes
perceel	breedtegraad	Yes	Yes	Yes
perceel	lengtegraad	Yes	Yes	Yes
perceel	rdx	Yes	Yes	Yes
perceel	rdy	Yes	Yes	Yes
perceel	oppervlakte	Yes	Yes	Yes
perceel	gebruiksdoel	Yes	Yes	Yes
perceel	reeksid	Yes	Yes	Yes
pcreeks	reeksid	Yes	Yes	Yes
pcreeks	wijkcode	Yes	Yes	Yes
pcreeks	lettercombinatie	Yes	Yes	Yes
pcreeks	huisnr_van	Yes	Yes	Yes
pcreeks	huisnr_tm	Yes	Yes	Yes
pcreeks	reeksindicatie	Yes	Yes	Yes
pcreeks	straatid	Yes	Yes	Yes
pcreeks	plaatsid	Yes	Yes	Yes
pcreeks	gemeenteid	Yes	Yes	Yes
pcreeks	provinciecode	Yes	Yes	Yes
straat	straatid	Yes	Yes	Yes
straat	straatnaam		Yes	Yes
straat	straatnaam_nen		Yes	Yes
straat	straatnaam_utf8	Yes		Yes
straat	straatnaam_utf8_nen	Yes		Yes
straat	plaatsid	Yes	Yes	Yes
plaats	plaatsid	Yes	Yes	Yes
plaats	plaatsnaam		Yes	Yes
plaats	plaatsnaam_utf8	Yes		Yes
plaats	plaatsnaam_utf8_nen	Yes		Yes
plaats	plaatscode	Yes	Yes	Yes
plaats	gemeenteid	Yes	Yes	Yes
gemeente	gemeentenaam		Yes	Yes
gemeente	gemeentenaam_utf8	Yes		Yes
gemeente	gemeentenaam_utf8_nen	Yes		Yes
gemeente	provinciecode	Yes	Yes	Yes
provincies	provinciecode	Yes	Yes	Yes
provincies	provincienaam	Yes	Yes	Yes

5.3 Pro Merged, Fields per Encoding

All fields are contained in a single merged "pcdata" table. See section 7 for field definitions.

Table	Field	UTF-8	ASCII	UTF-8 and ASCII
pcdata	perceelid	Yes	Yes	Yes
pcdata	perceeltype	Yes	Yes	Yes
pcdata	huisnr	Yes	Yes	Yes
pcdata	huisnr_bag_letter	Yes	Yes	Yes
pcdata	huisnr_bag_toevoeging	Yes	Yes	Yes
pcdata	bag_nummeraanduidingid	Yes	Yes	Yes
pcdata	bag_adresseerbaarobjectid	Yes	Yes	Yes
pcdata	breedtegraad	Yes	Yes	Yes
pcdata	lengtegraad	Yes	Yes	Yes
pcdata	rdx	Yes	Yes	Yes
pcdata	rdy	Yes	Yes	Yes
pcdata	oppervlakte	Yes	Yes	Yes
pcdata	gebruiksdoel	Yes	Yes	Yes
pcdata	reeksid	Yes	Yes	Yes
pcdata	wijkcode	Yes	Yes	Yes
pcdata	lettercombinatie	Yes	Yes	Yes
pcdata	huisnr_van	Yes	Yes	Yes
pcdata	huisnr_tm	Yes	Yes	Yes
pcdata	reeksindicatie	Yes	Yes	Yes
pcdata	straatid	Yes	Yes	Yes
pcdata	straatnaam		Yes	Yes
pcdata	straatnaam_nen		Yes	Yes
pcdata	straatnaam_utf8	Yes		Yes
pcdata	straatnaam_utf8_nen	Yes		Yes
pcdata	plaatsid	Yes	Yes	Yes
pcdata	plaatsnaam		Yes	Yes
pcdata	plaatsnaam_utf8	Yes		Yes
pcdata	plaatsnaam_utf8_nen	Yes		Yes
pcdata	gemeenteid	Yes	Yes	Yes
pcdata	gemeentenaam		Yes	Yes
pcdata	gemeentenaam_utf8	Yes		Yes
pcdata	gemeentenaam_utf8_nen	Yes		Yes
pcdata	provinciecode	Yes	Yes	Yes
pcdata	provincienaam	Yes	Yes	Yes

6 PCT-BAG formats

The "PCT-BAG" format is compatible with the "PostNL postcodetabel" format. The standard "ASCII" and "UTF-8" encoding options cover the most commonly used variants, as described in the PCT-BAG specification document [PCT-BAG] (in Dutch).

