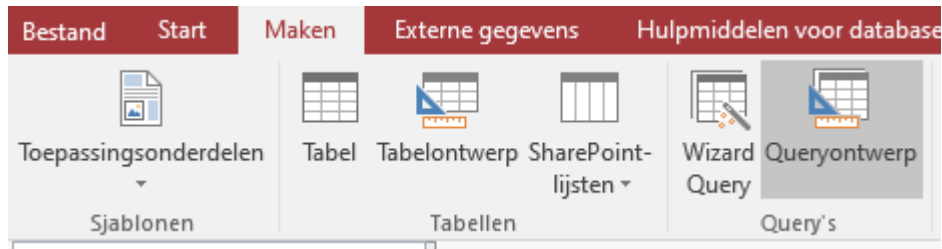


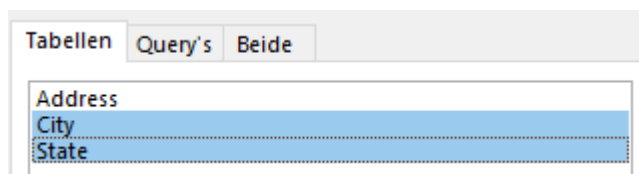
4 (Stored) procedures

Stored-procedures zijn SQL-opdrachten die opgeslagen zijn in de database. We gebruiken de relatie diagram zoals weergegeven in “Figuur 2: Relatie diagram”. We selecteren de optie Queryontwerp in Access.

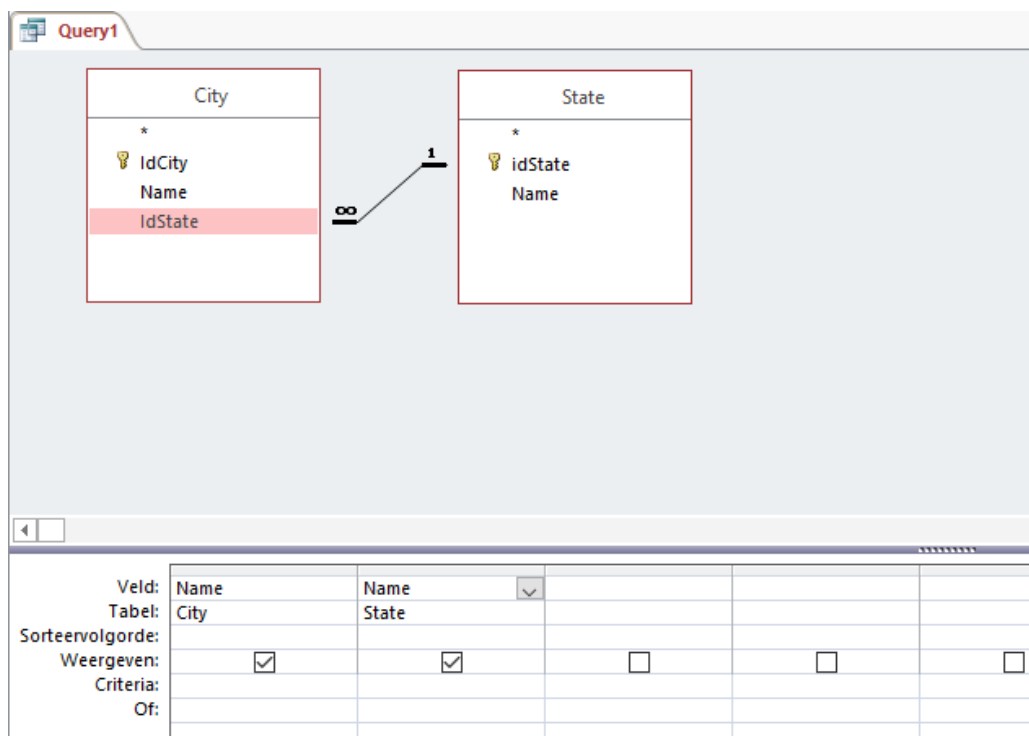


Figuur 3: Queryontwerp starten

We willen een instructie maken van de combinatie met de tabellen City en State. In het scherm dat naar voren komt selecteren we beide namen en drukken op {Toevoegen}.

