

Softwareontwikkeling 6IT – preteaching programma april 2020

Leerstof – Databases met MySQL : een database-schema bouwen

Behandelt aspecten van leerplandoelstelling 19-26-28

TIJDLIJN

Voor deze opdracht worden **5 lestijden** voorzien.

maandag 20/04/2020 : **Alle leerlingen om 8u30 aanwezig in de Teams-chat.** Er wordt uitleg gegeven over het verloop van de week. Tot 10u00 kunnen er via de chat vragen worden gesteld.

Dinsdag 21/04/2020 : mogelijkheid om via video-chat vragen stellen van 13u00 tot 15u00

Woensdag 24/04/2020 : **Alle leerlingen om 10u00 aanwezig in de Teams-chat.** Tussentijdse bespreking van de opdrachten. Tot 12u00 kunnen er vragen worden gesteld.

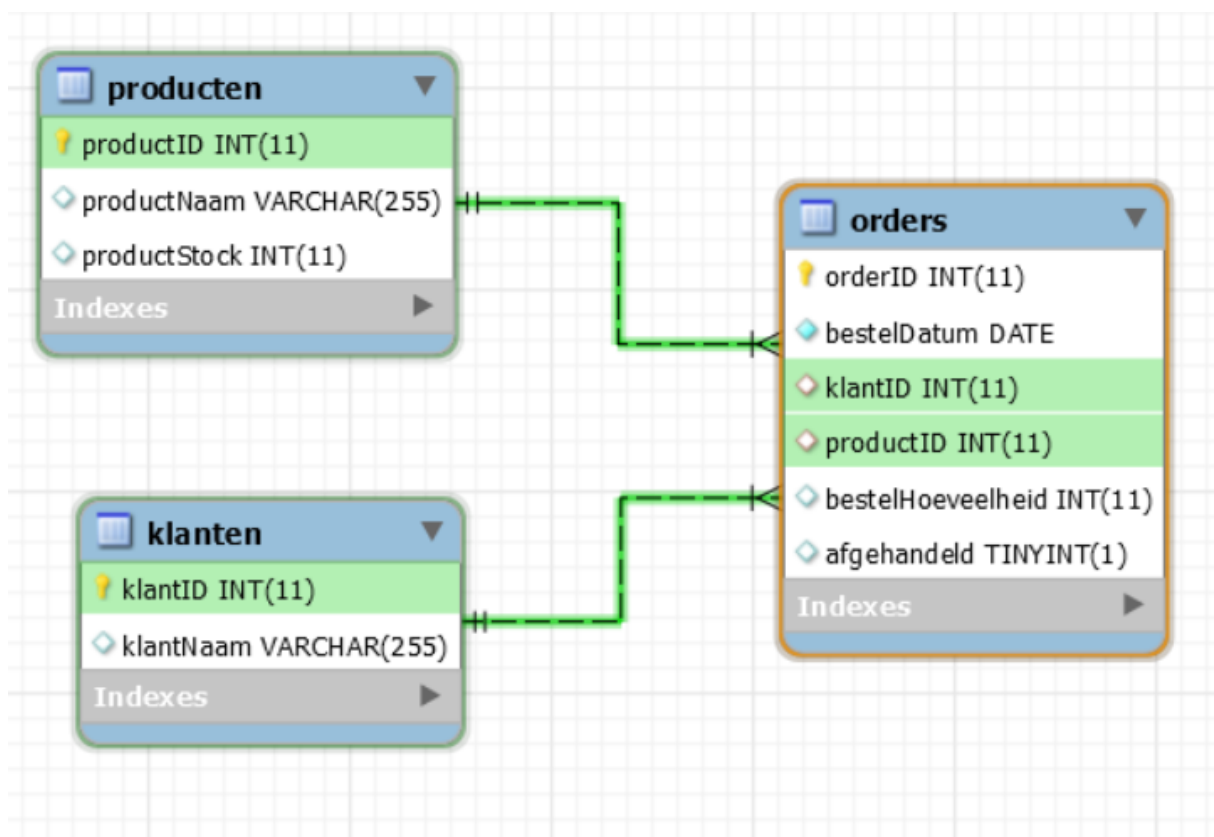
Vrijdag 24/04/2020 : **Alle leerlingen om 8u30 aanwezig in de Teams-chat.** De voorbije week wordt besproken. Tot 10u00 kunnen er via de chat vragen worden gesteld.

GEGEVEN

Een onderneming verkoopt producten aan klanten. Ze wil deze gegevens bijhouden in een MySQL-database. De database bevat volgend tabellen:

- klanten
- producten
- orders

Het schema met de relaties ziet er uit zoals in onderstaande afbeelding.



GEVRAAGD

Het gegeven schema moet met behulp van MySQL Workbench worden opgebouwd. Je zal drie soorten query's moeten schrijven:

- Query's om het schema te creëren
- Query's om gegevens in de database in te voeren
- Query's om gegevens uit de database te halen

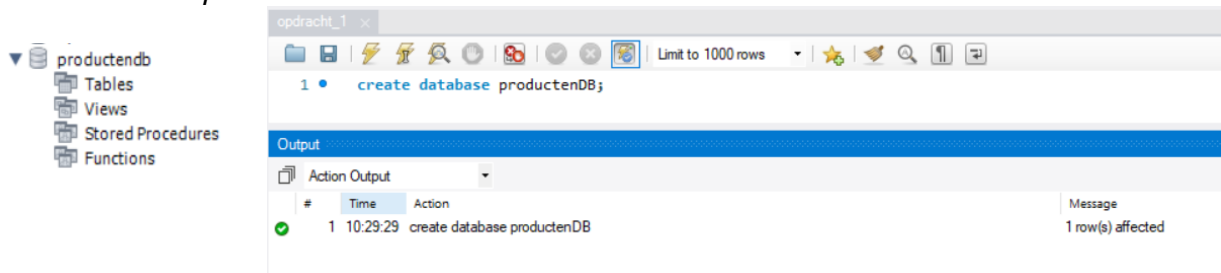
In de volgende opdrachten schrijf je stap voor stap deze query's. Wanneer iets niet duidelijk is dan kan je terecht op het [YouTube-kanaal van deze lessen](#).

Heel wat opdrachten zijn gemakkelijk uit te voeren. Sommige zijn wat lastiger.

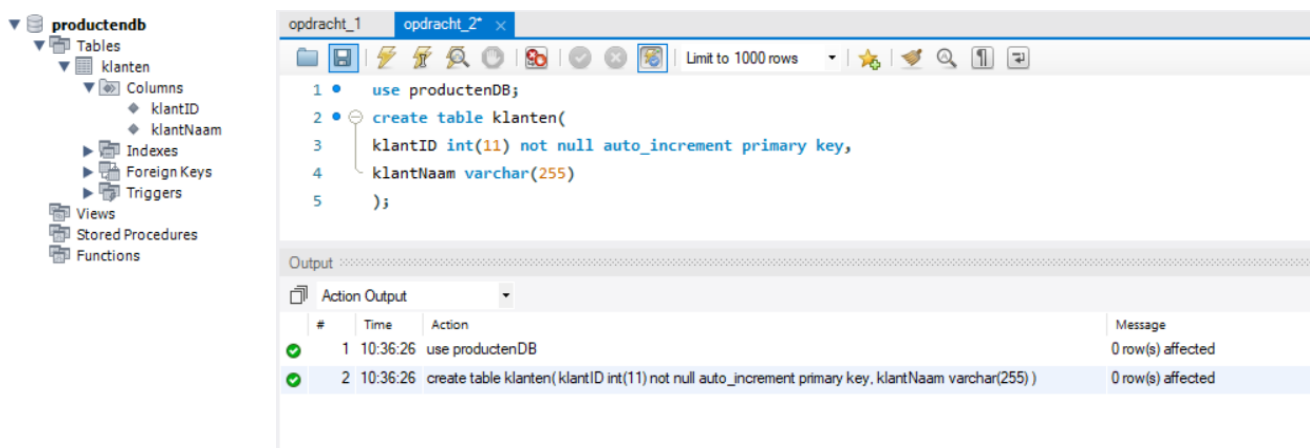
Werkt nauwgezet en zorg dat alle opdrachten uitgevoerd zijn en werken. Je hebt dit schema en alle query's nodig in een vervolgoopdracht.

