Database Ontwerp

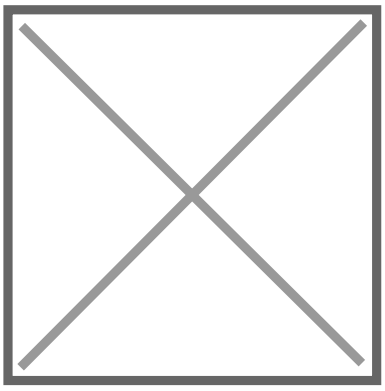
Database ontwerp voor 3de jaars. ERD, relaties, primary keus, foreign keys, koppeltabel, normaliseren.
Oefenen met examen cases. Flight case met complexere queries.

- DB Ontwerp, 3 opgaven
- Gegevensverzameling / testset aanleggen
- Market Research - 'flights case'
- Socrative Quiz - Database Ontwerp
- Case TV Company (en)
- Case Parking Lot (en)
- Northwind Case
- Practicum Snackbar
- Inner en Outer Joins
- Inner en Outer Joins - Opgaven
- Antwoorden Inner en Outer Joins - open!

DB Ontwerp, 3 opgaven

Opgave 1

De secretaris van een voetbalclub wil aan het begin van het seizoen de teamindeling in het clubblad publiceren volgens het voorbeeld hieronder.
Ook wil hij een overzicht publiceren met per team de trainingsavond en -tijden, de trainer en de coach.
Voor de verzending van het clubblad heeft hij adresstickers van alle spelers nodig.
Een speler kan slechts in één team meespelen.



Ontwerp het datamodel Maak 1,2 en 3 met pen en papier.

1. Welke entiteiten (personen, dingen gebeurtenissen) kun je onderscheiden? Teken een ERD met deze objecten (=entiteiten) en de relaties daartussen.
2. Noteer per entiteit (=tabel) de attributen (=velden) die nodig zijn om de informatie zoals hierboven beschreven staat, te kunnen leveren.
3. Bedenk per tabel wat je als sleutelveld kiest en in welke tabellen je een Foreign Key opneemt.
4. Maak het uiteindelijke datamodel in Workbench

Opgave 2

Afdeling personeelsadministratie van een ziekenhuis registreert in een dossier per medewerker welke training hij/zij heeft gevolgd. Tijdens het jaarlijkse functioneringsgesprek wordt deze informatie gebruikt om te bespreken of de kennis van de medewerker op peil is en welke trainingen deze medewerker het komende jaar moet volgen. Welke trainingen verplicht zijn, is afhankelijk van de functie van de medewerker. Er worden verschillende soorten trainingen aangeboden voor verschillende functies. De volgende tabellen worden hiervoor gebruikt:

MEDEWERKER (MedNummer, MedNaam, MedGeboortedatum, FunctieCode, DatumInDienst)

TRAININGSOORT (TrCode, TrainingOmschrijving, Doelgroep, StudieBelastingUren)

TRAINING (TrainingID, TrCode, docent, startdatum, AantalWeken, Aanvang, Eindtijd, Locatie)

DEELNAME (MedNummer, TrainingID, startdatum, resultaat)

FUNCTIE (FunctieCode, FunctieOmschrijving, MaxSalaris, StartSalaris)

1. Onderstreep de Primary Keys
2. Zet een sterretje bij de Foreign Keys*
3. Teken een ERD met tabelnamen (geen velden) en de relaties tussen de tabellen. Toon in de relaties de één en véél kant met kraaienpootjes.

Opgave 3

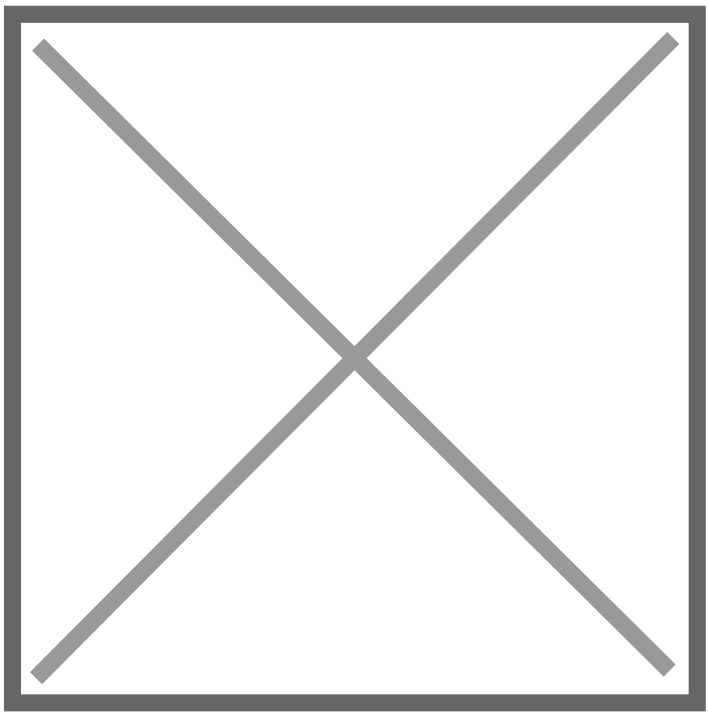
Een database bestaat uit tabellen die aan een aantal voorwaarden moeten voldoen. Hieronder staan een aantal tabelontwerpen. (Deze tabellen hebben niets met elkaar te maken.)