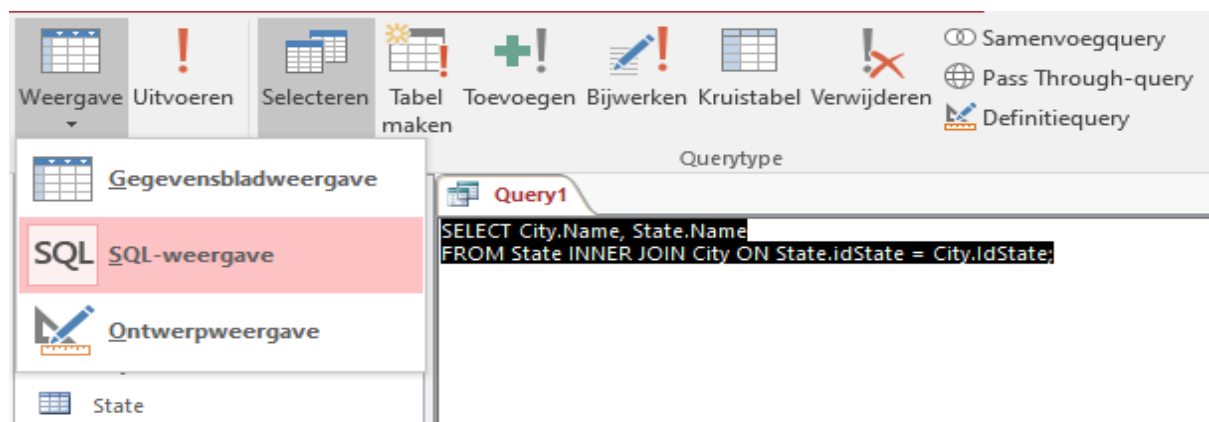


We krijgen dan de twee tabellen te zien, en dat ze een relatie met elkaar hebben. Vervolgend dubbelklikken we in de tabel City op Name, en in de tabel State ook op Name. Onder aan het venster zien we de twee velden verschijnen.



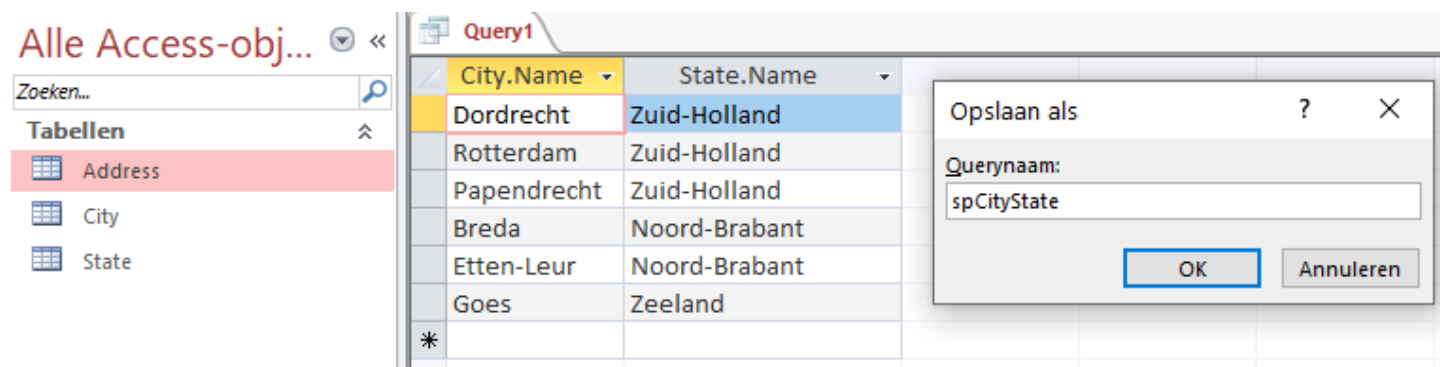
We kunnen nu op de knop “Weergave” kiezen dat we de SQL-Statement willen zien van wat we net hebben gemaakt. Als we op {uitvoeren} drukken zien we het resultaat van de code. Met een SQL-statement kan je de inhoud van tabellen bekijken. Je kan ook een combinatie van verschillende

tabellen bekijken zoals we nu hebben gezien. Als we nu de SQL-Statement opslaan kunnen we deze later nogmaals gebruiken. Zo wordt het een opgeslagen procedure oftewel een *stored-procedure*.