Je maakt een digitaal verslag van jouw werk. **Dit verslag bevat alle query's samen met een screenshot van het resultaat van de query.** Het bestand wordt in pdf-formaat ingediend via de Teams-opdracht.

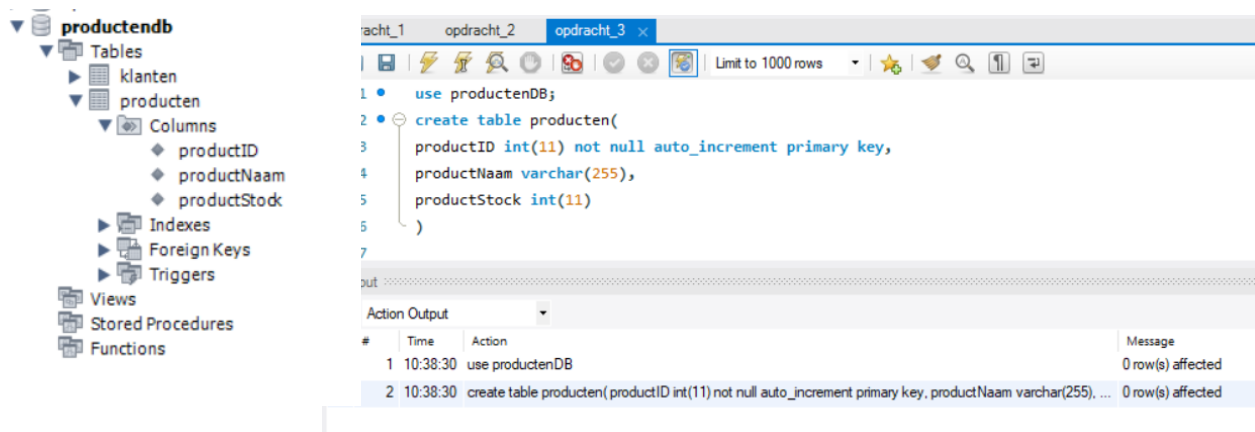
1. Schrijf de query om de database aan te maken. Noem deze database 'productenDB'.



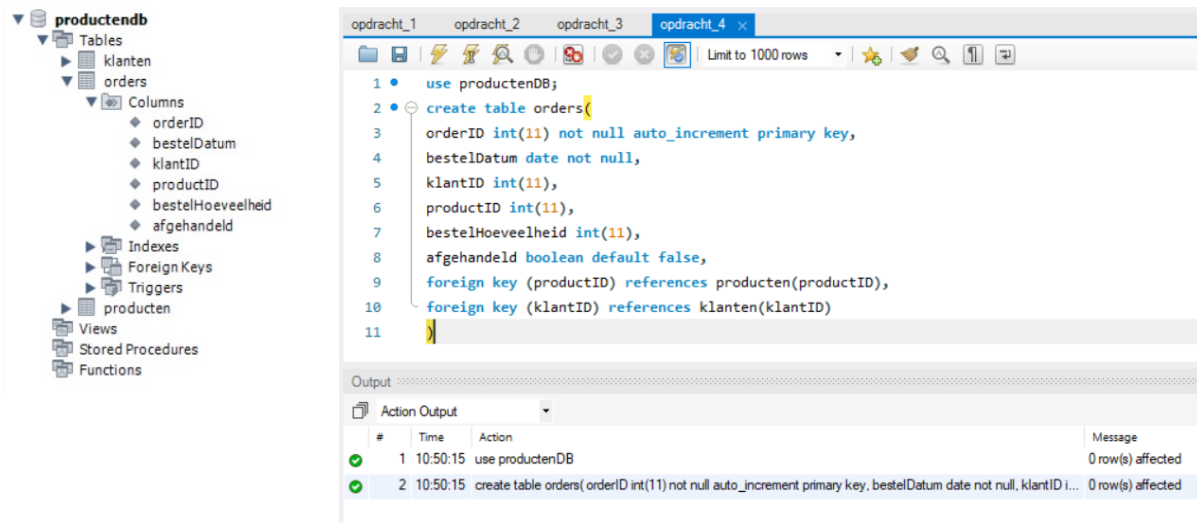
2. Schrijf de query om de tabel 'klanten' aan te maken.



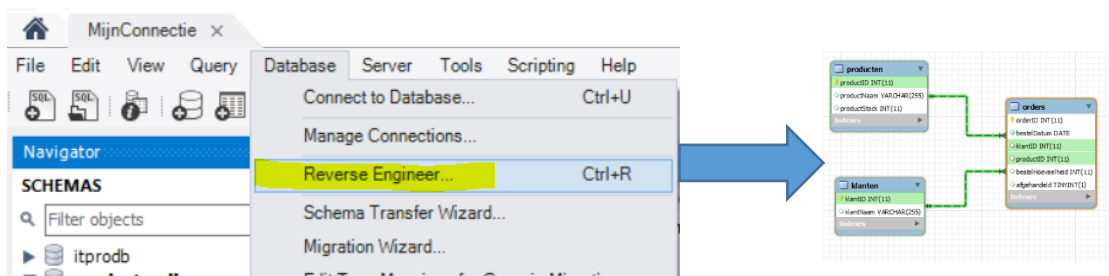
3. Schrijf de query om de tabel 'producten' aan te maken.



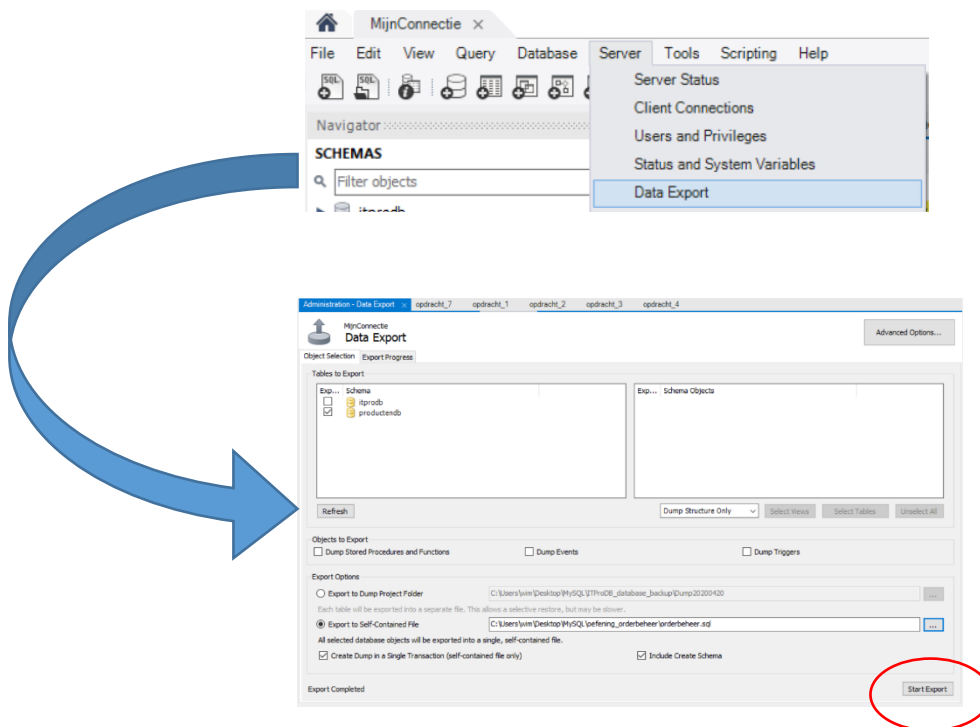
4. Schrijf de query om de tabel 'orders' aan te maken.