1. Wat is er fout aan de volgende tabellen?
2. Onderstreep in iedere tabel het sleutelveld.

- LEERLING (voornaam, tussenv, achternaam, leerlingnummer, leeftijd, adres, postcode, woonplaats)
- BESTELLING(bestelnummer, besteldatum, klantnr, product1, aantal1, product2, aantal2, product3, aantal3)
- CONTRIBUTIE_BETALING(lidno, achternaam, bankrekening, datum_betaling, bedrag)
- PRODUCT(productnummer, prijs, voorraad, leveranciersnaam, leverancierstelefoonnummer)

Gegevensverzameling / testset aanleggen

De afdeling calculatie van een meubelfabriek wenst periodiek een product-artikel-overzicht:



Een product is opgebouwd uit één of meer artikelen. Een artikel kan in meer producten voorkomen. Maak een document waarin je de volgende stappen (1. ERD, 2. Datadictionary, 3. testgegevens, 4. SQL en testresultaat voor het product-artikel-overzicht, 5. SQL en testresultaat voor het waardeoverzicht) documenteert. Laat dit aftekenden door de docent / lever dit in bij de docent.

1. Maak in Workbench een ERD voor database Meubelfabriek.
2. Forward Engineer dit naar mySQL en vraag de datadictionary op.
3. Vul de tabellen met testgegevens volgens bovenstaand overzicht (zonder product DEUR).
4. Maak een SQL query die de gegevens van bovenstaand overzicht oplevert.

Afdeling calculatie wil tevens de totale waarde per product zien

Productnaam	Waarde
Bovenkast	36.00

Onderkast	89.00
Lade	46.00

1. Maak een SQL query die de gegevens van dit waarde-overzicht oplevert. Gebruik hiervoor GROUP BY (zie w3schools 'group by')

Market Research - 'flights case'

De Case

Als free-lancer ben je benaderd door een luchtwaartmaatschapij die zich mogelijk wil gaan richten op de Amerikaanse markt. Jouw opdrachtgever heeft een aantal vragen waarbij hij jouw hulp vraagt.

> Voor het juist beantwoorden van de vragen heb je kennis nodig van: SELECT, WHERE, COUNT(), AVG(), MIN(), MAX(), SUM() DISTINCT, FROM, INNER JOIN, GROUP BY en HAVING.
>
> Kijk eventueel op w3schools hoe het ook weer precies zat.

Vragen

De vragen hebben betrekking op de Amerikaanse vliegmarkt:

1. Hoeveel vluchten vliegen er op het vliegveld *Philadelphia International Airport* *(tip: (1)* met *vliegen op* wordt de bestemming bedoeld; (2) de volledige naam van het vliegveld moet onderdeel van de query zijn).
2. Hoeveel vluchten zijn er in totaal gecancelled?
3. Hoevel vluchten zijn er in totaal omgeleid (diverted)?
4. Hoe groot was de grootste vertraging bij vertrek (departure delay)?
5. Soms wordt vertrekt een vlucht te vroeg en wordt er een negatieve departure delay vastgelegd. Voor het berekenen van de gemiddelde vertraging mogen deze vluchten niet meetellen. Wat is de gemiddelde vertraging (dus negatieve departure delay niet meegerekend).
6. Hoeveel vluchten zijn er die binnen de staat Calefornia vliegen, dus vertrekken en aankomen in Calefornia?
7. Hoeveel airtime (dus tijd in de lucht) heeft de kortste vlucht (in tijd)?
8. Wat is de kortste afstand van een vlucht en hoe lang duurde deze vlucht?
9. Wat is de airtime van de vlucht die de langste afstand heeft gevlogen?
10. Wat is de gemiddelde '*air time*' (dus tijd in de lucht) van alle vluchten die binnen de staat Calefornia vliegen?
11. Hoeveel vluchten zijn er die binnen een staat in Amerika blijven, dus vertrekken en aankomen indezelfde staat?

12. In welke staat is de gemidelde airtime van de vluchten die binnen een staat in Amerika blijven het laagst?
13. Op (naar) welke luchthavens wordt het meest gevlogen?
14. Vanaf welke luchthavens wordt het meest gevlogen?
15. Waarom kun je niet goed bepalen wat de drukste maand is? Maak de query en leg uit waarom je deze query niet goed kan testen.
16. Welk vliegveld ligt het meest zuidelijk?
    (tip: zoek eens op wat er in de kolom *latitude* staat, waar staat *latitide* voor?)
17. Welke luchtvaartmaatschappij legt de meeste vluchten af?
18. Welke luchtvaartmaatschappij legt legt de meeste airmiles af?
19. ~~Welke luchtvaartmaatschapij vliegt op de meeste bestemmingen?~~
    Ik weet niet of deze query mogelijk is zonder gebruik te maken van een temp table.
20. Maak een query die alle luchtvaarmaatschapijen laat zien die meer dan 350 bestemmingen hebben.
21. Tussen welke twee vliegvelden vindt de meeste vertraging bij vertrek plaats.?
22. Op welke vluchten (van-naar) is het vliegtuig gemiddeld het langs aan het taxiën?

Jouw opdrachtgever heeft alle vluchtinformatie van alle vluchten uit de USA uit 2015. Alle bovenstaande vragen kunnenn dus afgeleid worden uit de data van 2015.

Data

De data bestaat uit drie bestanden:

1. de vluchtgegevens
2. de luchthaven gegegevens
3. de airlines gegevens

Je krijgt een Excel sheet met drie tabjes waarin deze gegevens staan.

PoC

De vluchtgegevens zijn er dermate veel dat in de Excel sheet alleen de eerste 5000 vluchten zijn opgenomen. Het volledige bestand telt ruim 580 000 vluchten (en bijnan 600 MB). Als je PC krachtig genoeg is kun je het volledige bestand inlezen, anderes kun je volstaan met de eerste 5000 vluchten. Het gaat hierbij toch om een POC, Proof of Concept waarbij geldt dat als je queries werken op een set van 5000 dan werken ze ook op een set van 580 000.

