

Module Database Design L2

In deze module gaan we leren zelf een ERD op te stellen. De eindopdracht is een 'klant opdracht; waarbij de klant jouw een paar overzichten laat zien en jou vraagt daar een database model van te maken.

- [Normaliseren](#)
- [Splitsen 1:N relatie](#)
- [Splitsen N:M relatie](#)
- [Opdracht De filmadministratie*](#)
- [Opgaven ERD maken](#)
- [Recycle Zonnepanelen](#)
- [Eindopdracht bestellingen en voorraad*](#)
- [Formule 1 Teammanagement Systeem](#)

Normaliseren

Leerdoel

In deze module leren we normaliseren. Normaliseren is het opzetten van een goed database design.

ERD

In de praktijk vraag je wat een klant wil. De klant vertelt wat hij wil en jij zet dat om in een ERD.

In Database design level 1 heb je al kennis gemaakt met een ERD. Om je kennis op te halen kan je nog een keer naar de uitleg in e video kijken: <https://web.microsoftstream.com/video/c4c62660-1c4f-4c89-b09b-948cf7559c86>

Waarom normaliseren?

Als je goed normaliseert en dus een goed datamodel maakt dan zorg je ervoor dat je gegevens zo worden opgeslagen dat:

- gegevens niet in tegenspraak zijn met elkaar
- gegevens maar één keer worden opgeslagen
- gegevens, maar op een plaats kunnen wijzigen

Wat je bijvoorbeeld niet wilt, is dat als je in een database iemand zijn geboortedatum vastlegt en dat je daarnaast ook de leeftijd vastlegt. Over een jaar of meer dan is iemand een jaar ouder en zijn de gegevens met elkaar in tegenspraak.

Of stel je hebt de prijs met en zonder BTW opgeslagen. Stel het BTW-tarief wijzigt dan moet je alle prijzen in de database opnieuw berekenen. Een goed database model zorgt ervoor dat dat niet hoeft.

Of stel je hebt iemand zijn naam verkeerd gespeld. Als je die aanpast in de database dan wil je dat één keer op één plek doen. Alleen op die manier weet je zeker dat je niets bent vergeten.

Zorgen dat gegevens maar één keer worden opgeslagen heet ook wel het voorkomen van redundantie.

Dus: voorkomen van redundantie is een het voorkomen van het dubbel vastleggen van gegevens.

Een goed datamodel bevat geen redundantie.

Hoe maak je een ERD?

Elke opgave bestaat uit een "verhaaltje" dit wordt ook wel een case genoemd. Lees de case goed door en bepaal van welke entiteiten je gegevens wilt vastleggen. Bepaal van alle entiteiten *wat* je wilt vastleggen, dit zijn de attributen. Bepaal van alle attributen het datatype (integer, float, date, time, datetime of boolean). Vervolgens bepaal je de relatie tussen de entiteiten en bepaal je de primary keys en foreign keys.

Dus de stappen zijn:

1. Bepaal van welke entiteiten je gegevens wilt vastleggen in de database.
2. Bepaal van elke entiteit welke gegevens je wilt vastleggen, dit zijn de attributen en bepaal het datatype per attribuut.
3. Bepaal de relaties tussen de entiteiten
4. Bepaal de PK's en FK's.

De datamodellen worden gemaakt in [Lucichart](#), een datamodel kun je in PDF exporteren en op je eigen systeem bewaren.

[image 1603478193579.png](#)

Om een ERD in [Lucichart](#) te maken, maak je een gratis account en zoek je de ERD-template "Database ER Diagram", zie plaatje hierboven.

Naming conventions

Het database design (het ERD) wordt in het Engels gemaakt. Entiteiten en Attributen worden via de [Snake Case](#) (in lower case) benoemd, bijvoorbeeld:

Entiteit: user

Attribuut 1: user_name

Attribuut 2: user_login_count

Attribuut 3: last_login_date

Let ook op dat entiteiten **enkelvoud** zijn. Dus bijvoorbeeld *user* en niet *users*, of *leerlingen* niet *leerlingen*.

Voorbeeld ERD

[image-1603486029048.png](#)

Dit diagram is in Lucichart gemaakt. Herken je de entiteiten en de attributen? Wat kun je van de relatie zeggen? Wat wordt er vastgelegd in deze database? Herken je de foreign key? Waar verwijst de foreign key naar?

Opgave 1

Bestudeer het ERD

[image-1603486029048.png](#)

Vraag 1

Hoeveel entiteiten zie je in het diagram?

- 1
- 2
- 3
- 8

Vraag 2

Wat is de foreign key in dit ERD?