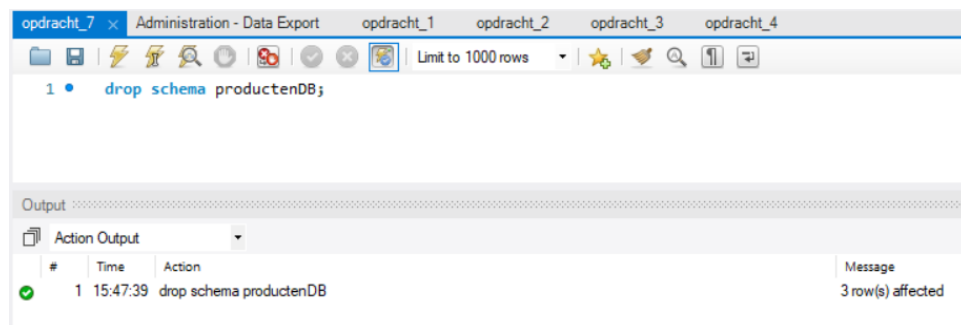
5. Controleer of het schema correct is opgebouwd.



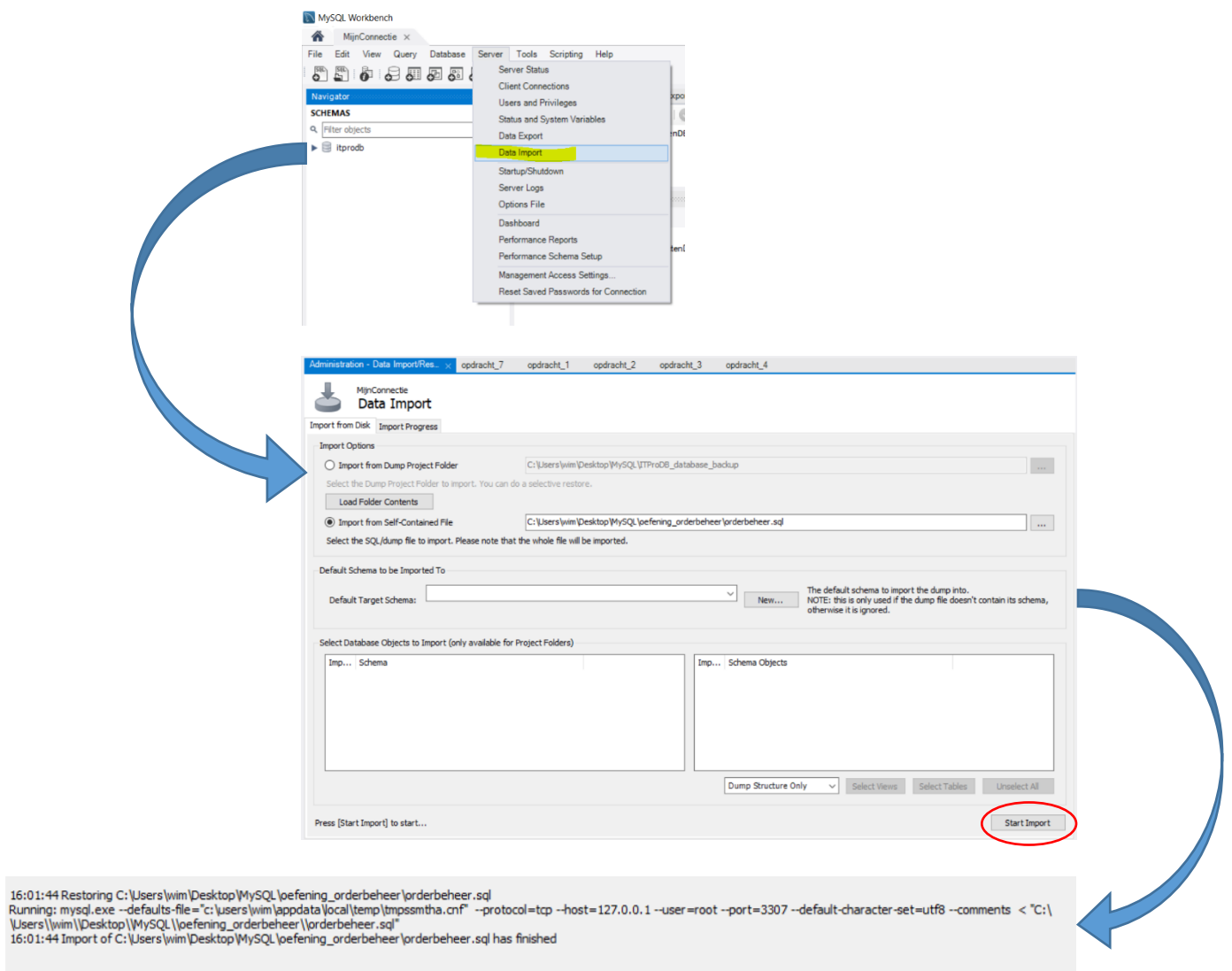
6. Exporteer de database naar één op zichzelf staande query.



7. Verwijder het databaseschema *productenDB*.



8. Importeer het schema vanuit het in 6. aangemaakte bestand.



9. Kies de branche waarin je bedrijf actief is. Kies ook een bedrijfsnaam.

SafeMed bvba, Medical supplies

10. Schrijf een query waarmee je aan de producten-tabel acht producten toevoegt die specifiek zijn voor de gekozen branche. Bij elk product geef je ook het aantal in dat in stock is.

productNaam	productStock
3 laags chirurgisch wegwerp mondmaskers medium 25st/doos	500
3 laags chirurgisch wegwerp mondmaskers large 25 st/doos	500
FFP2 mondmaskers large 25 st/doos	30
FFP3 mondmaskers medium 2 st/verpakking	1000
Thermoscan 6000 Pro thermometer	20
Alcogel 70% + citrus bag-N-box 1000 ml	200
Nitrilhandschoenen blauw medium 100 st/doos	50
Nitrilhandschoenen blauw large 100 st/doos	40

```
1 use productenDB;
2 insert into producten(productNaam, productStock)
3 values
4 ('3 laags chirurgisch wegwerp mondmaskers medium 25st/doos',500),
5 ('3 laags chirurgisch wegwerp mondmaskers large 25 st/doos',500),
6 ('FFP2 mondmaskers large 25 st/doos',30),
7 ('FFP3 mondmaskers medium 2 st/verpakking',1000),
8 ('Thermoscan 6000 Pro thermometer',20),
9 ('Alcogel 70% + citrus bag-N-box 1000 ml',200),
10 ('Nitrilhandschoenen blauw medium 100 st/doos',50),
11 ('Nitrilhandschoenen blauw large 100 st/doos',40);
12
```