It is also possible to request a custom "PCT-BAG" format tailored to your specific needs. This may include additional records such as unbuilt streets, different encodings like ASCII city names in both lower and upper case letters instead of all capital letters, and other options. Refer to the PCT-BAG specification document [PCT-BAG] for all current options. New options can also be developed upon request.

6.1 PCT with ASCII Encoding

The ASCII variant of the PCTR and PCTP formats matches the "ascii-compatibel" format as described in the PCT-BAG specification document [PCT-BAG]. It contains field variants most compatible with the original PCT format.

Main features are as follows:

- All fields are ASCII encoded.
- City and municipality names are in capital letters.
- City names have a custom suffix if not unique within the Netherlands.
- All derived fields without an official equivalent field are still filled in.

For example, there are two cities named "Hoorn" in the Netherlands. The city in the province of "Noord-Holland" is named "HOORN NH," while the one in "Friesland" is named "HOORN FR." Derived non-official fields, such as the shorter "Plaatsnaam (PostNL)," are still filled in and are derived from the official names.

6.2 PCT with UTF-8 Encoding

The UTF-8 variant of the PCTR and PCTP formats matches “utf-8-postcode” format as described in the PCT-BAG specification document [PCT-BAG]. This format is similar to the ASCII format, with the following differences:

- Full and NEN street, city and municipality names are UTF-8 encoded.
- Full and NEN city and municipality names are in both capital and lowercase letters.
- Full and NEN city names do not receive a suffix if not unique within the Netherlands.
- Derived fields remain the same as the ASCII encoded variant.

As the PCT-BAG uses a fixed-length column format, the column length is defined by UTF-8 grapheme length, not bytes.

7 Field definitions

We provide database-agnostic descriptions of fields. Depending on your database implementation, these fields may be stored as INT, DECIMAL, VARCHAR, ENUM, etc.

Typical mappings to common database field types include:

- [type: int]: Fields are mapped to 32-bit unsigned integers unless otherwise specified.
- [type: decimal], [range: [-XX.XXXX, XX.XXXX]]: Fixed precision fields with precision specified by their range.
- [type: text], [length: X]: Fields are always encoded in UTF-8 and have a specified UTF-8 grapheme length.
- [subset: ASCII], [length: X]: For ASCII fields, the grapheme length is the same as the ASCII string length in bytes.
- [empty-value: X]: Fields can be encoded as NULL values, with the given value mapped to NULL.
- [unique: X]: Fields are unique within a given domain, but this uniqueness may be temporarily broken during mutations.

Not all mappings need to be implemented. These definitions are also described in a ".schema" file included in each delivery.

7.1 Address Fields (“perceel”)

The "perceel" table contains fields specific to one address. For PO Box ("postbus") addresses, a single dummy record is provided for the entire postcode range, as address-specific information is not available for PO boxes.

Field	Description
perceelid	Postcode.nl identifier for 'perceel' address records. - [type: int]: Integer number - [primary key]: Unique identifier for 'perceel' address records
perceeltype	Type of 'perceel' address record. - [type: text]: UTF-8 encoded string value - [length: 15]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [value: 'verblijfsobject']: Permanent residence - [value: 'standplaats']: Mobile home residence - [value: 'ligplaats']: Houseboat residence - [value: 'postbus']: Postbox, one address record per range - [value: 'onbebouwd']: Unbuilt street, one address record per range
huisnr	House number of an address. - [type: int]: Integer number - [range: [1, 99999]]: Positive numbers - [empty-value: 0]: When perceeltype is 'postbus' (`straatnaam` = 'Postbus') or 'onbebouwd' (`lettercombinatie` = '')
huisnr_bag_letter	House letter part of the house number addition. - [type: text]: UTF-8 encoded string value - [length: 1]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [empty-value: '']: House letter is optional