- id
- student_id
- student
- sickness_notification

Vraag 3

De relatie die is getekend heeft een 'kraaienpootje' (of harkje). Het hakje staat aan de kant van de sickness_notification.

Wat betekent dit 'kraaienpootje'?

1. Elke student kan meer keer ziek zijn (geweest)
2. Elke ziektemelding kan door meer studenten worden gedaan.
3. Een ziektemelding hoort altijd bij één student.
4. Elk ID komt precies één keer voor en is dus uniek

Vraag 4

In de database wordt de birth_date (=geboortedatum) vastgelegd. Je zou ook de leeftijd kunnen vastleggen?

Is dat een goed idee en waarom?

1. Ja goed idee om in plaats van birth_day de leeftijd vast te leggen. Leeftijd is namelijk kleiner en kost minder opslag ruimte.
2. Ja goed idee om beide vast te leggen. Je kunt dan heel snel de leeftijd of de geboortedatum (birth_day) opzoeken.
3. Nee geen goed idee om beide vast te leggen, kost veel te veel opslag ruimte.
4. Nee geen goed idee om leeftijd vast te leggen, dat verandert te vaak en je kunt de leeftijd altijd berekenen als je de birth_day weet.

Vraag 5

Waarom normaliseer je (meerdere antwoorden mogelijk)?

1. Om dubbele opslag van gegevens te voorkomen.
2. Om te voorkomen dat gegevens in tegenspraak zijn met elkaar.
3. Om gegevens eenvoudig te kunnen wijzigen.
4. Om ervoor te zorgen dat je heel snel gegevens uit een database kunt opzoeken.
5. Om te voorkomen dat je telkens updates moet draaien.

Vraag 6 - Tandarts

Je wilt bijhouden welke patiënten bij de tandarts welke behandelingen hebben gekregen.

Je wilt vastleggen welke patiënt met welke behandeling door welke tandarts is behandeld en wanneer dat is gebeurd.

Je wilt ook van een patiënt kunnen zien welke behandelingen bij hem of haar zijn uitgevoerd.

Je moet er rekening mee houden dat patiënten van tandarts kunnen wisselen omdat zij bijvoorbeeld verhuizen.

Elke behandeling heeft een prijs die je wilt vastleggen in de database.

Vink de entiteiten aan:

1. Tandarts
2. Patiënt
3. Behandeling
4. Lijst
5. Behandelprijs
6. Behandeldatum
7. Verhuizingen

--

Splitsen 1:N relatie

Normaliseren doen we in een aantal stappen.

Stap 1, maak één lijst

We beginnen met het opschrijven van alle attributen/velden die we moeten opslaan. Het is handig om dit Excel te doen. Bijvoorbeeld, stel je wilt van leerling, het leerlingnummer, de naam, de klas en de studiecoach vastleggen. Dan maak je de volgende lijstje.

[image-1632404042140.png](#)

Stap 2, splits attributen

Neem de lijst en bekijk of het nodig is om attributen (velden) te splitsen. In het bovenstaande voorbeeld zou het handig zijn om voor- en achternaam te splitsen. Je kunt je namelijk voorstellen dat je een lijst wilt kunnen maken gesorteerd op achternaam of juist op gesorteerd op voornaam.

[image-1632404381453.png](#)

Stap 3, splits in twee entiteiten

In het bovenstaande voorbeeld zien we dat de naam "Judith Mustafa" twee maal voorkomt. In het "echte leven" zou het zomaar kunnen dat deze naam 100 keer voorkomt. Wat nu als Judith Mustafa ontslag neemt en wordt vervangen door "Ray Hilton"? Dan moeten we dit meerdere maken aanpassen. Dat willen we niet.

Kijk eens naar deze tabel.

[image-1632405235058.png](#)

Op de eerste regel bij de voornaam van de studiecoach staat Judit en op de tweede regel staat Judith. Gaat het om dezelfde Judith en is er een tikfout gemaakt of gaat het om dezelfde studiecoach? Dit willen we ook niet.

We zien nu twee problemen die kunnen we oplossen door te gaan splitsen;

[image-1632408246837.png](#)

We splitsen de entiteit in twee nieuwe entiteiten: *leerling* en *studiecoach*.

In dit voorbeeld is Judit een andere persoon als Judith. Ze hebben namelijk beide een ander ID gekregen. Zou er wel een tikfout zijn gemaakt dan zou het er als volgt hebben uitgezien.