Output

#	Time	Action	Message
1	16:27:08	use productenDB	0 row(s) affected
2	16:27:08	insert into producten(productNaam, productStock) values ('3 laags chirurgisch wegwerp mondmaskers me...	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0

11. Schrijf een query waarmee je de volledige inhoud van de producten-tabel kan tonen.

```
1 use productenDB;
2 select * from producten;
```

Result Grid

	productID	productNaam	productStock
1		3 laags chirurgisch wegwerp mondmaskers medi...	500
2		3 laags chirurgisch wegwerp mondmaskers large...	500
3		FFP2 mondmaskers large 25 st/doos	30
4		FFP3 mondmaskers medium 2 st/verpakking	1000
5		Thermoscan 6000 Pro thermometer	20
6		Alcogel 70% + citrus bag-N-box 1000 ml	200
7		Nitrilhandschoenen blauw medium 100 st/doos	50
8		Nitrilhandschoenen blauw large 100 st/doos	40
*		NULL	NULL

producten 2

Output

#	Time	Action	Message
1	16:35:00	use productenDB	0 row(s) affected
2	16:35:00	select * from producten LIMIT 0, 1000	8 row(s) returned

12. Kies vijf klantnamen en schrijf een query waarmee je deze klanten toevoegt aan de klanten-tabel.

AZ Sint Blasius Dendermonde
Apotheek De Rijbel Aalst
Colruyt NV Halle
Departement Welzijn, Gezondheid en Gezin - Vlaamse Gemeenschap
Rode Kruis afdeling Oost-Vlaanderen

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL script:

```
1 use productenDB;
2 insert into klanten (klantNaam)
3 values
4 ("AZ Sint Blasius Dendermonde"),
5 ("Apotheek De Rijbel Aalst"),
6 ("Colruyt NV Halle"),
7 ("Departement Welzijn, Gezondheid en Gezin - Vlaamse Gemeenschap"),
8 ("Rode Kruis afdeling Oost-Vlaanderen");
9
```

The Output pane shows the execution results:

#	Time	Action	Message
1	16:45:12	use productenDB	0 row(s) affected
2	16:45:12	insert into klanten (klantNaam) values ("AZ Sint Blasius Dendermonde"), ("Apotheek De Rijbel Aalst"), ("C...", ("Departement Welzijn, Gezondheid en Gezin - Vlaamse Gemeenschap"), ("Rode Kruis afdeling Oost-Vlaanderen"))	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

13. Schrijf een query waarmee je de volledige inhoud van de klanten-tabel kan tonen.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL script:

```
1 use productenDB;
2 select * from klanten;
```

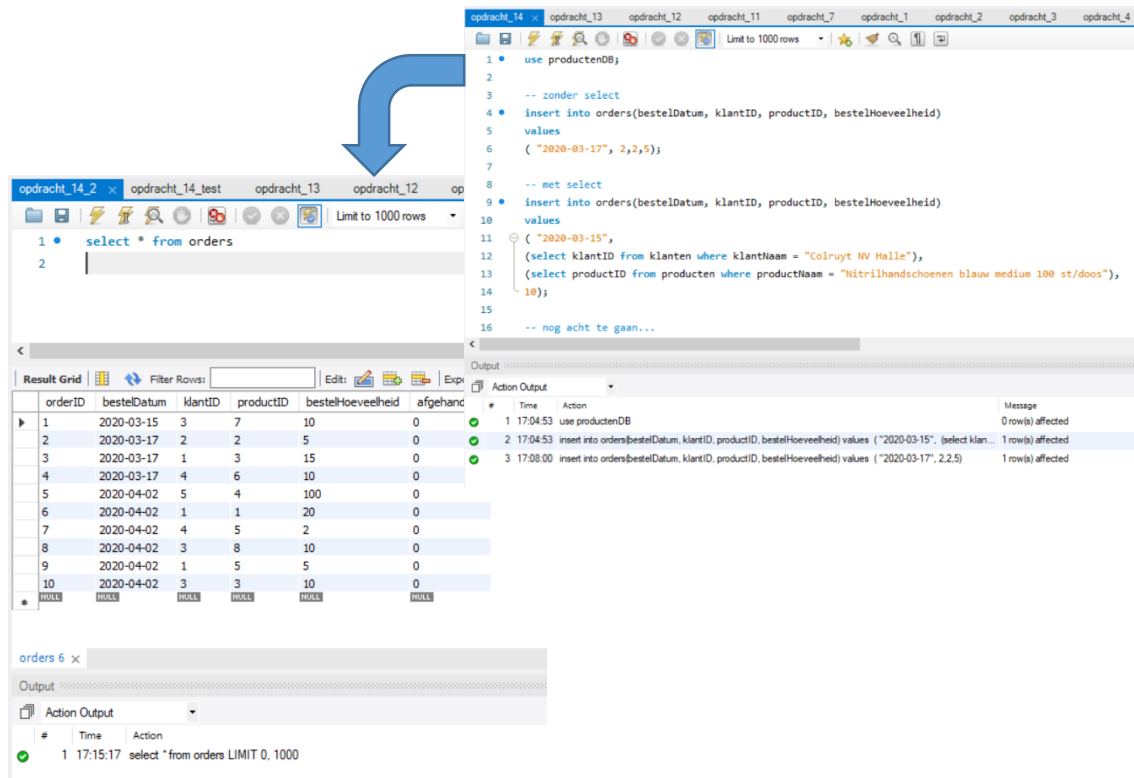
The Result Grid shows the data returned by the query:

klantID	klantNaam
1	AZ Sint Blasius Dendermonde
2	Apotheek De Rijbel Aalst
3	Colruyt NV Halle
4	Departement Welzijn, Gezondheid en Gezin - Vl...
5	Rode Kruis afdeling Oost-Vlaanderen
NULL	NULL

The Output pane shows the execution results:

#	Time	Action	Message
1	16:47:49	use productenDB	0 row(s) affected
2	16:47:49	select * from klanten LIMIT 0, 1000	5 row(s) returned

14. Schrijf een query waarmee je tien orders toevoegt aan de orders-tabel. Zorg dat er voor elke klant minstens één order is toegevoegd. Kies realistische datums. Bij de orders is er ook een boolean-attribuut *afgehandeld* voorzien. Die geeft aan of een bestelling is afgehandeld of niet. *afgehandeld*=false wil zeggen dat een order nog niet is afgewerkt.



The screenshot shows a database management tool with a SQL query editor and a results grid. The query editor contains the following SQL code:

```

1 use productenDB;
2
3 -- zonder select
4 insert into orders(bestelDatum, klantID, productID, bestelHoeveelheid)
5 values
6 ( "2020-03-17", 2,2,5);
7
8 -- met select
9 insert into orders(bestelDatum, klantID, productID, bestelHoeveelheid)
10 values
11 ( "2020-03-15",
12 (select klantID from klanten where klantNaam = "Colruyt NV Halle"),
13 (select productID from producten where productNaam = "Nitrilhandschoenen blauw medium 100 st/doos"),
14 10);
15
16 -- nog acht te gaan...