Field	Description
huisnr_bag_toevoeging	House extension part of the house number addition. - [type: text]: UTF-8 encoded string value - [length: 4]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [regexp: [A-Z0-9-]{0,4}]: House extension is a single word - [empty-value: '']: House extension is optional
bag_nummeraanduidingid	Identifier linking to the currently active BAG 'Nummeraanduiding' record. - [type: text]: UTF-8 encoded string value - [length: 16]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [regexp: [0-9]{16}']: BAG 'Nummeraanduiding' identifier - [empty-value: '']: When perceeltype equals 'postbus' (`straatnaam` = 'Postbus') or 'onbebouwd' (`lettercombinatie` = '') - [unique: dataset]: Non-empty value is unique within the dataset, but may temporarily be non-unique during update process
bag_adresseerbaarobjectid	Identifier linking to the currently active BAG 'AdresseerbaarObject' record. - [type: text]: UTF-8 encoded string value - [length: 16]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [regexp: [0-9]{16}']: BAG 'AdresseerbaarObject' identifier - [empty-value: '']: When perceeltype equals 'postbus' (`straatnaam` = 'Postbus') or 'onbebouwd' (`lettercombinatie` = ''), or `AdresseerbaarObject` is inactive or unknown
breedtegraad	Latitude coordinate of address in '(EPSG) 4326 WGS84' geodetic coordinate system. - [type: decimal]: Fixed precision and scale number - [range: [-90.00000000, 90.00000000]]: Precision of 8 decimals - [empty-value: '']: When `bag_adresseerbaarobjectid` is empty
lengtegraad	Longitude coordinate of address in '(EPSG) 4326 WGS84' geodetic coordinate system. - [type: decimal]: Fixed precision and scale number - [range: [-180.00000000, 180.00000000]]: Precision of 8 decimals - [empty-value: '']: When `bag_adresseerbaarobjectid` is empty
rdx	X coordinate of address, in '(EPSG) 28992 Amersfoort / RD New' ('Rijksdriehoeksmeting') planar coordinate system. - [type: decimal]: Fixed precision and scale number - [range: [0.0000, 300000.0000]]: Precision of 4 decimals - [empty-value: '']: When `bag_adresseerbaarobjectid` is empty
rdy	Y coordinate of address, in '(EPSG) 28992 Amersfoort / RD New' ('Rijksdriehoeksmeting') planar coordinate system. - [type: decimal]: Fixed precision and scale number - [range: [300000.0000, 620000.0000]]: Precision of 4 decimals - [empty-value: '']: When `bag_adresseerbaarobjectid` is empty
oppervlakte	Living area of 'verblijfsobject' accommodation in square meters. - [type: int]: Integer number - [range: [1, 999999]]: When `perceeltype` is 'verblijfsobject' and `bag_adresseerbaarobjectid` is not empty - [empty-value: 0]: Unspecified, when `perceeltype` is not 'verblijfsobject' or `bag_adresseerbaarobjectid` is empty

Field	Description
gebruiksdoel	Functions of 'verblijfsobject' accommodation as a set of zero or more comma-separated options. - [type: text]: UTF-8 encoded string value - [length: 255]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [regexp: [:option:](,[:option:])*]: When `perceeltype` is 'verblijfsobject' and `bag_adresseerbaarobjectid` is not empty - [option: 'woonfunctie']: Residential living function - [option: 'bijeenkomstfunctie']: Gathering function - [option: 'celfunctie']: Forced residence function - [option: 'gezondheidszorgfunctie']: Medical or health-related function - [option: 'industriefunctie']: Industrial or agricultural function - [option: 'kantoorfunctie']: Administration function - [option: 'logiesfunctie']: Recreational or temporary residential living function - [option: 'onderwijsfunctie']: Educational function - [option: 'sportfunctie']: Sport facilitating function - [option: 'winkelfunctie']: Trading of materials, goods, or services function - [option: 'overige gebruiksfunctie']: Other functions where the residential function is of subservient importance - [empty-value: '']: Unspecified, when `perceeltype` is not 'verblijfsobject' or `bag_adresseerbaarobjectid` is empty
reeksid	Postcode.nl identifier for 'pcreeks' postcode range records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'pcreeks' postcode range records

7.2 Postcode range fields (“pcreeks”)

The “pcreeks” table contains fields specific to postcode ranges.