Het is in zijn algemeenheid aan te raden om eerst met een kleine set data te werken. Het importeren van data en het testen van queries gaat dan namelijk veel sneller. Als je eenmaal hebt bedacht hoe je de database kunt opbouwen en hoe de queries er uit moeten zien, dan kun je daarna opschalen naar de volledige data set.

Jij ontvangt een Excel sheet van de opdrachtgever met de drie bestanden in drie afzonderlijke tabjes.

Database

Eén en ander kun je wellicht in Excel uitvoeren, maar Excel zal geen 500 000+ regels kunnen verwerken, bovendien wil de klant mogelijk later nog meer vluchtgegvens (uit meer jaren) toevoegen en zal de data-set nog verder vergroten. We zullen de gegevens dus moeten importeren in een database.

De Uitdaging

Je hebt dus een aantal uitdagingen:

1. Hoe krijg ik de Excel data in een (mySQL) database?
2. Hoe controleer ik en weet ik zeker dat alle data goed in de database zit?
3. Hoe maak ik de queries die antwoord geven op de door de klant gestelde vragen?

Planning

Maar voordat je met de uitvoering begint, denk eerst eens na over *hoe* je dit gaat aanpakken en maak een urenschatting. In het (mbo) examen zul je een Programma van Eisen en projectplan moeten opstellen. Dat gat hier te ver, maar bedenk wel voor jezelf welke stapjes je gaat doen en hoeveel tijd die kosten. Schrijf dit of leg dit vast op de computer. Het opstellen van een plan mag best wat werk kosten, want als het goed is kun je hier later veel tijd mee besparen.

Succes!

Excel bestand staat in Teams.

Socrative Quiz - Database Ontwerp

(Voor instructie)

- 10 vragen, 15 minuten.
- Bij de meeste vragen zijn meer dan 1 antwoord mogelijk
- Een test, geen cijfer.
- Doel: niveau groep controleren, kijken of groeps- en individueel niveau in de pas loopt
- Doel: voor jullie om te testen: is dit beter dan op papier?
- Doel: have fun!

Ga naar socrative.com

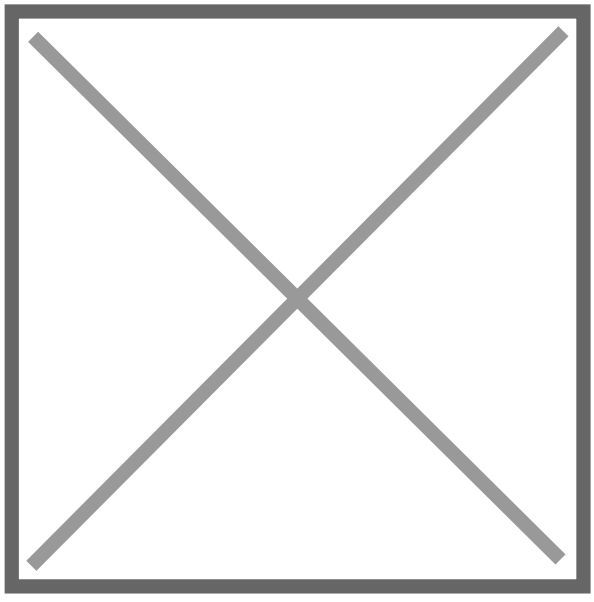
Kies student login

En vul Room Name in: ROCAMSTERDAM

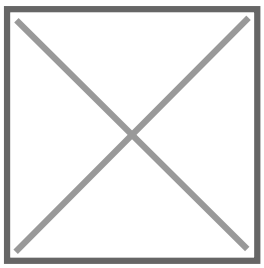
Vul je voornaam in (namen die ik niet herken kunnen geen score krijgen).

(Je kan taal op Nederlands instellenl werkt volgens mij niet)

Onderaan is navigatie



Niet op knop rechtsboven klikken, tenzij je klaar bent.



Case TV Company (en)

TV Company Case

A TV company wishes to develop a database to store data about the TV series that the company produces. The data includes information about actors who play in the series, and directors who direct the episodes of the series.

Actors and directors are employed by the company. A TV series are divided into episodes. Each episode may be transmitted at several occasions. An actor is hired to participate in a series, but may participate in many series. Each episode of a seriesis directed by one of the directors, but different episodes may be directed by differentdirectors.

Examples of database queries:

- Which actors play in the series Big Sister?
- In which series does the actor Bertil Bom participate?
- Which actors participate in more than one series?
- How many times has the first episode of the series Wild Lies been transmitted? Atwhat times?
- How many directors are employed by the company?
- Which director has directed the greatest number of episodes?

Develop an E/R model of this system. Find attributes of the entity sets. Determine whichof the attributes that can be used as primary keys.

Hulp

De opdracht is in het Engels, zorg ervoor dat je alles goed begrijpt. Zoek woorden op die je niet begrijpt. Als je de tekst verder niet begrijpt vraag het dan aan je collega-student of aan de docent.

Maak een verzameling van gegevens die je wilt gaan vastleggen. Schrijf hierbij alle atributen op. Let op dat je attributen zoveel mogelijk opdeeld. Adres wordt in principe opgedeeld in 'straatnaam', 'huisnummer' en 'toevoeging'.

Attributen die een vaste relatie met elkaar hebben leg je niet apart vast. Bedrag met en zonder btw leg je dus niet vast in twee kolommen. Je legt bedrag zonder btw vast en het btw percentage als deze variabel is.

Velden waarvan de waarde variaeert naarmate de tijd verloopt leg je ook niet vast. Een voorbeeld hiervan is leeftijd. In plaats van leeftijd leg je geboortedaum vast.