```

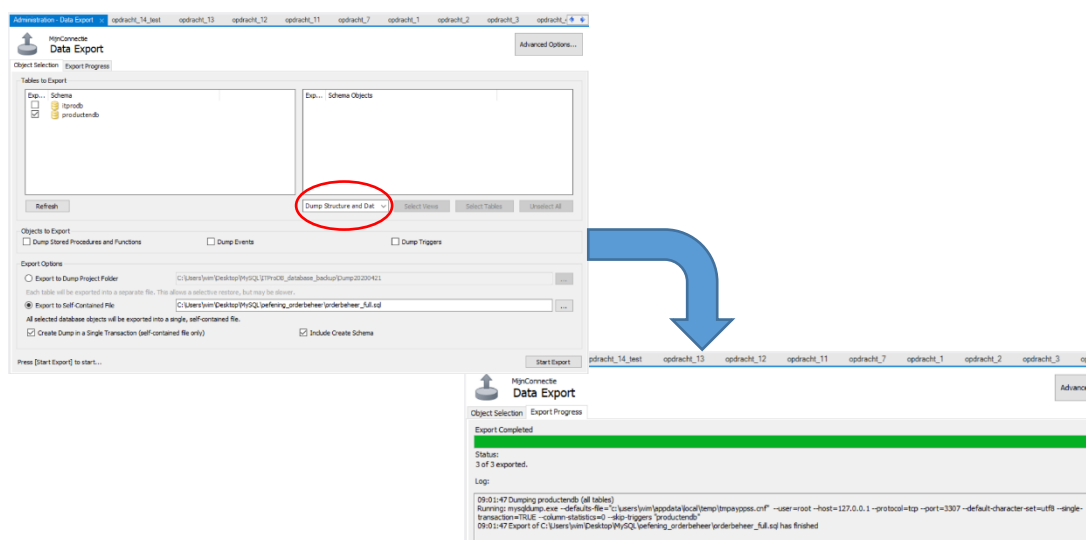
The results grid shows the following data:

orderID	bestelDatum	klantID	productID	bestelHoeveelheid	afgehandeld
1	2020-03-15	3	7	10	0
2	2020-03-17	2	2	5	0
3	2020-03-17	1	3	15	0
4	2020-03-17	4	6	10	0
5	2020-04-02	5	4	100	0
6	2020-04-02	1	1	20	0
7	2020-04-02	4	5	2	0
8	2020-04-02	3	8	10	0
9	2020-04-02	1	5	5	0
10	2020-04-02	3	3	10	0

The output pane shows the following actions:

#	Time	Action	Message
1	17:04:53	use productenDB	0 row(s) affected
2	17:04:53	insert into orders(bestelDatum, klantID, productID, bestelHoeveelheid) values ("2020-03-15", (select klantID from klanten where klantNaam = "Colruyt NV Halle"), (select productID from producten where productNaam = "Nitrilhandschoenen blauw medium 100 st/doos"), 10);	1 row(s) affected
3	17:08:00	insert into orders(bestelDatum, klantID, productID, bestelHoeveelheid) values ("2020-03-17", 2,2,5);	1 row(s) affected

15. Exporteer de database – inclusief inhoud - naar één op zichzelf staande query.

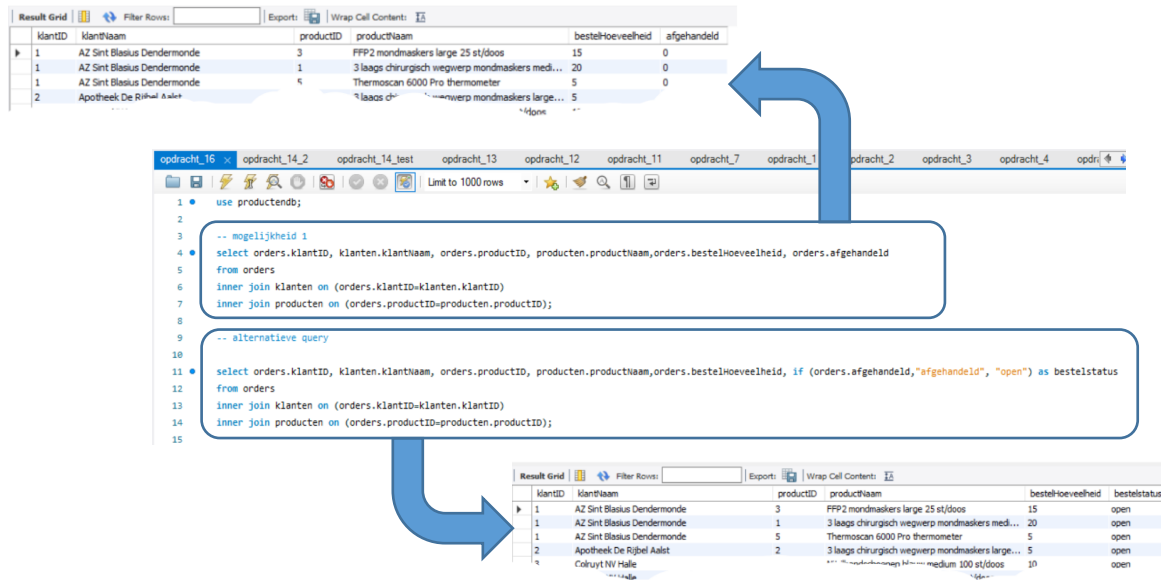


The screenshot shows the MySQL Data Export wizard. The 'Export Options' tab is selected, showing the following options:

- ☐ Export to Dump Project Folder
- ☒ Export to Self-Contained File
- ☒ Create Dump in a Single Transaction (self-contained file only)
- ☒ Include Create Schema

The 'Export Progress' tab is also visible, showing the status of the export process.

16. Schrijf een query waarmee je de volledige inhoud van de orders-tabel toont. De attributen *klantID*, *klantnaam*, *productID*, *productnaam*, *bestelhoeveelheid* en *afgehandeld* moeten getoond worden.



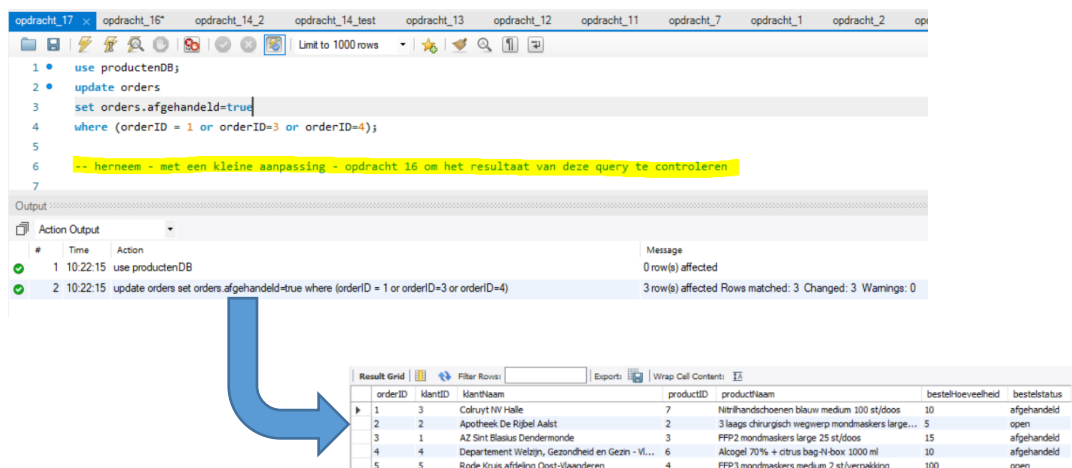
```

1 use productenDB;
2
3 -- mogelijkheid 1
4 select orders.klantID, klanten.klantnaam, orders.productID, producten.productnaam, orders.bestelhoeveelheid, orders.afgehandeld
5 from orders
6 inner join klanten on (orders.klantID=klanten.klantID)
7 inner join producten on (orders.productID=producten.productID);
8
9 -- alternative query
10
11 select orders.klantID, klanten.klantnaam, orders.productID, producten.productnaam, orders.bestelhoeveelheid, if (orders.afgehandeld, "afgehandeld", "open") as bestelstatus
12 from orders
13 inner join klanten on (orders.klantID=klanten.klantID)
14 inner join producten on (orders.productID=producten.productID);
15

