

Case Gemeenten



- Gebaseerd op postcodebestand van Bpost
http://www.bpost2.be/zipcodes/files/zipcodes_alpha_nl.xls

Case Gemeenten: creëer tabel



```
create table Gemeenten  
(  
    Postcode          int,  
    Plaatsnaam        varchar(200),  
    Deelgemeente      varchar(10),  
    Hoofdgemeente     varchar(200),  
    Provincie         varchar(200)  
)
```

Case Gemeenten: inlezen gegevens



```
BULK INSERT Gemeenten
FROM 'E:\Hfdst 05 MS SQL Server - Case Gemeenten - data.csv'
WITH
(
    CODEPAGE = 'ACP', -- correct importeren van accenten
    firstrow = 2, -- eerste rij zijn kolomnamen
    fieldterminator = ';', -- velden gescheiden door ;
    rowterminator = '\n' -- rijen gescheiden door NewLine teken
)
```

Case Gemeenten: select *

- Selecteer alle gegevens uit de tabel, om voeling te krijgen met de data

Select * from Gemeenten

Postcode	Plaatsnaam	Deelgemeente	Hoofdgemeente	Provincie
1000	BRUSSEL	Neen	BRUSSEL	Brussel (19 gemeenten)
1020	Laken	Ja	BRUSSEL	Brussel (19 gemeenten)
1030	SCHAARBEEK	Neen	SCHAARBEEK	Brussel (19 gemeenten)
1040	ETTERBEEK	Neen	ETTERBEEK	Brussel (19 gemeenten)
1050	ELSENE	Neen	ELSENE	Brussel (19 gemeenten)
1060	SINT-GILLIS	Neen	SINT-GILLIS	Brussel (19 gemeenten)
1070	ANDERLECHT	Neen	ANDERLECHT	Brussel (19 gemeenten)
1080	SINT-JANS-MOLENBEEK	Neen	SINT-JANS-MOLENBEEK	Brussel (19 gemeenten)
1081	KOEKELBERG	Neen	KOEKELBERG	Brussel (19 gemeenten)
1082	SINT-AGATHA-BERCHEM	Neen	SINT-AGATHA-BERCHEM	Brussel (19 gemeenten)
1083	GANSHOREN	Neen	GANSHOREN	Brussel (19 gemeenten)
1090	JETTE	Neen	JETTE	Brussel (19 gemeenten)
1120	Neder-Over-Heembeek	Ja	BRUSSEL	Brussel (19 gemeenten)
1130	Haren	Ja	BRUSSEL	Brussel (19 gemeenten)
1140	EVERE	Neen	EVERE	Brussel (19 gemeenten)
1150	SINT-PIETERS-WOLUWE	Neen	SINT-PIETERS-WOLUWE	Brussel (19 gemeenten)
1160	OUDEGEM	Neen	OUDEGEM	Brussel (19 gemeenten)
1170	WATERMAAL-BOSVOORDE	Neen	WATERMAAL-BOSVOORDE	Brussel (19 gemeenten)
1180	UKKEL	Neen	UKKEL	Brussel (19 gemeenten)
1190	VORST	Neen	VORST	Brussel (19 gemeenten)

Case Gemeenten: telling per provincie



- Tel het aantal gemeenten per provincie
Schrijf het SQL-statement

Case Gemeenten: telling per provincie

- Tel het aantal gemeenten per provincie

```
select
    count(*) aantalGemeenten,
    provincie
from Gemeenten
group by provincie
order by provincie
```

161	Antwerpen
22	Brussel (19 gemeenten)
468	Henegouwen
220	Limburg
374	Luik
239	Luxemburg
375	Namen
309	Oost-Vlaanderen
239	Vlaams-Brabant
121	Waals-Brabant
253	West-Vlaanderen

Case Gemeenten: gemiddelde postcode per provincie



- Geef per provincie de laagste en hoogste postcode
Schrijf het SQL-statement

Case Gemeenten:

gemiddelde postcode per provincie

- Geef per provincie de laagste en hoogste postcode

```
select
  min(postcode) laagstePostcode,
  max(postcode) hoogstePostcode,
  provincie
from Gemeenten
group by provincie
order by provincie
```

2000	2990	Antwerpen
1000	1210	Brussel (19 gemeenten)
6000	7973	Henegouwen
3500	3990	Limburg
4000	4990	Luik
6600	6997	Luxemburg
5000	5680	Namen
9000	9992	Oost-Vlaanderen
1500	3473	Vlaams-Brabant
1300	1495	Waals-Brabant
8000	8980	West-Vlaanderen

Case Gemeenten: toevoegen kolommen



- Voeg 2 kolommen toe:
 - ▣ Id (*int*) *primary key, autonumber*
 - ▣ IsDeelgemeente (*bit*) \rightarrow *true/false*
- Schrijf het SQL-statement

Case Gemeenten: toevoegen kolommen

- Voeg 2 kolommen toe:

- ▣ Id (*int*) *primary key, autonumber*

- ▣ IsDeelgemeente (*bit*) \rightarrow *true/false*

Alter table Gemeenten

Add

Id int IDENTITY(1, 1) PRIMARY KEY,
IsDeelgemeente bit

Case Gemeenten: update kolom

- Update de kolom 'IsDeelgemeente'
 - ▣ Zet deze op true (=1) als kolom DeelGemeente = 'Ja'
 - ▣ Zet deze op false (=0) als kolom DeelGemeente = 'Neen'

→ Doe dit in één SQL-statement

Case Gemeenten: update kolom

- Update de kolom 'IsDeelgemeente'
 - ▣ Zet deze op true (=1) als kolom DeelGemeente = 'Ja'
 - ▣ Zet deze op false (=0) als kolom DeelGemeente = 'Neen'

```
update Gemeenten
set IsDeelgemeente =
    case Deelgemeente
        when 'Ja' then 1
        else 0
    end
```