[image-1632408480432.png](#)

Relatie

Bij deze vorm van splitsen (we leren nog een andere methode) splits je altijd in een 1:1 of een 1:N relatie. Dat is een één op één of een één op veel relatie (1:N staat voor één op veel). Je zult zien dat je vrijwel altijd een 1:N relatie overhoudt. In een ERD komen 1:1 relaties niet vaak voor, ook daar volgt later meer uitleg over.

Hoe zit het nu met het voorbeeld? Heeft één leerling meerdere studietoelichters of heeft één studietoelichter meerdere leerlingen?

In dit voorbeeld heeft één studietoelichter meerdere leerlingen.

Wanneer wel en wanneer niet afsplitsen?

Maar wacht eens even.... waarom moeten we klas niet afsplitsen? Klassen komen toch ook meerdere keren voor? Laten we klas eens afsplitsen en kijken wat er gebeurt.

[image-1632413682000.png](#)

Je ziet dat we in de eerste entiteit *leerling* klas hebben vervangen door de foreign key *klas_id*. Klas staat dan in een aparte entiteit en heeft een primary key ID. Wat was ook alweer een primary key? Ja dat was een uniek attribuut. In de entiteit klas is klas zelf uniek en zouden we dus geen aparte unieke key hoeven toe te voegen. Je kan de kolom ID weglaten en de foreign key in de leerling entiteit wordt dan klas.

[image-1632413948728.png](#)

En als we dit dan zien dan kunnen we de entiteit dus net zo goed weglaten.

De regels voor het normaliseren zijn zo dat je door splitsen moet voorkomen dat je dubbele gegevens vast legt zoals in het voorbeeld naam van de studietoelichter. Maar dat geldt alleen voor strings, en dus niet voor numerieke waarden.

Voorbeelden van attributen die dubbel mogen voorkomen:

- getallen
- prijzen
- nummers
- datums
- temperatuur

- klasnummer
- ...

Je ziet dus dat als je een klasnummer hebt dat je dat niet afsplitst.

Stel elke klas heeft naam zoals klas 'Code Warriors', 'Fast Hackers', 'Genius Coders',... dan zou je die wel afsplitsen:

[image1632414587830.png](#)

Opdracht 1

Een garage wil van zijn kanten de volgende gegevens vastleggen:

[image1632415868785.png](#)

Splits dit op in drie entiteiten en voeg de primary keys toe.

Werk het uit in Excel en lever Excel bestand in.

Opdracht 2

De stappen voor normaliseren (tot nu toe) die je hebt geleerd zijn;

1. één lijst maken
2. attributen splitsen
3. entiteiten splitsen

Bepaal welke stappen je nodig hebt om onderstaand lijstje te normaliseren.

[image1632415331697.png](#)

Maak drie entiteiten en voeg de primary keys toe.

Werk het uit in Excel en lever Excel bestand in.

Opdracht 3

Werk de twee gesplitste entiteiten uit het voorbeeld verder uit tot een ERD.

[image-1632408480432.png](#)

Voeg keys (PK en FK) toe en teken de relatie. Gebruik Lucichart.

Lever een schermafdruck in.

Opdracht 4

Bekijk dit lijstje.

[image-1632419578954.png](#)

Tijd om te laten zien dat je alle stappen beheerst!

Gebruik de stappen zoals je die hebt geleerd en maak een ERD in Lucichart.

Schrijf eerst alle gegevens op, en splits de attributen en de entiteiten waar nodig.

Maakt het complete **ERD** met **datatypes**, **keys** en **relaties**.

Kom je er niet uit, kijk dan naar dit filmpje: <https://web.microsoftstream.com/video/856f3111-8f95-440c-8434-573e5063eb17>

Let op het splitsen wordt in het filmpje op een iets andere manier uitgelegd als in deze les, maar het resultaat is hetzelfde.

Best lastig he? Kom je er nog niet uit, kijk dan naar deze video: <https://youtu.be/6CDc2IMbTiw>

<https://www.youtube.com/embed/6CDc2IMbTiw>

Kom je er **na het kijken van beide video's** nog niet uit, probeer dan duidelijk aan te geven wat je niet begrijpt en vraag hulp aan jouw docent.

Kom je er wel uit (met of zonder video's)? Check dan je foreign keys! Staat bij elke foreign key een relatie en staat het 'harkje' aan de kant van de foreign key?

Goed gedaan, je zit al bijna op examen-niveau!

Lever een schermafbeelding van je ERD in.

Succes!

--

Splitsen N:M relatie

Normaliseren

In de vorige les hebben we de eerste stappen van het normaliseren geleerd.

Stap 1, maak een lijst
Stap 2, splits waar nodig de attributen
Stap 3, splits waar nodig in twee entiteiten

Nu gaan we nog een stap toevoegen.