```

klantID	klantnaam	productID	productnaam	bestelhoeveelheid	afgehandeld
1	AZ Sint Blasius Dendermonde	3	FFP2 mondklappers large 25 st/doos	15	0
1	AZ Sint Blasius Dendermonde	1	3 laag chirurgisch wegwerp mondklappers med...	20	0
1	AZ Sint Blasius Dendermonde	5	Thermoscan 6000 Pro thermometer	5	0
2	Apotheek De Rijbel Aalst	2	3 laag chirurgisch wegwerp mondklappers large...	5	0

17. Om de voorgaande opdrachten uit te voeren kan je gebruik maken van de statements die in de video-lessen zijn besproken. *In deze opdracht ga je op zoek naar een statement waarmee je de waarde van een attribuut kan wijzigen.* Schrijf een query waarmee je de afgehandeld-status van orders kan wijzigen. Wijzig de afgehandeld-status van de orders met *orderID* 1, 3 en 4 van 'false' naar 'true'.



```

1 use productenDB;
2
3 update orders
4 set orders.afgehandeld=true
5 where (orderID = 1 or orderID=3 or orderID=4);
6
7 -- herneem - met een kleine aanpassing - opdracht 16 om het resultaat van deze query te controleren

```

Output

#	Time	Action	Message
1	10:22:15	use productenDB	0 row(s) affected
2	10:22:15	update orders set orders.afgehandeld=true where (orderID = 1 or orderID=3 or orderID=4)	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0

orderID	klantID	klantnaam	productID	productnaam	bestelhoeveelheid	bestelstatus
1	3	Colruyt NV Halle	7	Nitrilhandschoenen blauw medium 100 st/doos	10	afgehandeld
2	2	Apotheek De Rijbel Aalst	2	3 laag chirurgisch wegwerp mondklappers large...	5	open
3	1	AZ Sint Blasius Dendermonde	3	FFP2 mondklappers large 25 st/doos	15	afgehandeld
4	4	Departement Welzijn, Gezondheid en Gezin - Vl...	6	Alcogel 70% + citrussag-N-box 1000 ml	10	afgehandeld
5	5	Rode Kruis afdeling Oost-Vlaanderen	4	FFP3 mondklappers medium 2 st/verpakking	100	open

18. Schrijf een query waarmee je de openstaande orders zichtbaar maakt. De attributen *klantID*, *klantnaam*, *productID*, *productnaam*, *bestelhoeveelheid* moeten getoond worden.

The screenshot shows a SQL query in the editor:

```

1 select orders.klantID, klanten.klantNaam, orders.productID, producten.productNaam, orders.bestelHoeveelheid
2 from orders
3 inner join klanten on (orders.klantID=klanten.klantID)
4 inner join producten on (orders.productID=producten.productID)
5 where orders.afgehandeld=false;

```

The result grid displays the following data:

	klantID	klantNaam	productID	productNaam	bestelhoeveelheid
1	AZ Sint Blasius Dendermonde		1	3 laags chirurgisch wegwerp mondklappers medi...	20
1	AZ Sint Blasius Dendermonde		5	Thermoscan 6000 Pro thermometer	5
2	Apotheek De Rijbel Aalst		2	3 laags chirurgisch wegwerp mondklappers large...	5
3	Colruyt NV Halle		8	Nitrilhandschoenen blauw large 100 st/doos	10
3	Colruyt NV Halle		3	FFP2 mondklappers large 25 st/doos	10
4	Departement Welzijn, Gezondheid en Gezin - Vil		5	Thermoscan 6000 Pro thermometer	?

The output pane shows a message: "7 row(s) returned".

19. Schrijf een query waarmee je voor één klant toont welke orders hij na een bepaalde datum heeft geplaatst. De attributen *klantID*, *productID*, *productnaam*, *bestelhoeveelheid* en *afgehandeld* moeten getoond worden.

The screenshot shows a SQL query in the editor:

```

1 select orders.klantID, orders.productID, producten.productNaam, orders.bestelHoeveelheid, if (orders.afgehandeld,"afgehandeld", "open") as bestelstatus
2 from orders
3 inner join klanten on (orders.klantID=klanten.klantID)
4 inner join producten on (orders.productID=producten.productID)
5 where (klanten.klantNaam = "Colruyt NV Halle" and orders.bestelDatum>"2020-04-01");
6

```

The result grid displays the following data:

	klantID	productID	productNaam	bestelhoeveelheid	bestelstatus
3	8	8	Nitrilhandschoenen blauw large 100 st/doos	10	open
3	3	3	FFP2 mondklappers large 25 st/doos	10	open

The output pane shows a message: "2 row(s) returned".

20. Een product verdwijnt uit het gamma. Kies een product waarvoor er orders zijn en verwijder dit product uit de tabel *producten*. Wat stel je vast? Wat zou je moeten doen om dit product alsnog te verwijderen?

The left screenshot shows a SQL query in the editor:

```

1 select * from producten;
2
3
4
5

```

The result grid displays the following data:

	productID	productNaam	productStock
1	3	3 laags chirurgisch wegwerp mondklappers medi...	500
2	3	3 laags chirurgisch wegwerp mondklappers large...	500
3	3	FFP2 mondklappers large 25 st/doos	30
4	3	FFP2 mondklappers medium 2 st/verpakking	1000
5	3	Thermoscan 6000 Pro thermometer	20

The right screenshot shows a SQL query in the editor:

```

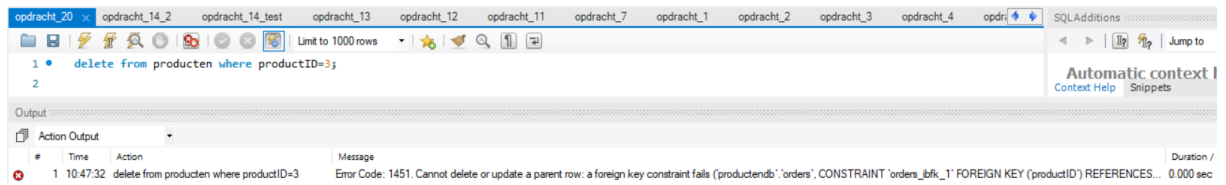
7 select * from orders;
8
9
10
11

```

The result grid displays the following data:

	orderID	bestelDatum	klantID	productID	bestelhoeveelheid	afgehandeld
1	2020-03-15	3	7	10	1	
2	2020-03-17	2	2	5	0	
3	2020-04-17	1	3	10	1	
4	2020-03-17	4	6	10	1	
5	2020-04-02	5	4	100	0	
6	2020-04-02	1	1	20	0	
7	2020-04-02	4	5	2	0	
8	2020-04-02	3	8	10	0	
9	2020-04-02	1	5	5	0	
10	2020-04-02	3	3	10	0	

The output pane shows a message: "10 row(s) returned".