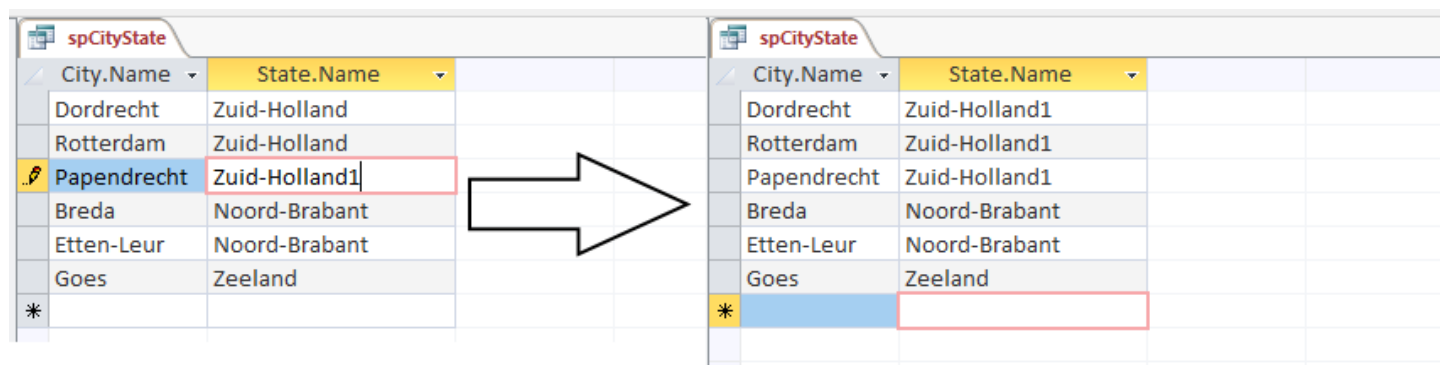
Figuur 4: SQL-presentatie stored-procedure

Met <CTRL><s> kunnen we de stored-procedure opslaan. Voor de naam beginnen we altijd met het voorvoegsel {sp} met daarachter in camel case een beschrijving voor het resultaat wat uit de functie komt. De stored-procedure is nu toegevoegd in de lijst van “Alle access-objecten”. Je kan nu op de stored-procedure klikken, en dan zie je het resultaat. Het verschil met een tabel is dat in een tabel echt de data zit die je ziet. Een stored-procedure is alleen de SQL-statement die de data bij elkaar verzameld en dan laat zien.



Figuur 5: Opslaan van een stored-procedure

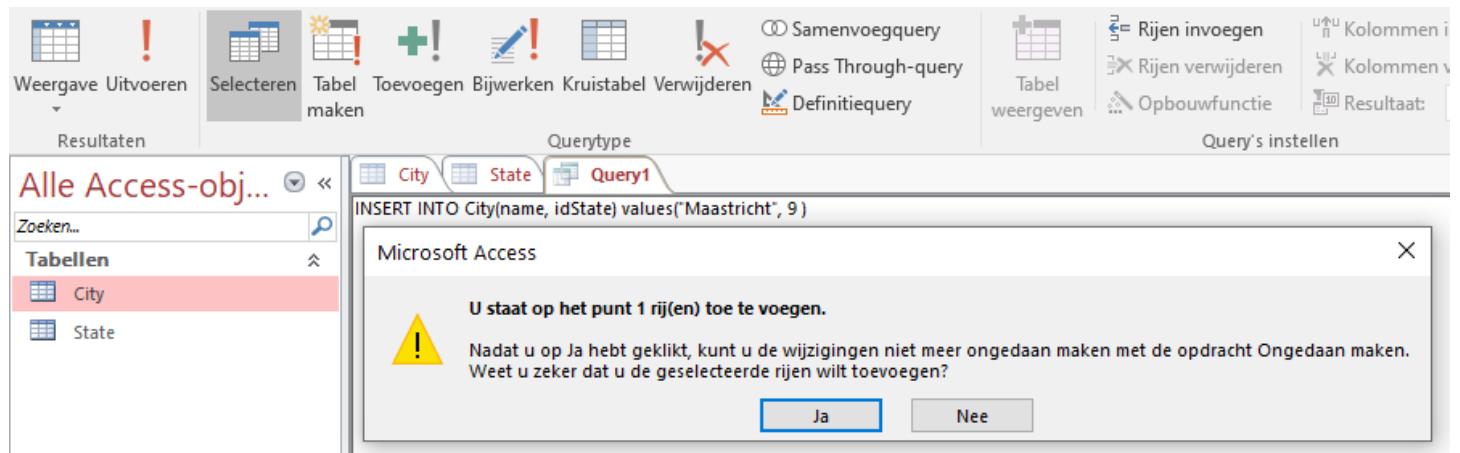
We kunnen in Access we de waarde van een veld nu veranderen. Maar als we bijvoorbeeld de naam Zuid-Holland wijzigen dan zien we dan gelijk alle velden waar Zuid-Holland gewijzigd zijn. Dit is de kracht van een database. Door de wijziging op het ene veld, hebben we de waarde in de State-tabel gewijzigd.



4.1 Data toevoegen door middel van SQL

SQL is er niet alleen om data mee uit te lezen, maar je kunt er ook data mee invoeren.

Bij het invoeren is het altijd erg belangrijk dat de regels die je invoert correct zijn. Ze moeten voldoen aan de definities die zijn gemaakt in het database ontwerp.