Field	Description
reeksid	Postcode.nl identifier for 'pcreeks' postcode range records. - [type: int]: Integer number - [primary key]: Unique identifier for 'pcreeks' postcode range records.
wijkcode	Four-number 'wijkcode' neighbourhood code, the first part of a postcode. - [type: text]: UTF-8 encoded string value - [length: 4]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [range: ['1000', '9999']]: Neighbourhood code always has length 4
lettercombinatie	Two-character 'lettercombinatie' letter combination, the second part of a postcode. - [type: text]: UTF-8 encoded string value - [length: 2]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [range: ['AA', 'ZZ']]: Always uppercase
huisnr_van	Lowest house number of the postcode range record. - [type: int]: Integer number - [range: 1-99999]: Always positive - [max: huisnr_van]: Must be lower or equal to the field 'huisnr_tm' - [oddness: huisnr_van]: Must match the oddness (even/uneven) of the field 'huisnr_tm'
huisnr_tm	Highest house number of the postcode range record. - [type: int]: Integer number - [range: 1-99999]: Always positive - [min: huisnr_tm]: Must be higher or equal to the field 'huisnr_van' - [oddness: huisnr_van]: Must match the oddness (even/uneven) of the field 'huisnr_van'
reeksindicatie	Postcode range record type indication. - [type: text]: UTF-8 encoded string value - [length: 1]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [value: '0']: Range of odd house numbers - [value: '1']: Range of even house numbers - [empty-value: '']: Unbuilt street
straatid	Postcode.nl identifier for 'straat' street records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'straat' street records
plaatsid	Postcode.nl identifier for 'plaats' locality records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'plaats' locality records
gemeenteid	Postcode.nl identifier for 'gemeente' municipality records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'gemeente' municipality records
provinciecode	Postcode.nl identifier for 'provincie' province records. - [type: text]: UTF-8 encoded string value - [length: 1]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [foreign key]: Foreign key identifier for 'provincie' province records

7.3 Street fields (“straat”)

The “straat” table contains fields specific to streets in a particular city.

Field	Description
straatid	Postcode.nl identifier for 'straat' street records. - [type: int]: Integer number - [primary key]: Unique identifier for 'straat' street records.
straatnaam	Street name as stored in the BAG registry, shortened to match PostNL-defined maximum street name length. - [type: text]: UTF-8 encoded string value - [length: 43]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [unique: locality]: Value is unique within its city, but may temporarily be non-unique during update process - [value: 'Postbus']: PO Box records use the 'Postbus' street name
straatnaam_nen	Street name in accordance with NEN-5825 conventions. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [unique: locality]: Value is unique within its city, but may temporarily be non-unique during update process - [value: 'Postbus']: PO Box records use the 'Postbus' street name
straatnaam_utf8	Street name as stored in BAG registry with diacritics. - [type: text]: UTF-8 encoded string value - [length: 80]: Maximum UTF-8 grapheme length - [unique: locality]: Value is unique within its city, but may temporarily be non-unique during update process - [value: 'Postbus']: PO Box records use the 'Postbus' street name
straatnaam_utf8_nen	Street name as stored in BAG registry with diacritics, in accordance with the NEN-5825 standard. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [unique: locality]: Value is unique within its city, but may temporarily be non-unique during update process - [value: 'Postbus']: PO Box records use the 'Postbus' street name
plaatsid	Postcode.nl identifier for 'plaats' city records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'plaats' locality records.

7.4 City Fields (“plaats”)

The “plaats” table contains fields specific to cities in a particular municipality.

Field	Description
plaatsid	Postcode.nl identifier for 'plaats' city records. - [type: int]: Integer number - [primary key]: Unique identifier for 'plaats' city records.
plaatsnaam	City name in accordance with the NEN-5825 standard. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [unique: municipality]: Value is unique within its municipality, but may temporarily be non-unique during update process
plaatsnaam_utf8	City name as stored in BAG registry with diacritics. - [type: text]: UTF-8 encoded string value - [length: 80]: Maximum UTF-8 grapheme length - [unique: municipality]: Value is unique within its municipality, but may temporarily be non-unique during update process
plaatsnaam_utf8_nen	City name as stored in BAG registry with diacritics, in accordance with the NEN-5825 standard. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [unique: municipality]: Value is unique within its municipality, but may temporarily be non-unique during update process
plaatscode	BAG 'Woonplaats' city identifier. - [type: int]: Integer number - [range: [1000, 9999]]: BAG 'Woonplaats' city identifier - [unique: dataset]: Value is unique within the dataset, but may temporarily be non-unique during update process - [commentary]: In the rare case of multiple possible identifiers, the highest value is used
gemeenteid	Postcode.nl identifier for 'gemeente' municipality records. - [type: int]: Integer number - [foreign key]: Foreign key identifier for 'gemeente' municipality records

7.5 Municipality fields (“gemeente”)

The "gemeente" table contains fields specific to municipalities within a particular province.