In onze vorige les hadden we het over leerlingen en hun studietoestel. In dat voorbeeld had één studietoestel meerdere leerlingen maar een leerling had één studietoestel.

De relatie **studietoestel:leerling** was **1:N**

Maar het kan ook anders, namelijk:

De relatie **studietoestel:leerling** is **N:M**

Dat betekent dat een studietoestel meerdere leerlingen heeft, maar dat andersom ook geldt dat een leerling meerdere studietoestels heeft.

Laten we de stappen nog een keer doorlopen met dit nieuwe voorbeeld.

Stap1, maak een lijst

[image-1632423177376.png](#)

In dit voorbeeld heeft Bailey Spears twee studietoestels: Judit Mustafa en Ales Maxwell.

Stap 2, splits attributen

Dit is in dit voorbeeld al gedaan dus dit hoeft niet verder meer.

Stap 3, splits waar nodig in entiteiten

We moeten nu ook splitsen, maar omdat we meer studietoestels per leerling kunnen hebben moeten we anders splitsen. Let op dat in het voorbeeld er twee studietoestels bij één leerling horen. Het zou ook zo kunnen zijn dat er drie of nog meer studietoestels aan één leerling kunnen worden gekoppeld.

Als we dit soort situaties tegen komen waar de relatie N:M is. dan moeten we splitsen via een koppelentiteit.

[image-1632410260924.png](#)

We zien nu drie entiteiten; leerling, studiecoach en de **koppelentiteit**. Kijk goed naar de nummers in de koppelentiteit. Zie je hoe de koppeling werkt? De koppeling verbindt als het ware de linker entiteit *leerling* met de rechter entiteit *studiecoach*. In de koppelentiteit staan telkens twee ID's die verwijzen naar een leerling en een *studiecoach*. Dus leerling 213421 is gekoppeld aan studiecoach 1 en dezelfde leerling 213421 is ook gekoppeld aan studiecoach 2.

Keys van de koppelentiteit

In de koppelentiteit moet je nog een PK plaatsen. Een PK mag een samengestelde key zijn Dat is een key die uit twee of meer velden bestaat. De combinatie *leerling_id* en *studiecoach_id* is altijd uniek en zou dus een PK kunnen zijn.

In de praktijk is het echter veel eenvoudiger om gewoon nog een extra key ID toe te voegen.

[image-1632423535537.png](#)

Opdracht 1

De koppelentiteit vertaald zich in een database naar een koppeltabel. In deze korte uitleg wordt nog een keer uitgelegd wat de functie van de koppeltabel is.

https://www.youtube.com/embed/4bwu4s_wjUs

Als je tussen student en vak een N:M relatie hebt dan moet je een koppeling maken met een koppelentiteit.

Vraag 1

Stel dat je voor elke vak voor elke leerling wil vastleggen welk cijfer hij voor dat vak heeft, waar leg je dat dan vast? Dus in welke entiteit (tabel) wordt het cijfer vastgelegd?

1. student
2. vak
3. koppel
4. cijfer

(het antwoord staat in de video).

Vraag 2

Hoeveel FK's (foreign keys) staan er in de koppelentiteit/koppeltabel?

1. 0
2. 1
3. 2
4. 3

Vraag 3

Een relatie in een ERD heeft een kraaienpootje. Het kraaienpootje staat *altijd* bij een....

1. de PK, de primary key
2. de FK, de foreign key
3. de koppelentiteit
4. is niet te zeggen

Opdracht 2

Maak een ERD van de volgende drie entiteiten. Neem daarbij ook de koppelentiteit op. Benoem de keys, PK en FK en teken de relaties op de juiste manier.

[image.1632410260924.png](#)

Lever een schermafdruck in van je ERD.

Opdracht 3

Maak een ERD van de volgende gegevens. Bepaal de entiteiten, attributen, keys en relaties.

[image.1632422136615.png](#)

Lever een schermafdruck in van je ERD.

Opgave 4, Fietsenmaker Snelle Jelle

In de video die bij de opgave hoort gaan uit van drie basis regels voor het normaliseren:

1. Geen dubbele kolommen, anders afsplitsen
2. Alle regels moeten uniek zijn
3. Geen afgeleide kolommen

Om tot deze basis regels te komen passen we de technieken toe die we hebben geleerd.

Fietsenmaker Snelle Jelle wil na een reparatiebeurt zijn klanten per SMS of Whatsapp op de hoogte stellen dat de reparatie klaar is. In dit bericht wil hij ook vertellen hoe hoog de reparatiekosten zijn.