Deel deze gegevens op in groepjes (tabellen) zodanig dat je geen informatie dubbel hoeft op te slaan en dat er logische groepjes ontstaan. Op deze manier zal het aanpassen of verwijderen van gegevevens ook maar op één plek hoeven te gebeuren.

Definieer je relaties.

Kijk bij het definiëren van je relaties goed naar de tekst en naar de queries die je vanuit dit design kunnen worden gesteld.

Check

Check nog een keer door de hele tekst terug te lezen en bij elk stukje informatie te kijken of dat klopt in je datamodel.

Check of al je primary keys uniek zijn, bij twijfel maak je een apart ID aan.

Check of je relaties een foreign- en primary key heben.

Check bij 1:1 relaties of het niet logischer is om de twee tabellen samen te voegen.

Check nog en keer goed of er geen informatie dubbel opgeslagen wordt.

Case Parking Lot (en)

Develop an E/R model of a database that is to be used in the following system.

A homeowners association (that is, an association of people who own apartments) owns a parking lot. The parking lot has a number of parking spaces. The owners and their guests may freely use all the parking spaces, except some spaces that have electric sockets for electric cars. Such a parking space is rented by one of the apartment owners, who has exclusive use of the space. The rent for the space is added to the apartment rent.

The spaces with electric outlets are popular, and there is a queue of apartments that wish to rent such a space. Each apartment may have at most one place in the queue. When a space becomes available, the apartment with the longest queue time may rent it. One of the apartment owners must sign the contract (an apartment may have more than one owner, and an owner may own more than one apartment).

The association sometimes has problems with scrap cars that are deposited in theparking lot. It is often a difficult procedure to get rid of these cars. Discovery of a scrap car must be registered in the database, as well as each thing that happens with the car (Parking space 43: 2009–04–26 “Discovered suspect car, license number XYZ789”,2009–05–02 “Called the police about the car”, and so on).

Aanpak, zelfde aanpak als vorige case.

Northwind Case

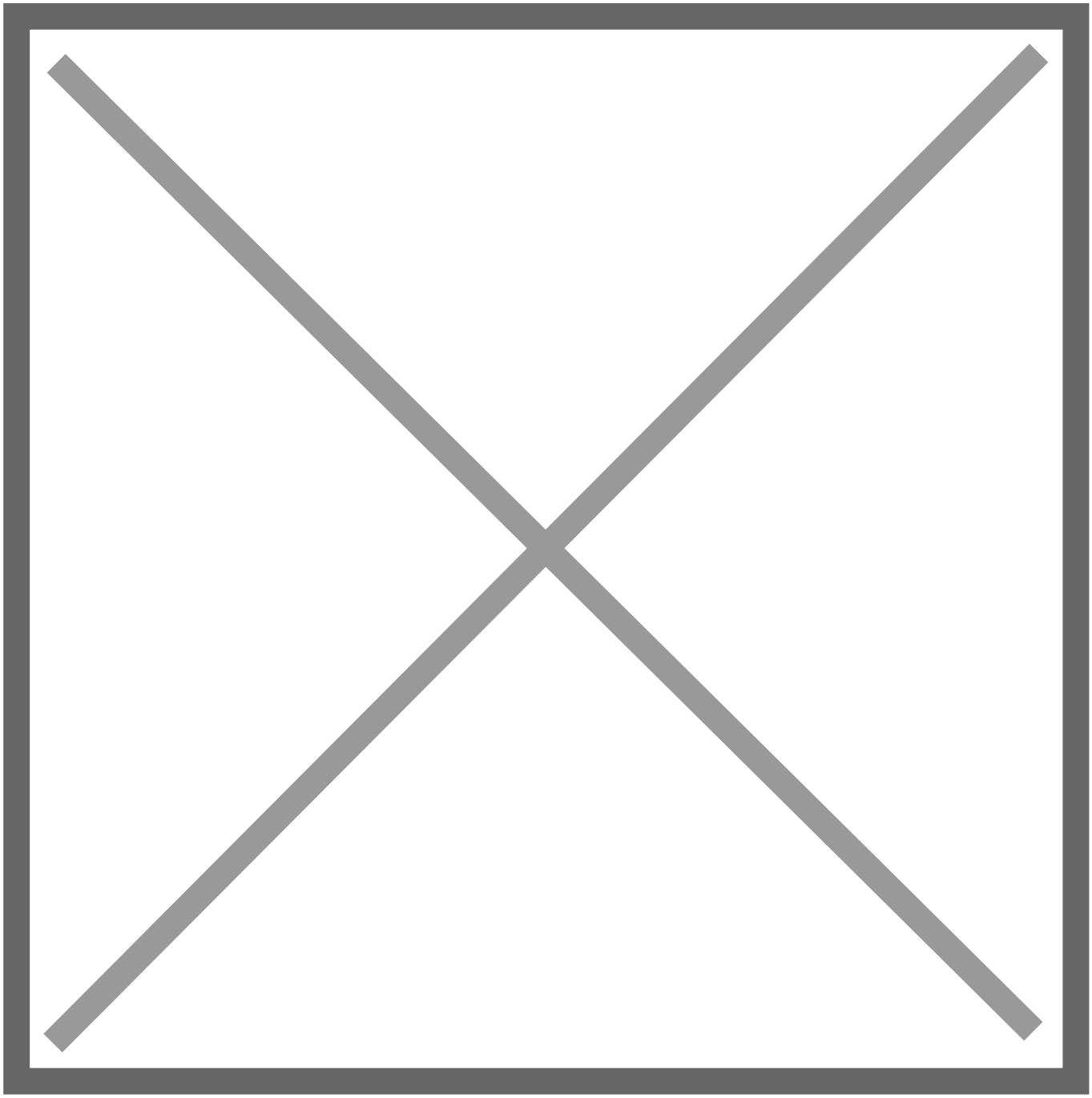
Deze case is afgeleid uit opdracht 7.4 van Stichting Praktijkleren-SQL Opdrachten 1.1 t/m 12.2-Module I en II

Doelen

...