Field	Description
gemeenteid	Postcode.nl identifier for 'gemeente' municipality records. - [type: int]: Integer number - [primary key]: Unique identifier for 'gemeente' municipality records
gemeentenaam	Municipality name using PostNL naming conventions. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [unique: dataset]: Value is unique within the dataset, but may temporarily be non-unique during update process - [text: PostNL]: In capital letters
gemeentenaam_utf8	Municipality name as stored in BAG related 'Landelijke Tabel 33' registry with diacritics. - [type: text]: UTF-8 encoded string value - [length: 80]: Maximum UTF-8 grapheme length - [unique: dataset]: Value is unique within the dataset, but may temporarily be non-unique during update process
gemeentenaam_utf8_nen	Municipality name as stored in BAG related 'Landelijke Tabel 33' registry with diacritics, in accordance with the NEN-5825 standard. - [type: text]: UTF-8 encoded string value - [length: 24]: Maximum UTF-8 grapheme length - [unique: dataset]: Value is unique within the dataset, but may temporarily be non-unique during update process
gemeentecode	Municipality code as provided by 'Landedlijke Tabel 33'. - [type: int]: Integer number - [range: [1, 9999]]: Positive numbers - [unique: dataset]: Code is unique within the table, but may temporarily be non-unique during update process
cebucocode	Code assigned to the municipality by CeBuCo ('Centraal Bureau voor Courantenpubliciteit'), currently administered by the organization MOA. - [type: int]: Integer number - [range: [1, 9999]]: Positive numbers
provinciecode	Postcode.nl identifier for 'provincie' province records. - [type: text]: UTF-8 encoded string value - [length: 1]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [foreign key]: Foreign key identifier for 'provincie' province records

7.6 Province fields (“provincies”)

The “provincies” table contains fields specific to provinces in the Netherlands. Please note that this table is static and does not undergo mutations.

Field	Description
provinciecode	Postcode.nl identifier for 'provincie' province records. - [type: text]: UTF-8 encoded string value - [length: 1]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [primary key]: Unique identifier for 'provincie' province records
provincienaam	Name of the province, in Dutch. - [type: text]: UTF-8 encoded string value - [length: 20]: Maximum UTF-8 grapheme length - [subset: ASCII]: Ensures conversion to ASCII is always possible - [unique: dataset]: Value is unique within the dataset, but may temporarily be non-unique during update process

8 Using the Database

8.1 Applying Mutation Records

In our database, mutation records are applied differently depending on the format:

- **(Applied) Method**: This format provides SQL commands for updating, inserting, and deleting records directly.
- **Separate Mutation Records**: If the database format doesn't support the "(Applied)" method, mutations are handled as separate records: "insert", "delete", "update", and "downdate". These are stored in mutation tables prefixed with "mut_".

When applying these mutation records to the database tables, the following logic is followed:

1. **Insert and Update Records**:
    - Records are inserted and updated in descending table order, starting with "gemeente" (municipality), then "plaats" (city), "straat" (street), "pcreeks" (postcode range), and finally "perceel" (parcel).
    - Optionally, a matching "downdate" record can be used to validate the old record being updated.
2. **Delete Records**:
    - Records are deleted in ascending table order, starting with "perceel", then "pcreeks", "straat", "plaats", and finally "gemeente".

During the mutation process, it's important to note that some uniqueness constraints, such as unique street names in a locality and postcode range uniqueness, may be temporarily broken. Additionally, mutations can be inverted by swapping "insert" with "delete" and "update" with "downdate".

8.2 Dutch Postcode System

The Dutch postal code system ensures that a combination of postcode, house number, and house number addition uniquely defines a single postal address, simplifying address validation and enrichment using just the postcode and house number combination.