Omdat veel klanten meer dan één fiets hebben, wil hij van de fietsen ook wat kenmerken vastleggen. Hij wil het merk, model, type en kleur kunnen vastleggen.

Van elke reparatiebeurt wil hij verder vastleggen wanneer het onderhoud plaatsvond, hoe lang de reparatie duurde, wat er is uitgevoerd en de prijs. Alle reparatiebeurten hebben een standaard prijs. Zo kost het vervangen van een binnenband 12 euro en het vervangen van de ketting op een stadsfiets 37 euro.

In deze uitleg wordt uitgelegd hoe je dit moet doen: <https://youtu.be/qKuzxf2iH1I>

<https://www.youtube.com/embed/qKuzxf2iH1I>

Let op de opdracht in de video is **bijna** hetzelfde, maar in de video worden de prijzen van de reparatie niet vastgelegd.

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

Opgave 5

Een snackbar wil bijhouden welke order door welke koerier is weggebracht, hoe laat dat was en hoeveel tijd dat heeft gekost om de bestelling weg te brengen.

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

Bedenk of je zelf een ERD kan maken en bekijk dan de uitleg in de video.

<https://youtu.be/B5QhBuchsRM>

<https://www.youtube.com/embed/B5QhBuchsRM>

Opdracht 6

Maak van de volgende gegevens een ERD. Bepaal de entiteiten, attributen, keys en relaties.

Je kunt de gegevens uit de onderstaande tabel via copy/pase naar Excel overzetten. Excel kan je dan gebruiken bij het splitsen.

Student	Cursus	Coach	telefoon_coach
Kevin Drum	Digitale Vaardigheden, Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Ayoub	06 1221 3268
John Jones, 1234	C++	Ayoub	06 1221 3268
Sally-Jane Jones	HTML, Python	Ayoub	06 1221 3268
David (getrouwd)	Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Samina	06 2100 3485
Murvin Drake	C++	Samina	06 2100 3485

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

Opdracht 7

Maak van de volgende gegevens een ERD. Bepaal de entiteiten, attributen, keys en relaties.

Je kunt de gegevens uit de onderstaande tabel via copy/pase naar Excel overzetten. Excel kan je dan gebruiken bij het splitsen.

Iant	boek_id	boek ISBN	Titel	uitgeleend	uitleentermijn	ingeleverd
Caris Villa	234189	817525766-0	De boze wolff	09/06/2021	21	
Caris Villa	234288	817333566-1	De lastige wereld	07/05/2021	21	14/05/2021

Caris Villa	134585	817525766-0	Op reis met JJ	28/04/2021	21	29/04/2021
Caris Villa	232269	817528866-3	Het grote feest	04/05/2021	21	25/05/2021
Ayana Tucker	232270	817528866-4	Het grote feest	29/05/2021	35	07/06/2021
Shola Greig	134585	817525766-0	Op reis met JJ	28/05/2021	21	

Lever een schermafdruck in van je ERD.

Opdracht 8

Bij autohandel "Krakkemik" staan er auto's van verschillende merken op het terrein. In hun database staat bijvoorbeeld dat er; een rode Opel Astra uit 1998 voor 1500 euro te koop is, een groene Ford Escort uit 2002 voor 2000 euro en een Peugeot waarvan de prijs 4000 euro is.

Herken jij de entiteit? Van wat wordt hier gegevens vastgelegd?

En wat zijn de attributen van de entiteit? Benoem alle attributen.

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

Opgave 9, ToDo App

Maak een datamodel (ERD) voor een ToDo applicatie. In de Todo applicatie wil je de naam van een ToDo item kunnen vastleggen. Van elk ToDo item wil je de titel kunnen vastleggen, bijvoorbeeld "paspoort verlengen". Verder wil je van elk ToDo item vastleggen wanneer hij is aangemaakt en of hij afgerond is.

Verder wil je de ToDo items opdelen in categorieën. Bijvoorbeeld, de categorie "boodschappen", "privé" of "werk". Van elke categorie wil je vastleggen wanneer deze voor het laatst gewijzigd is.

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

Opgave 10, Covid-19 test

Het RIVM wil een eenvoudige database waarin ze kunnen zien welke persoon wanneer is getest en wat de uitslag was (positief of negatief). Van alle geteste hoeft alleen de voor- en achternaam en het e-mailadres te worden vastgelegd.

Het RIVM wil ook graag weten of de geteste persoon de Corona app heeft geactiveerd. Als dat zo is dan kunnen ze namelijk via de app andere mensen op de hoogte stellen als zij langere tijd bij een positief getest persoon in de buurt zijn geweest.

Maak het ERD in [Lucichart](#).

Lever een schermafdruck in van je ERD.