Case

Je werkt net bij een nieuw bedrijf en ze willen dat jij een paar queries uitvoert op hun database. het eninge dat je hebt is een ERD diagram. Je kunt de diagram als PNG (plaatje) downloaden en dan op je laptop vergroten.



Opdracht

Voor deze opdracht dien je gebruik te maken van de database Northwind. Als je deze database nog niet hebt, dan kun je het script (instwnd.sql) downloaden van het netwerk of de elektronische leeromgeving vanuithet mapje ‘Scripts’. In deze database staat de tabel Customers. Je moet alle volgendevragen hierop uitvoeren.Als er vragen zijn over lesstof die je niet terug kunt vinden in de reader, gebruik dan de helpfunctie.Geef de SELECT-statements van de volgende vragen.

1. Laat alle gegevens zien van alle klanten uit de database.
2. Laat de naam (Companyname) en het land (Country) zien van alle klanten

3. Laat dezelfde gegevens zien als bij vraag twee, maar nu alleen van alle klanten uit Frankrijk. Let op! Het is een Engelstalige database
4. Geef nu dezelfde gegevens weer als bij vraag drie,maar nu dient het overzicht op naam te zijn gesorteerd

Voor de volgende vragen dien je gebruik te maken van de tabelProducts.

1. Laat van alle producten de naam (ProductName), prijs (UnitPrice) en categorie (CategoryID) zien.
2. Laat dezelfde gegevens zien als bij vraag vijf,maar dan alleen van de producten die meer dan €20,00 kosten.
3. Laat dezelfde gegevens zien als bij vraag zes, maar dan alleen van de producten uit categorie 1.
4. Laat de naam zien van alle producten die beginnenmet een ‘C’.
5. Laat alle producten zien waarvan de naam begint met een ‘G’en eindigt met een ‘E’.
6. Laat alle producten (de naam + prijs) zien die tussen de €30,00 en €40,00 kosten. Sorteer deze lijst op prijs,van hoog naar laag.

Practicum Snackbar

Maak een ERD voor een database op grond van het interview hieronder.

“Ik ben de eigenaar van een snackbar. We verkopen heel veel soorten snacks. Sommige werknemers werken achter de balie en nemen bestellingen op. Een bestelling kan uit één of meer snacks bestaan.”

“Ik wil graag weten wie van mijn werknemers het hardste werkt; wie neemt de meeste bestellingen op? Ik wil weten wat de drukste tijden op een dag zijn en wat de drukste dagen in de week zijn. Ik wil ook weten welke snacks het meest populair zijn. Ik heb verschillende soorten werknemers maar van allemaal moet ik de voornaam, achternaam, adres, stad, postcode, leeftijd en telefoonnummer weten.”

“Oh ja, iedere werknemer krijgt een salaris. Verder moet ik een aantal dingen weten die afhangen van de taak van de werknemer:

- Een kok heeft normaal gesproken op één of andere manier ervaring opgedaan: kokschool, zelfstudie, stage etc. Ik wil dat vastleggen.
- De bestelling-opnemer krijgt overuren betaald bovenop het normale salaris. Daarom wil ik vastleggen hoeveel hij per uur krijgt voor overuren.
- De manager is verantwoordelijk voor de supervisie over alle werknemers, heeft een budget voor uitgaven en een streef-omzet voor het restaurant waar hij/zij de leiding van heeft

Wanneer we groeien, zou ik andere soorten werknemers kunnen aannemen, maar ik weet op dit moment niet zeker wat voor soort werknemers dat zouden kunnen zijn.”

“Wanneer een klant een bestelling plaatst bij een opnemer dan is die opnemer verantwoordelijk voor het verloop van die bestelling: zorgen dat de kok hem krijgt, het uitleveren en de betaling in ontvangst nemen. Als een klant veranderingen wil of vragen heeft over de bestelling dan moet hij/zij bij de persoon zijn bij wie hij/zij de bestelling geplaatst heeft. De opnemer kan niet iemand anders vragen om dat over te nemen.”

“Je vroeg wat voor een soort dingen er in een bestelling voor kunnen komen? Dat zijn altijd allerlei soorten snacks. Alle snacks hebben een naam, een omschrijving en een prijs. Veel klanten hebben een klantenkaart. Met deze kaart krijgt de klant korting in de snackbar. We hebben dan wel meer informatie over de klant zoals zijn naam en zijn adres. Daarmee kunnen we de klant reclame toesturen en kortingsbonnen. Een ander voordeel is dat we nu bij kunnen houden welke snacks de klant vaak bestelt. Wanneer een klant de kaart gebruikt, kunnen we bijhouden welke bestellingen hij/zij met die kaart gedaan heeft.

Iedere klant kan slechts één klantenkaart hebben en iedere klantenkaart kan maar bij één klant horen.”

“Iedere werknemer hoort bij een ploeg. Momenteel hebben we een ochtendploeg en een middagploeg maar we overwegen ook een vooravondploeg. Momenteel gebruiken we een inschrijfformulier voor iedere ploeg maar dat raakt steeds kwijt en dan is het moeilijk voor mij om te achterhalen hoeveel iedereen gewerkt heeft. Enkele werknemers werken maar in één ploeg. Er zijn ook werknemers die in opeenvolgende ploegen werken. Het helpt me om te zien welke werknemers teveel en welke meer zouden kunnen werken. Daarom wil ik bijhouden wie dubbele ploegendienst draait, wie te weinig ploegendienst doet enz. Wanneer er een probleem is met een ploeg wil ik meteen weten welke werknemers er op dat moment werkten.”