Dutch term	Description
postcode	A combination of 4 numbers (Dutch "wijkcode") and 2 letters (Dutch "lettercombinatie") used to identify an address.
wijkcode	The 4-digit number part of a postcode (ranging from 1000 to 9999).
lettercombinatie	The 2-letter part of a postcode, always in capital letters (from "AA" to "ZZ").
postcode reeks	A postcode range representing even or odd addresses in the same street with the same postcode. E.g. "2012 ES 2-30".
postbus	A Dutch PO Box, using "Postbus" as its street name. Only postcode range information is available for PO Box addresses.
perceel	Represents a single address, derived from the old postage system.
huisnummer	The numerical part of a street house identification (ranging from 1 to 99999).
huisnummertoevoeging	Additional non-numeric part of a street house identification, can include a space, up to 6 characters.

Address Formatting:

Official Dutch addresses are formatted as follows:

```
straatnaam huisnummer (huisnummer_toevoeging)
wijkcode lettercombinatie  plaatsnaam
```

For example:

```
Zijlstraat 41 B
2011 TK  HAARLEM
```

The format requires a space between the wijkcode and lettercombinatie in the postcode, and two spaces between the lettercombinatie and plaatsnaam elements. This formatting convention dates back to the era of typewriters when strict formatting was crucial due to limited automated address reading capabilities. Additionally, the "NEN" and "PostNL" often abbreviated street and city names due to storage or display size limitations.

Modern Dutch address formatting is as follows:

Zijlstraat 41B
2011TK Haarlem

This format eliminates the extra spaces around the postcode and uses normal capitalization for the city name. UTF-8 encoded street and city names are also permitted. House number additions can be written with either a space prefix or directly attached if the addition starts with a letter.

8.3 Validating Addresses Using Postcode Range “pcreeks” Table

To validate an address using only its postcode and house number, match the "wijkcode" and "lettercombination" fields to the postcode. Filter on the house number being within the range defined by "huisnr_van" and "huisnr_tm" and ensure that the house number matches the oddness as stored in "reeksindicatie".

See below for a simplified SQL example for validating the address “2012ES 30” using a normalized database, but for a merged database the query is the same but without the joins:

```
SELECT
      straat.straatnaam,
      plaats.plaatsnaam,
      gemeente.gemeentenaam
FROM
      pcreeks
      JOIN straat ON (pcreeks.straatid = straat.straatid)
      JOIN plaats ON (pcreeks.plaatsid = plaats.plaatsid)
      JOIN gemeente ON (pcreeks.gemeenteid = gemeente.gemeenteid)
WHERE
      pcreeks.wijkcode = '2012'
      AND pcreeks.lettercombinatie = 'ES'
      AND pcreeks.huisnr_van <= 30
      AND pcreeks.huisnr_tm >= 30
```

```
        AND pcreeks.reeksindicatie = IF(30 % 2, '0', '1')
```

8.4 Validating Addresses Using Address “perceel” Table

In addition to matching the fields as described in the "postcode range" validation section, we also need to verify the house number itself and its optional house number addition.

See below for a simplified SQL example for validating the address “2012ES 30A” using a normalized database, but for a merged database the query is the same but without the joins:

```
SELECT
      perceel.latitude,
      perceel.longitude,
      straat.straatnaam,
      plaats.plaatsnaam,
      gemeente.gemeentenaam
FROM
      perceel
      JOIN pcreeks ON (pcreeks.reeksid = perceel.reeksid)
      JOIN straat ON (pcreeks.straatid = straat.straatid)
      JOIN plaats ON (pcreeks.plaatsid = plaats.plaatsid)
      JOIN gemeente ON (pcreeks.gemeenteid = gemeente.gemeenteid)
WHERE
      pcreeks.wijkcode = '2012'
      AND pcreeks.lettercombinatie = 'ES'
      AND pcreeks.huisnr_van <= 30
      AND pcreeks.huisnr_tm >= 30
      AND pcreeks.reeksindicatie = IF(30 % 2, '0', '1')
      AND perceel.huisnummer = 30
      AND perceel.huisnummer_toevoeging = 'A'
```

Note that PO Box addresses ("postbussen") lack "perceel" level information and can only be validated at the "postcode range" level.

Bibliography

Data-API: Data API specification (https://www.postcode.eu/specifications/PC-Data-API-Spec.pdf)
PCT-BAG: PCT-BAG Specification, in Dutch (https://postcode.eu/specifications/PCT-BAG-Spec.pdf)