Case Gemeenten: delete kolom

- De kolom 'DeelGemeente' is nu vervangen door een kolom met een passender dataType: IsDeelgemeente
- Verwijder de kolom 'Deelgemeente'
Schrijf het SQL-statement

Case Gemeenten: delete kolom

- De kolom 'DeelGemeente' is nu vervangen door een kolom met een passender dataType: IsDeelgemeente
- Verwijder de kolom 'Deelgemeente'

```
alter table Gemeenten  
drop column Deelgemeente
```

Case Gemeenten: provincie afsplitsen



- In de volgende stappen gaan we de provincienaam afsplitsen in een aparte tabel.
- De tabel 'Gemeenten' gaat een referentiële sleutel hebben naar de nieuwe tabel 'Provincies'

Case Gemeenten: provincie afsplitsen



- Creëer een tabel 'Provincies' met 2 velden
 - ▣ Id (*int*) *primary key, autonumber*
 - ▣ Naam (*varchar(200)*)

- Schrijf het SQL-statement

Case Gemeenten: provincie afsplitsen

- Creëer een tabel 'Provincies' met 2 velden
 - ▣ Provinciëld (*int*) *primary key, autonumber*
 - ▣ Naam (*varchar(200)*)

```
create table Provincies
(
    ProvincieId int IDENTITY(1, 1) PRIMARY KEY,
    Naam varchar(200)
)
```

Case Gemeenten: opvullen provincies



- We gaan onze nieuwe tabel opvullen met alle provincienamen die we vinden in de tabel 'Gemeenten'
- Schrijf het SQL-statement

Case Gemeenten: opvullen provincies



- We gaan onze nieuwe tabel opvullen met alle provincienamen die we vinden in de tabel 'Gemeenten'

```
insert into provincies (naam)
select distinct provincie from Gemeenten
order by provincie
```

Case Gemeenten: nieuwe kolom



- In de tabel 'Gemeenten' gaan we een nieuw veld creëren, Provinciaal.
- Dit veld wordt onze verwijzing naar de Id in de tabel 'Provincies'
- We gaan nog geen referentie leggen, we voegen alleen het veld toe
- Schrijf het SQL-statement

Case Gemeenten: nieuwe kolom

- In de tabel 'Gemeenten' gaan we een nieuw veld creëren, ProvinciaId.
- Dit veld wordt onze verwijzing naar de Id in de tabel 'Provincies'

```
alter table Gemeenten  
add ProvinciaId int
```

Case Gemeenten: nieuwe kolom invullen



- We schrijven een UPDATE-statement om deze nieuwe kolom op te vullen met de overeenkomstige waarde van de Id uit de tabel 'Provincies'

Case Gemeenten: nieuwe kolom invullen

- We schrijven een UPDATE-statement om deze nieuwe kolom op te vullen met de overeenkomstige waarde van de Id uit de tabel 'Provincies'

```
update G
      set G.ProvincieId = P.ProvincieId
from
Gemeenten G inner join Provincies P
on G.Provincie = P.Naam
```

Case Gemeenten: nieuwe kolom invullen

- We hadden dit ook met een CASE statement kunnen doen, maar dat is 'minder mooi'.

Bovendien hebben we telkens de waarden (1, 2, ... 11) moeten opzoeken in de tabel Provincies om dit statement te kunnen schrijven.

```
update gemeenten
set provincieId =
case Provincie
  when 'Antwerpen' then 1
  when 'Brussel' then 2
  when 'Henegouwen' then 3
  when 'Limburg' then 4
  when 'Luik' then 5
  when 'Luxemburg' then 6
  when 'Namen' then 7
  when 'Oost-Vlaanderen' then 8
  when 'Vlaams-Brabant' then 9
  when 'Waals-Brabant' then 10
  when 'West-Vlaanderen' then 11
end
```


Case Gemeenten: referentiële integriteit



- Als alles goed gegaan is, is de kolom Provinciaal nu correct ingevuld.

Dan kunnen we nu de foreign key leggen tussen Provincie.Provinciaal en Gemeente.Provinciaal

- Schrijf het SQL-statement hiervoor

Case Gemeenten: referentiële integriteit

- Als alles goed gegaan is, is de kolom ProvincieId nu correct ingevuld.

Dan kunnen we nu de foreign key leggen tussen Provincie.Id en Gemeente.ProvincieId

```
ALTER TABLE Gemeenten  
ADD CONSTRAINT FK_PROVINCIEID FOREIGN KEY (ProvincieId)  
REFERENCES Provincies(ProvincieId)
```

Case Gemeenten: referentiële integriteit

- Controleer dat je foreign key effectief werkt!

Case Gemeenten: referentiële integriteit

- Controleer dat je foreign key effectief werkt!

```
update gemeenten  
set provincieId = 3000 --onbestaande waarde  
where provincieId = 2
```

Msg 547, Level 16, State 0, Line 14
The UPDATE statement conflicted with the FOREIGN KEY constraint
"FK_PROVINCIEID". The conflict occurred in database "GEMEENTEN", table
"dbo.Provincies", column 'Id'.
The statement has been terminated.



Case Gemeenten: verwijder kolom

- De kolom 'Provincie' in de tabel Gemeenten is nu overbodig. Drop it!
- Drop ook de kolom 'HoofdGemeente'

Case Gemeenten: verwijder kolom

- De kolom 'Provincie' in de tabel Gemeenten is nu overbodig. Drop it!
- Drop ook de kolom 'HoofdGemeente'

```
alter table Gemeenten  
drop column Provincie, HoofdGemeente
```

Case Gemeenten