Inner en Outer Joins

*In deze les leer je wat een **Cartesian product** is en hoe je deze kunt visualiseren. Daarna leer je het verschil tussen de **inner-** en de veschillende **outer joins**.*

Join

Je kunt een join visualiseren.

Stel je maakt een join tussen de tabellen persoon en de tabel ziektemeldingen.

Tabel persoon

ID	voornaan	achternaam
1001	Aagje	de Groot
1002	Amber	Nieuwenhuzien

Tabel ziektemeldingen

ID	persoon	datum_van	datum_tot
1	1002	14-06-2018	16-6-2018
2	1002	1-8-2018	6-8-2018
3	1003	2-8-2018	3-8-2018

INNER JOIN

Nu maak je een query waarbij je de twee tabellen samenvoegd.

```
SELECT *
FROM persoon, ziektemelding

SELECT *
FROM persoon
INNER JOIN ziektemelding
```

Deze queries zijn hetzelfde, hoewel de tweede is volgens de ANSI standaard en is 'hoe het hoort', omdat die duidelijker is.

Door deze query met een join wordt het **Cartesian product** bepaald:

ID	voornaam	achternaam	ID	persson	dataum_van	datum_tot
1001	Aagje	de Groot	1	1002	14-06-2018	16-6-2018
1001	Aagje	de Groot	2	1002	1-8-2018	6-8-2018
1001	Aagje	de Groot	3	1003	2-8-2018	3-8-2018
1002	Amber	Nieuwenhuizen	1	1002	14-06-2018	16-6-2018
1002	Amber	Nieuwenhuizen	2	1002	1-8-2018	6-8-2018
1002	Amber	Nieuwenhuizen	3	1003	2-8-2018	3-8-2018

Alle combinaties van alle rijen van de twee tabellen worden dus naast erlkaar gezet. Hierna begint de slelectie aan de hand van de WHERE clause van de query.

```
SELECT *
FROM persoon, ziektemelding
where persoon.id=ziektemelding.persoon

SELECT *
FROM persoon
INNER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

Dit resulteert in het 'verwijderen' van de vier regels van het Catersian product:

ID	voornaam	achternaam	ID	persoon	dataum_van	datum_tot
~~1001~~	Aagje	de Groot	1	~~1002~~	14-06-2018	16-6-2018
~~1001~~	Aagje	de Groot	2	~~1002~~	1-8-2018	6-8-2018
~~1001~~	Aagje	de Groot	3	~~1003~~	2-8-2018	3-8-2018
1002	Amber	Nieuwenhuizen	1	1002	14-06-2018	16-6-2018
1002	Amber	Nieuwenhuizen	2	1002	1-8-2018	6-8-2018
~~1002~~	Amber	Nieuwenhuizen	3	~~1003~~	2-8-2018	3-8-2018

Het resltaat is dat je alleen de regels krijgt waarvan de primary key en foreign key hetzelfde zijn.

LEFT OUTER JOIN

```
SELECT *
FROM persoon
LEFT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

ID	voornaam	achternaam	ID	persson	dataum_van	datum_tot
~~1001~~	Aagje	de Groot	1	~~1002~~	14-06-2018	16-6-2018
~~1001~~	Aagje	de Groot	2	~~1002~~	1-8-2018	6-8-2018
~~1001~~	Aagje	de Groot	3	~~1003~~	2-8-2018	3-8-2018
1002	Amber	Nieuwenhuizen	1	1002	14-06-2018	16-6-2018
1002	Amber	Nieuwenhuizen	2	1002	1-8-2018	6-8-2018
~~1002~~	Amber	Nieuwenhuizen	3	~~1003~~	2-8-2018	3-8-2018

Bij een **left** outer join wordt in de eerste stap hetzelfde gedaan als bij een inner join. Er wordt nu alleen gekeken of er er in de **linker** tabel regels zijn die niet één keer matchen met de rechter. Als dat zo is dan wordt deze regel uit de linker table dus toch afgedrukt. De kolommen uit de rechter kolom zijn er niet en zullen dus ook niet worden afgedrukt (je ziet NULL values).

> Dus bij een left outer join worden alle regels uit de linker tabel (dat is de tabel na de select) ten minste één keer afgedrukt.