Nadat we op uitvoeren hebben gedrukt zal de rij toegevoegd worden aan de tabel.

Omdat idCity een autonummer veld is, hoeven we die waarde niet mee te geven. Als we dezelfde SQL-opdracht nog eens uitvoeren, maar nu niet met de waarde 9, maar met 99 dan krijgen we de volgende melding. idState=99 is niet bekend, en daarom kan de regel niet ingevoerd worden. Dit is om de database correct te houden (database integrity).



4.2 Data wijzigen met SQL

Het volgende statement geeft een goede nieuwe stad in de database.

```
INSERT INTO City ( name, idState ) VALUES ("Eindhoven", 1);
```

Als we alleen naar de database gaan kijken zien we dat Eindhoven in Zuid-Holland ligt.

	2	Rotterdam	Zuid-Holland
	11	Eindhoven	Zuid-Holland
	4	Amsterdam	Noord-Holland
	3	Breda	Noord-Brabant

Nu kunnen we deze regel verwijderen en opnieuw invoeren, maar we kunnen hem ook via een SQL-statement wijzigen. Dit doen we met het update commando.

```
update City set idState = 3 where idCity = 11
```

We zien dat we de update niet op naam doen. Het kan zijn dat er meerdere Eindhoven zijn. Voor Eindhoven is dat niet het geval, maar de plaats Elst komt in twee provincies voor. idCity is daarentegen altijd uniek, want dat is dat we bij het maken van de tabel hebben ingesteld dat dit een unieke sleutel is.

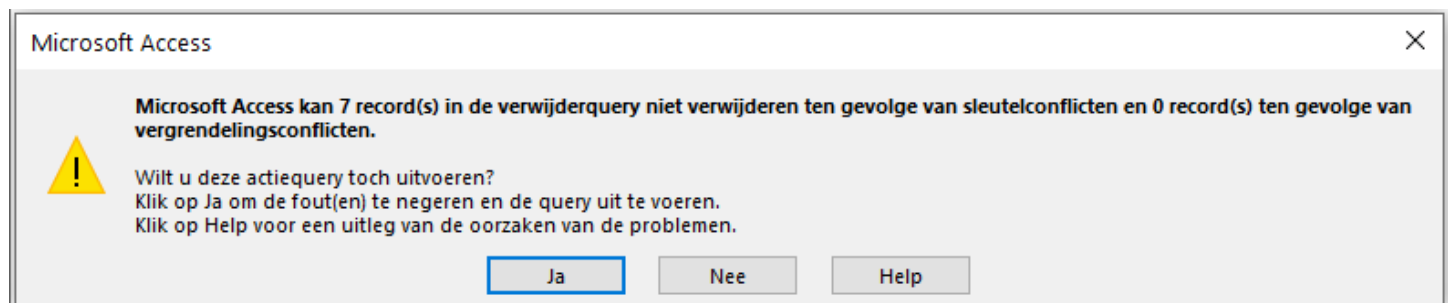
De regel lezen we als volgt. UPDATE {tabel} SET {wijziging} WHERE {voorwaarden}.

Als we geen voorwaarden invoeren, dan wijzigen we alle records. Op deze manier kunnen we ook een reeks rijen wijzigen die allemaal eenzelfde veld hebben.

4.3 Data verwijderen met SQL

Door middel van het DELETE-commando kunnen we elementen uit de database halen.

DELETE * from State



State			
City			
Query1			
	idState	name	klik om titel toe te voegen
+		Zuid-Holland	
+	2	Noord-Holland	
+	3	Noord-Brabant	
+	#Verwijderd	#Verwijderd	
+	5	Utrecht	
+	#Verwijderd	#Verwijderd	
+	#Verwijderd	#Verwijderd	
+	#Verwijderd	#Verwijderd	
+	9	Limburg	
+	10	Groningen	
+	11	Friesland	
+	#Verwijderd	#Verwijderd	
*	(Nieuw)		

Alleen de provincies die niet aan een stad zijn gekoppeld zijn nu verwijderd. Dit komt door de relatie die we hebben gemaakt tussen idState in de State tabel met idState in de City tabel. We kunnen de andere elementen pas weghalen als we de relaties niet meer gebruiken.

Hadden we maar 1 provincie willen verwijderen, dan hadden we dit in de delete-opdracht moeten toevoegen.

Delete * from State where idState = 4

4.4 Opdrachten

4.4.1 Verwijderen Zuid-Holland

Als we Zuid-Holland willen verwijderen uit de State tabel, dan zegt de database dat dat niet kan, Kan je verklaren waarom dat niet zou mogen!