--

Opdracht De filmadministratie*

Jouw klant wil een administratie op stellen voor het verwerken van filmgegevens.

Bij elke film hoort de lijst met acteurs en het personage dat zij in de film speelden.

Verder moeten de gegevens van de volledige filmcrew worden opgeslagen.

De filmcrew bestaat uit de regisseur, cameramensen, etc, etc.

Jouw klant heeft een voorbeeld gegeven hoe het overzicht dat hij uit de database wil kunnen halen er uit ziet.

Voorbeeld overzicht

Film gegevens		
FilmID	1999/57	
Titel	Bicentennial Man	
Productiehuis	Touschstone Pictures	
Release Data	17 december 1999	
MPAA rating	PG	
Korte inhoud	In the first decade of the new millenium a new discovery is done.....	
Main cast	Acteur	Personage
	Robin Williams	Andrew Martin
	Sam Neil	The Sentinel
	Loliver Patt	Rupert Burns

	An Delusion	Elise Stephard
	...	
Crew	Rol	Naam
	Director	Cax Bischof
	Screen Writer	Nick Kazan
	Special Effects	Steve Jewel
	Special Effects	Thor Knight
	Producer	Mark Gardiff
	Costume Designer	Elisabeth Grand
	Costume Designer	Jason Bonne
	Costume Designer	Carine Elisabeth Studs
	...	

Teken het volledige ERD in Lucichart en lever een screendump in waar alle entiteiten duidelijk te zien zijn en waar alle attributen, keys en relaties zijn weergegeven.

--

Opgaven ERD maken

ERD's maken

Van elke opgave voer je de volgende stappen uit.

1. Bepaal de entiteiten.
2. Bepaal de relatie tussen de entiteiten, 1:1, 1:N, of N:M
 - Voor 1:1 relaties bekijk je of je de entiteiten kan samenvoegen.
 - Voor N:M relaties maak je een constructie met een koppeltabel.
3. Teken de relaties en zet de harkjes aan de 'meer-kant'.
4. Zet bij elke entiteit een PK (id).
5. Zet bij elk harkje een FK die verwijst naar de PK van de gerelateerde entiteit.
6. Bepaal alle attributen (velden) die je wilt vastleggen.
7. Bepaal alle datatypes, meest voorkomende: *int*, *varchar()*, *date*, en *datetime*. Alleen *varchar* heeft een maximale lengte.

Taxi

Taxibedrijf 'Veilig' wil reserveringen bijhouden. Zij willen vastleggen wie voor welke rit staat ingepland. Zij willen vastleggen welke chauffeur met welke taxi welke rit uitvoert. Van de rit wordt vastgelegd hoe laat die begint, van waar to waar de rit loopt, wat de geschatte afstand is en wat de afgesproken prijs is.

Ritten worden van te voren besteld en betaald. Van elke geplande rit wordt vastgelegd wie de klant is. Van de klant wordt naam, email en telefoon vastgelegd. Op die manier kan de klant worden gebeld als er wat is met de geplande rit.

[Image-1643187071903.png](#) shown

Huiswerk

<p>De school wil bijhouden hoeveel opgaven een student in een week maakt.</p> <p>Alle opgaven moeten in de database staan. De opgave is bestaat uit een text en een vraag. Elke opgave kan één of meerdere malen door een student worden gemaakt. Dit zijn de inzendingen. Elke inzending heeft een inleverdatum, een nakijkdatum, beoordeling (1..10) en een teller die bijhoudt de hoeveelste inzending het voor deze student betreft.</p> <p>In de data base worden de namen, studentenummers en groep van de studenten vastgelegd.</p> <p>Tot slot hou je ook namen van de docenten bij omdat elke van elke nagekeken inzending moet worden bijgehouden wie die heeft nagekeken.</p>	
--	--

Markplaats

<p>Op Markplaats heb je categorieën, in elke categorie kan een product worden aangeboden.</p> <p>Maak een ERD van een database waarin alle producten die worden aangeboden worden vastgelegd. Je legt de naam van het product vast, de verkoper, de categorie waarin het staat en de prijs waarvoor het verkocht is.</p> <p>Ook wil je bijhouden hoe lang het product op Markplaats te koop heeft gestaan. Van de verkoper wil je al zijn gegevens vastleggen; naam, adres, email en wachtwoord.</p> <p>Elke advertentie heeft 1 of meer foto's. De foto's worden ook in de database opgenomen. Verder wil je vastleggen wie welk bod heeft uitgebracht op het artikel. Elke gebruiker kan maar een keer één bod op een artikel plaatsen.</p> <p>De foto's krijgen een unieke naam en de alleen deze unieke naam hoeft in de database te worden opgenomen.</p>	
--	--

Game PC

Als verkoper van game PC's wil je een bijhouden welke onderdelen je verkoopt aan wie. Onderdelen zijn verdeeld in categorieën. Zo heb je CPU's, Graphical cards, voedingen, kasten, geheugen, SSD's, moederbord en koeling.