RIGHT OUTER JOIN

```
SELECT *
FROM persoon
RIGHT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

ID	voornaam	achternaam	ID	persoon	dataum_van	datum_tot
~~1001~~	Aagje	de Groot	1	~~1002~~	14-06-2018	16-6-2018

~~1001~~	Aagje	de Groot	2	~~1002~~	1-8-2018	6-8-2018
~~1001~~	Aagje	de Groot	3	~~1003~~	2-8-2018	3-8-2018
1002	Amber	Nieuwenhuizen	1	1002	14-06-2018	16-6-2018
1002	Amber	Nieuwenhuizen	2	1002	1-8-2018	6-8-2018
~~1002~~	Amber	Nieuwenhuizen	3	1003	2-8-2018	3-8-2018

Bij een **right** outer join wordt in de eerste stap hetzelfde gedaan als bij een inner join. Er wordt nu alleen gekeken of er er in de **rechter** tabel regels zijn die niet één keer matchen met de rechter. Als dat zo is dan wordt deze regel uit de rechter table dus toch afgedrukt. De kolommen uit de linker kolom zijn er niet en zullen dus ook niet worden afgedrukt (je ziet NULL values).

> Dus bij een right outer join worden alle regels uit de rechter tabel (dat is de tabel na de on) ten minste één keer afgedrukt.

FULL OUTER JOIN

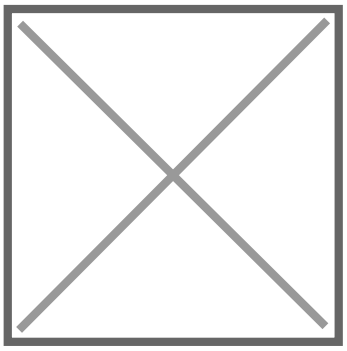
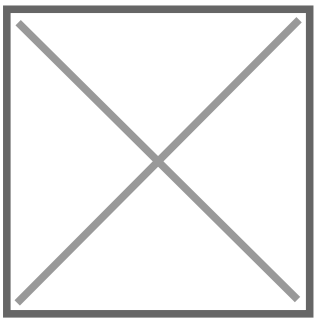
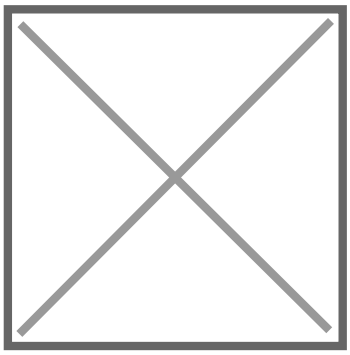
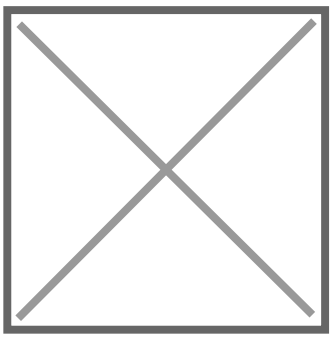
Bij een full outer join worden alle regels uit de linker en uit de rechter kolom ten minste één keer afgedrukt. Dit is een combinatie van de left- en right outer join. MariaDB en MySQL ondersteund geen full outer join. Deze kan worden uitgevoerd door de left- en right outer join te combineren met een union.

```
SELECT *
FROM persoon
LEFT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon

UNION

SELECT *
FROM persoon
RIGHT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

Samengevat

	SELECT * FROM A **INNER JOIN** B ON ...
	SELECT * FROM A **LEFFT OUTER JOIN** B ON ...
	SELECT * FROM A **RIGHT OUTER JOIN** B ON ...
	SELECT * FROM A **FULL OUTER JOIN** B ON ... (je kunt hier A en B ook omdraaien in de query)

Inner en Outer Joins - Opgaven

In deze les gaan we oefenen met de inner- en outer joins.

Case Meubelfabriek

We kijken nog een keer naar de Meubelfabriek case waar we eerder aan hebben gewerkt. Deze keer is de database echter in onderhoud en het lijkt erop dat nnog niet alle gegevens in de database staan. Toch willen we wat queries draaien. Om de juiste uitkomsten te krijgen zullen we de verschillende type joins moeten gebruiken.

Importeer eerst de (mini) database meubelfabriek_oj door het sql scipt te downloaden en te importeren (aan linker kant van deze pagina of onder files in Teams).

De query:

```
select count(*) from product, artikel, product_bestaat_uit
```

Moet als resutlaat 385 geven (Cartesian Product).

We hebben nu een database, meubelfabriek_oj (oj staat er achter om niet in de war te komen met de andere database die we al eerder hadden aangemaakt).

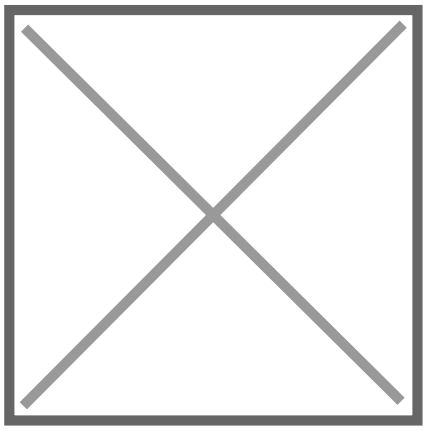
De database bestaat uit drie tabellen: product, artikel en een koppeltabel. De realtie product:artikel is N;M en de koppeltabel zorgt voor de ralatie.

Opgaven

Laat voor alle opgave zien welk query je hebt gebruikt.

Opgaven 1 (herhaling)

Maak een query die per product aangeeft uit welke en hoeveel artikelen dit product bestaan en druk ook de prijs van deze artikelen af. Het resutlaat ziet er ongeveer als volgt uit:



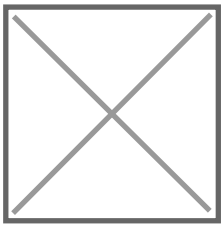
Opgaven 2 (herhaling)

Bereken de totaalprijs per product.

Product	Totaal Prijs
Bovenkast	36
Deur	20
Lade	32
Onderkast	89

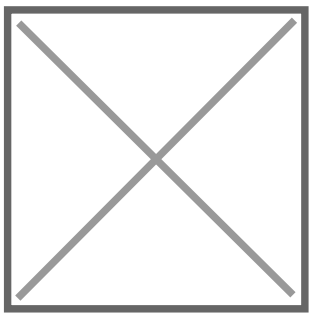
Opgave 3

Maak een overzicht gesorteerd op productnaam van alle producten dat laat zien uit hoeveel artikelen elk procuct bestaat. Bijvoorbeeld:



Opgave 4

Maak een overzicht gesorteerd op productnaam van alle producten dat laat zien uit welke artikelen plus aantal dit product bestaat. Dit is het detailoverzicht van opgave 3. Bijvoorbeeld:



Opgave 5

Het blijkt dat er een nieuwe product is ingevoerd maar dat de bijbehorende artikelen nog niet zijn ingevoerd. Pas nu het overzicht van opgave 4 aan zo dat het nieuwe product(en) ook zichtbaar wordt.