10:47:32 delete from producten where productID=3

Error Code: 1451.

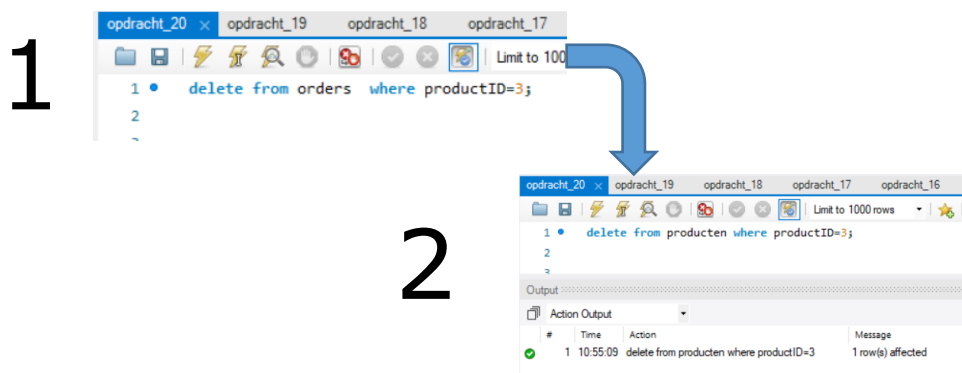
Cannot delete or update a parent row:

a foreign key constraint fails (`productendb`.`orders`,
CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`productID`) REFERENCES
`producten` (`productID`))

0.000 sec

Je kan niet zomaar een rij uit een tabel verwijderen wanneer er in een andere tabel nog een verwijzing is naar deze rij.

Oplossing : verwijder eerst in de tabel orders de rijen waar er nog een verwijzing staat naar het betreffende productID en verwijder dan in de tabel producten het betreffende product. => CASCADING



Opgelet : Het is misschien niet zo slim om orders te verwijderen uit een database. Gelukkig heb je nog een backup...

21. We willen de tabel producten uitbreiden met het attribuut 'beschikbaar'. Dit is een boolean attribuut dat default true is. Wanneer een product niet meer beschikbaar is of uit het gamma verdwijnt dan kan dit attribuut op false worden gezet. Zoek op met welk statement je een tabel kan wijzigen en gebruik dit statement in een query om het gevraagde attribuut toe te voegen.

The screenshot shows a SQL editor with the following query:

```
1 use productenDB;
2 alter table producten
3 add column beschikbaar boolean default true;
4
5 select * from producten;
```

The 'Result Grid' displays the following data:

productID	productNaam	productStock	beschikbaar
1	3 laags chirurgisch wegwerp mondklappers medi...	500	1
2	3 laags chirurgisch wegwerp mondklappers large...	500	1
3	FFP2 mondklappers large 25 st/doos	30	1
4	FFP3 mondklappers medium 2 st/verpakking	1000	1
5	Thermoscan 6000 Pro thermometer	20	1
6	Alcogel 70% + citrus bag-N-box 1000 ml	200	1
7	Nitrilhandschoenen blauw medium 100 st/doos	50	1
8	Nitrilhandschoenen blauw large 100 st/doos	40	1

The 'Output' section shows the execution results:

#	Time	Action	Message
1	12:23:14	use productenDB	0 row(s) affected
2	12:23:14	alter table producten add column beschikbaar...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	12:23:14	select * from producten LIMIT 0, 1000	8 row(s) returned

22. Voeg een product toe aan de producten-tabel. Zet het attribuut 'beschikbaar' op false. Schrijf vervolgens een query om een order voor dit product toe te voegen aan de orders-tabel. Deze query moet de beschikbaar-bit evalueren. Omdat deze false is mag het order niet toegevoegd worden aan de orders-tabel. Combineer insert-, select- en where-clause.

The screenshot shows a SQL editor with the following query:

```
1 -- tijdelijke tabel aanmaken. Deze is gelijkwaardig aan de orders-tabel.
2 -- we voegen er een extra kolom 'beschikbaar' aan toe.
3 create temporary table temp_orders like orders;
4 alter table temp_orders add column beschikbaar boolean;
5
6 -- Het toe te voegen order wordt in de tijdelijke tabel weggeschreven.
7 -- Uit de producten-tabel halen we ook de beschikbaarheid. Die schrijven we ook in deze producten tijdelijke tabel.
8 insert into temp_orders (bestelDatum, klantID, productID, bestelHoeveelheid, beschikbaar)
9 select
10     "2020-04-15",
11     (select klantID from klanten where klantNaam = "AZ Sint Blasius Dendermonde"),
12     (select productID from producten where productNaam = "Alco Spray 750 ml"), 10 ,
13     (select beschikbaar from producten where productNaam = "Alco Spray 750 ml") ;
14
15 -- Invoeren van het order in de orders-tabel op voorwaarde dat het product beschikbaar is
16 insert into orders (bestelDatum, klantID, productID, bestelHoeveelheid)
17 select bestelDatum, klantID, productID, bestelHoeveelheid from temp_orders
18 where beschikbaar=true;
19
20 -- tijdelijke tabel verwijderen
21 drop table temp_orders;
```

The 'Result Grid' displays the following data:

productID	productNaam	productStock	beschikbaar
1	3 laags chirurgisch wegwerp mondklappers medi...	500	1
2	3 laags chirurgisch wegwerp mondklappers large...	500	1
3	FFP2 mondklappers large 25 st/doos	30	1
4	FFP3 mondklappers medium 2 st/verpakking	1000	1
5	Thermoscan 6000 Pro thermometer	20	1
6	Alcogel 70% + citrus bag-N-box 1000 ml	200	1
7	Nitrilhandschoenen blauw medium 100 st/doos	50	1
8	Nitrilhandschoenen blauw large 100 st/doos	40	1
	Alco Spray 750 ml	0	0

The 'Output' section shows the execution results:

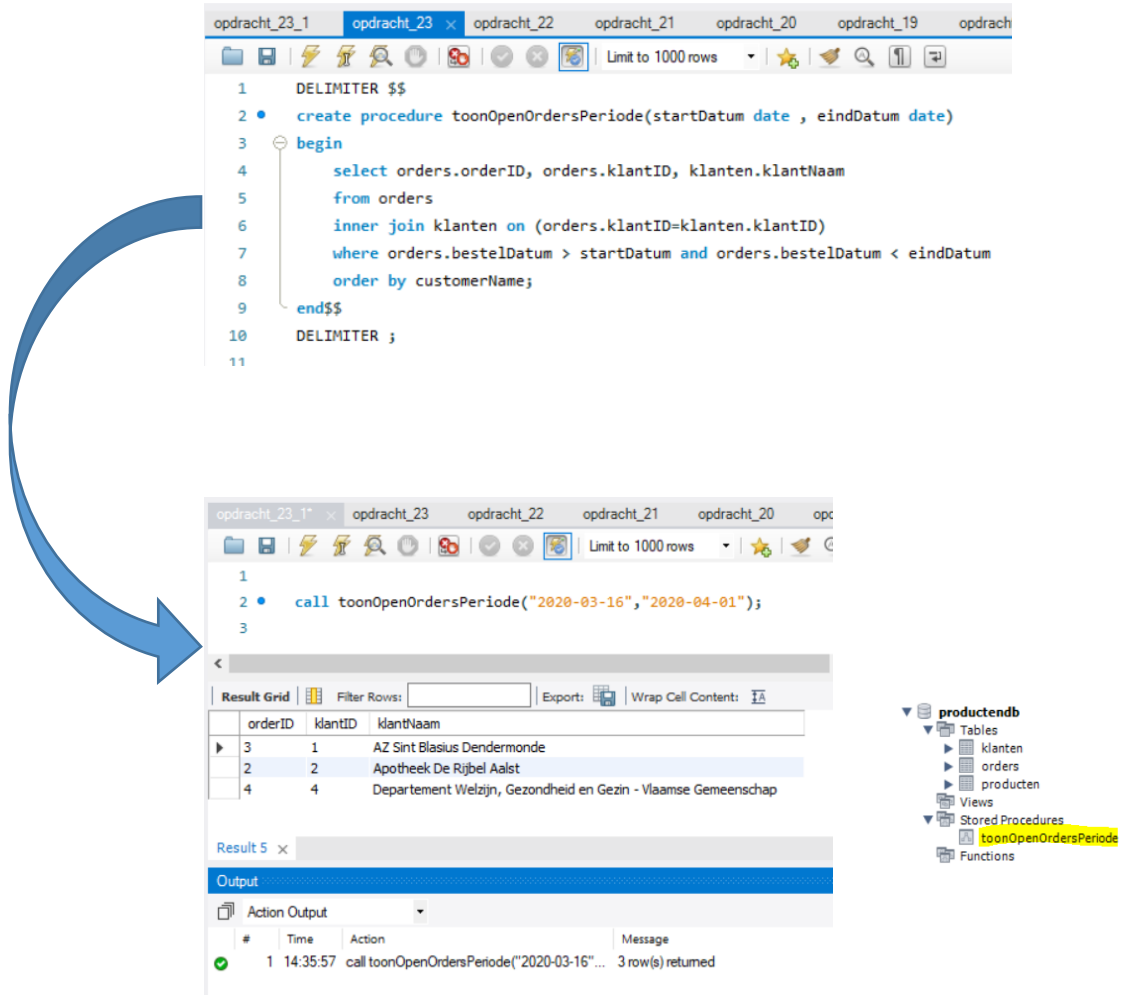
#	Time	Action	Message
2	14:13:05	alter table temp_orders add column besch...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	14:13:05	insert into temp_orders (bestelDatum, klant...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
4	14:13:05	insert into orders (bestelDatum, klantID, pro...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
5	14:13:05	drop table temp_orders	0 row(s) affected

Alternatieve oplossing : met een function()

```
opdracht_22_2  opdracht_22_1 x opdracht_22  opdracht_21  opdracht_20  opdracht_19  opdracht_18  opdracht_17  opd
Limit to 1000 rows
1 DELIMITER $$
2 • create function voegOrderToe(datum date, klant int, product int, hoeveelheid int) returns boolean
3 begin
4
5     set @boolVar = (select beschikbaar from producten where productID=product);
6
7     if @boolVar=true then
8         insert into orders (bestelDatum,klantID,productID,bestelHoeveelheid)
9             values (datum,klant,product,hoeveelheid);
10    end if;
11
12    return(@boolVar);
13 end$$
14 DELIMITER ;
15
```

```
opdracht_22_2  opdracht_22_1  opdracht_22  opdracht_21  opdracht_20  opdracht_19  opdracht_18  opdracht_17
Limit to 1000 rows
1 -- voegOrderToe(datum date, klant int, product int, hoeveelheid int) returns boolean
2 -- product dat beschikbaar is toevoegen
3 • set @resultaat = voegOrderToe("2020-04-17",2 ,7 ,10);
4 • select @resultaat;
5
6 -- voegOrderToe(datum date, klant int, product int, hoeveelheid int) returns boolean
7 -- product dat niet beschikbaar is toevoegen
8 • set @resultaat = voegOrderToe("2020-04-17",2 ,9 ,5);
9 • select @resultaat;
10
--
```

23. Schrijf een stored procedure 'toonOpenOrdersPeriode'. Deze procedure neemt twee datums (startDatum, eindDatum) als argument. Na uitvoering van de procedure worden de open orders getoond die tussen deze twee data zijn geplaatst. Je toont het *orderID*, het *klantID* en de *klantNaam*.



```

1 DELIMITER $$
2 • create procedure toonOpenOrdersPeriode(startDatum date , eindDatum date)
3 begin
4     select orders.orderID, orders.klantID, klanten.klantNaam
5     from orders
6     inner join klanten on (orders.klantID=klanten.klantID)
7     where orders.bestelDatum > startDatum and orders.bestelDatum < eindDatum
8     order by customerName;
9 end$$
10 DELIMITER ;
11

```

```

1
2 • call toonOpenOrdersPeriode("2020-03-16", "2020-04-01");
3

```

orderID	klantID	klantNaam
3	1	AZ Sint Blasius Dendermonde
2	2	Apotheek De Rijbel Aalst
4	4	Departement Welzijn, Gezondheid en Gezin - Vlaamse Gemeenschap

Result 5 ×

Output

Action Output

#	Time	Action	Message
1	14:35:57	call toonOpenOrdersPeriode("2020-03-16" ...	3 row(s) returned

productendb

- Tables
 - klanten
 - orders
 - producten
- Views
- Stored Procedures
 - toonOpenOrdersPeriode
- Functions

24. Schrijf een stored procedure 'zoekKlantOrders'. Deze procedure krijgt als parameter een *klantID*. Na uitvoering van de procedure worden van orders voor die klant *productID*, *productnaam* en *bestelhoeveelheid* getoond.

The image shows a MySQL IDE with several windows. The top window displays the SQL code for creating the stored procedure 'zoekKlantOrders'.

```

1 DELIMITER $$
2 create procedure zoekKlantOrders(klantID int)
3 begin
4     select orders.productID, producten.productNaam, orders.bestelHoeveelheid
5     from orders
6     inner join producten on (orders.productID=producten.productID)
7     where orders.klantID = klantID;
8 end$$
9 DELIMITER ;
10
11

```

To the right, a database schema diagram for 'productendb' is shown, highlighting the 'klanten' table with columns 'klantID' and 'klantNaam', and the 'orders' table.

Below the main window, three smaller windows show the execution of the procedure with different parameters:

- Left window:** Shows the execution of 'call zoekKlantOrders(5);'. The result grid displays one row:

productID	productNaam	bestelhoeveelheid
4	FFP3 mondklappers medium 2 st/verpakking	100
- Right window:** Shows the execution of 'call zoekKlantOrders((select klantID from klanten where klantNaam = "Colruyt NV Halle"));'. The result grid displays three rows:

productID	productNaam	bestelhoeveelheid
7	Nitrilhandschoenen blauw medium 100 st/doos	10
8	Nitrilhandschoenen blauw large 100 st/doos	10
3	FFP2 mondklappers large 25 st/doos	10
- Bottom window:** Shows the execution of 'call zoekKlantOrders(@klantID);' after setting '@klantID = (select klantID from klanten where klantNaam = "AZ Sint Blasius Dendermonde");'. The result grid displays three rows:

productID	productNaam	bestelhoeveelheid
3	FFP2 mondklappers large 25 st/doos	15
1	3 laags chirurgisch wegwerp mondklappers med...	20
5	Thermocan 6000 Pro thermometer	5

Arrows indicate the flow from the procedure definition to its execution with different inputs.