De prijs van een artikel kan per dag anders zijn. Je wilt in de database kunnen bijhouden hoe duur een artikel op een bepaalde datum was en je wilt ook vastleggen wat je hebt verkocht aan welke klant op welke datum.

Per onderdeel wil je ook vastleggen hoe groot de voorraad is (aantal).

[Image-1643187044290.png](#)

De Jongh Rijwielen

Fietsenmaker de Jong uit Kortenhoef wil zijn voorraadbeheer verbeteren.

De Jongh wil van elke fiets vastleggen welke reparaties er worden uitgevoerd. Hij wil per reparatie aangeven welke onderdelen er worden gebruikt. Zo weet hij precies wanneer hij weer nieuwe onderdelen moet bestellen.

Tevens wil hij dit koppelen aan de factuur voor de klant. Dus hij wil voor een klant een reparatiebon kunnen uitdraaien. Op deze bon staan de uitgevoerde reparaties en van elke reparatie de kosten. Bij de kosten van elke reparatie worden de gebruikte onderdelen vermeld.

Verder staat er op de bon voor welke fiets de reparatie is uitgevoerd (bijvoorbeeld "Rode Dames Sparta 7 versnellingen").

[Image-1643186893079.png](#)

Examen

<p>Een examen bestaat uit een aantal werkprocessen.</p> <p>Als elk werkproces van het examen met een voldoende is beoordeeld dan is de examenkandidaat geslaagd voor het examen.</p> <p>Er zijn 4 examens en elk examen heeft een ander aantal werkprocessen.</p> <p>Iedere student moet elk examen halen. Als de student een examen niet haalt dan kan hij een volgende keer een tweede poging doen.</p> <p>Maak een ERD waarin je vastlegt wanneer een examen wordt gehouden (begin- en einddatum) en leg per werkproces de resultaten vast. Van een student leg je studentnummer, naam en klas vast.</p> <p>Houd ook bij welke poging (eerste, tweede, ...) het examen is.</p>	<p>Image 1643186177308.png Download</p>
---	--

Recycle Zonnepanelen

Hieronder zie je een wat uitgebreider verhaal zoals dat in de praktijk voorkomt. Je moet nu niet alleen een ERD maken maar je moet ook uit de tekst proberen te filteren wat wel en wat neit belangrijk is voor het maken van een ERD.

Lees het verhaal door en probeer een ERD te maken. Mis je informatie vraag het dan aan de 'klant'.

SunCycle

[image.1642879664424.png](#)

Het bedrijf SunCycle (SC) handelt in zonnepanelen.

Veel klanten willen hun oude zonnepanelen vervangen door nieuwe omdat nieuwe zonnepanelen een hoger rendement hebben.

De heer Zuinig van SC heeft zijn administratie in Excel opgezet. Hij houdt de inkoop- en verkoop bij. Maar het is lastig om met met de afdeling inkoop en verkoop gelijktijdig in de administratie in Excel te werken. De heer Zuinig wil QuickResults, waar jij werkt een website laten bouwen.

De website is bedoeld voor intern gebruik.

De afdeling inkoop moet vastleggen welke zonnepanelen worden ingekocht. Het type, de hoeveelheid, de kwaliteit en de inkoopprijs moet worden vastgelegd. De inkopers kopen vrijwel alleen in van installatiebedrijven. Bij elke inkoop wil SC vastleggen van welk installatiebedrijf de panelen zijn gekocht. Worden panelen ingekocht van een particulier dan worden de gegevens van de particulier vastgelegd.

Panelen worden verwerkt. Goede panelen worden schoongemaakt, getest en verkocht als tweedehands panelen. Minder goede panelen worden ontleed in onderdelen; aluminium, aansluitdoos, kabels en silicium. Deze onderdelen worden opgeslagen.

Van alle ingekochte panelen moet worden vastgelegd wat er dus gebeurt: refurbisch en verkopen of recycling. Dit wordt per paneel bepaald.

Het recycle proces levert dus op: refurbished panelen, aluminium, aansluitdozen, kabels en silicium. Dit wordt allemaal toegevoegd aan de voorraad.