Alle producten moeten dus worden getoond.

> Tip: gebruik outer join!

Opgave 6

Pas nu de query uit opgave 5 aan zodat je alleen de/het nieuwe product(en) uit het overzicht afdrukt.

Dit zijn de producten die wel in de database staan maar waaraan geen artikelen zijn gekoppeld.

Opgave 7

Er zijn artikelen uit de database verwijderd maar er zijn nog producten die een verwijzing hebben naar deze artikelen. Bijvoorbeeld product buro bestond uit de artikelen 4 poten en 1 blad. Het artikel poten is verwijderd. Je ziet nu dat het product buro bestaat uit 4 onbekende artikelen plus een blad.

Zoek uit of er in de database een product bestaat waarvan het artikel is verwijderd maar waarvan het aantal wel bekend is.

Laat zien welk query je hebt gebruikt.

Opgave 8

~~Maak een overzicht van alle producten en artikelen met het aantal. Zelfde als bij opgave 4.~~

~~Nu blijkt dat er producten bestaan waarvoor de artikelen nog niet zijn ingevoerd. Zorg ervoor dat jouw query alle producten laat zien dus ook de producten die geen artikelen hebben.~~

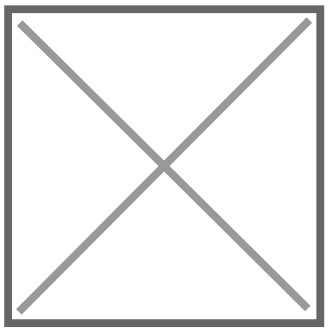
Is onduidelijk - kan worden overgeslagen.

Opgave 9

Welke artikel(en) wordt/worden in geen enkel product gebruikt?

Opgave 10 - bonus

Maak nu een overzicht zoals je bij opgave 4 maar laat alle producten en alle artikelen zien.



> Tip: Hiervoor heb je een full outer join nodig, kijk bij de vorige les hoe dat werkt.

--

Antwoorden Inner en Outer Joins - open!

```
--- (1) ---
SELECT p.naam product, a.naam artikel, aantal, prijs*aantal totaalprijs
FROM product p
INNER JOIN product_bestaat_uit k
on k.product_id=p.id
INNER JOIN artikel a
on a.id=k.artikel_id
order by .,naam

bovenkast - zijpaneel - 4 - 28
bovenkast - kastplank - 2 - 8
...

--- (2) ---
SELECT p.naam, sum(prijs*aantal) totaalprijs
FROM product p
INNER JOIN product_bestaat_uit k
on k.product_id=p.id
INNER JOIN artikel a
on a.id=k.artikel_id
group by p.naam

Bovenkast - 36
Deur - 20
Lade - 32
Onderkast 89

--- (3) ---
SELECT p.naam, sum(aantal) aantal
FROM product p
INNER JOIN product_bestaat_uit k
on k.product_id=p.id
```

```
INNER JOIN artikel a
on a.id=k.artikel_id
group by p.naam

Bovenkast - 6
Deur - 4
Lade - 7
Onderkast - 8



--- (4) ---
SELECT p.naam, a.naam, sum(aantal) aantal
FROM product p
INNER JOIN product_bestaat_uit k
on k.product_id=p.id
INNER JOIN artikel a
on a.id=k.artikel_id
group by p.naam, a.naam

Bovenkast - zijpaneel  - 4
Bovenkast - kastplank - 2
Deur - front - 2
...

--- (5) ---
SELECT p.naam, a.naam, aantal
FROM product p
LEFT OUTER JOIN product_bestaat_uit k
on k.product_id=p.id
LEFT OUTER JOIN artikel a
on a.id=k.artikel_id

Kan ook via product bestaat uit en dan een RIGHT OUTER Join met product en INNER JOIN met
artikel.

Resultaat: In ieder geval wipstoel erbij

(6)
SELECT p.naam, a.naam, aantal
FROM product p
LEFT OUTER JOIN product_bestaat_uit k
```

```
on k.product_id=p.id
LEFT OUTER JOIN artikel a
on a.id=k.artikel_id
where a.naam is NULL
```

Resultaat: Alleen de wipstoel

```
(7)
SELECT p.naam, a.naam, aantal
FROM product p
LEFT OUTER JOIN product_bestaat_uit k
on k.product_id=p.id
LEFT OUTER JOIN artikel a
on a.id=k.artikel_id
where a.naam is NULL and k.aantal is not NULL
```

Lade 2
Deur 6

```
(8)
onduidelijk


--- (9) ---
SELECT *
FROM artikel a
WHERE a.id not IN
(select product_bestaat_uit.artikel_id  from product_bestaat_uit)

of

SELECT *
FROM artikel a
LEFT OUTER JOIN  product_bestaat_uit
on product_bestaat_uit.artikel_id=a.id
WHERE product_bestaat_uit.product_id is NULL
```

1061 □afdekplank □3
1082 □dwars balk □5

```
(10)
SELECT p.naam product, a.naam artikel , sum(aantal) aantal
FROM product p
LEFT OUTER JOIN product_bestaat_uit k
on k.product_id=p.id
LEFT OUTER JOIN artikel a
on a.id=k.artikel_id
group by p.naam, a.naam

UNION

SELECT p.naam product, a.naam artikel, sum(aantal) aantal
FROM product p
RIGHT OUTER JOIN product_bestaat_uit k
on k.product_id=p.id
RIGHT OUTER JOIN artikel a
on a.id=k.artikel_id
group by p.naam, a.naam
```