Dan is er de afdeling verkoop die aan opkopers (een deel van) de voorraad kunnen verkopen. Per verkooporder moet worden vastgelegd wat er wordt verkocht voor welke prijs. Er wordt zaken gedaan met een vast aantal opkopers. Er wordt niet verkocht aan particulieren.

Verkooporders kunnen alleen worden gemaakt indien er voldoende voorraad bestaat.

Elke verkooporder zorgt ervoor dat de voorraad van een product minder wordt.

Dhr. Zuinig wil overzichten kunnen maken van de inkoop, van de verkoop en van de actuele voorraad.

Ook wil dhr Zuinig graag kunnen berkenen wat de winst is. De winst wordt bepaald door de verkoopkosten te nemen en daar de inkoopkosten van af te halen.

Eindopdracht bestellingen en voorraad*

*Als je alle opdrachten hiervoor hebt gemaakt en goed snapt kan je aan de eindopdracht beginnen.
Als je de vorige oefeningen nog niet goed snapt vraag dan hulp.*

Eindopdracht bestellingen en voorraad

De klant geeft aan jou twee overzichten die hij graag uit zijn systeem wil kunnen halen.

Jij moet een datamodel maken en daarvoor maak je eerst een ERD.

Je kunt eerst van elk overzicht een apart een ERD maken en daarna de twee ERD's combineren. Je kunt natuurlijk ook gelijk één ERD maken.

Lastig? Kom je er niet uit?

Probeer in Excel lijstjes te maken. Het helpt soms ook om de entiteiten eerst te herkennen. Hoe herken je die?

Vraag je af waar de attributen bij horen. Zo heb je een bestelnummer, een bestel datum die beide horen bij een bestelling (of wil je die bestelbon noemen, kan ook).

Dan zie je een klant, met een naam, een klantnummer adres en woonplaats.

Een bestelling of bestelbon heeft meerdere regels met een bestelnummer, omschrijving, maat, aantal en prijs.

Zet de entiteiten in Excel en bekijk of ze aan de regels voldoen. Splits entiteiten en controleer op afgeleiden velden.

Als je klaar bent met de bestelbon doe dan de voorraad en als je die klaar hebt kijk dan of je de twee ERD's kan combineren.

Succes en vergeet niet lastiger wordt het niet meer, wel (nog) leuker!

[image-1632426661971.png](#)

image-1632426728639.png

file:///C:/Users/lenovo/Desktop/

Formule 1

Teammanagement Systeem

Inleiding

Je bent een data-analist die is ingehuurd om een databaseontwerp te maken voor een Formule 1 team. Het team wil een systeem dat hen helpt bij het beheren van coureurs, raceauto's, en de verschillende Grand Prix evenementen waar ze aan deelnemen gedurende het seizoen. Het systeem moet de volgende entiteiten en relaties bevatten:

1. **Coureurs:** Elke coureur heeft een unieke ID, naam, nationaliteit, en startnummer. Een coureur kan gedurende zijn carrière voor verschillende teams rijden.
2. **Teams:** Elk team heeft een unieke ID, naam, en land van herkomst. Een team heeft meerdere coureurs, maar elke coureur rijdt op elk moment voor slechts één team.
3. **Raceauto's:** Elke raceauto heeft een uniek serienummer, modelnaam, en bouwjaar. Een raceauto wordt toegewezen aan precies één coureur, maar een coureur kan gedurende een seizoen in verschillende auto's rijden.
4. **Grand Prix Evenementen:** Elk evenement heeft een unieke ID, naam van de Grand Prix, en het land waar het wordt gehouden. Aan elk evenement nemen meerdere coureurs deel, en elke coureur kan aan meerdere evenementen deelnemen.

Vereiste Relaties:

- **Relatie tussen Teams en Coureurs:** Elk team heeft meerdere coureurs, maar elke coureur rijdt voor slechts één team op een gegeven moment.
- **Relatie tussen Coureurs en Raceauto's:** Hoewel een coureur gedurende een seizoen in verschillende auto's kan rijden, wordt elke auto op elk gegeven moment door slechts één coureur gebruikt.
- **Relatie tussen Coureurs en Grand Prix Evenementen:** Coureurs kunnen aan meerdere evenementen deelnemen, en elk evenement heeft meerdere deelnemende coureurs.

Opdracht:

Maak een volledig ERD waarin je de volgende onderdelen opneemt.

- alle entiteiten.
- alle attributen van de entiteiten.
- alle data-typen van de attributen.
- alle sleutels (keys), dit zijn de primary keys en foreign keys.
- alle relaties tussen de entiteiten.

Inleveren

Eigen gemaakt screenshot van het ERD dat met behulp van Lucichart is gemaakt.

--