Database 1

1 Wat is een database?

datasource

? Leerdoelen

- Je weet wat een database is en waarvoor die gebruikt wordt.
- Je kunt gegevens uit de echte wereld omzetten naar tabellen en kolommen.
- Je begrijpt het verschil tussen ruwe gegevens en een gestructureerd model.

? Uitleg

In het dagelijks leven slaan organisaties gegevens op: over klanten, producten, studenten, boeken, etc. Een **database** is een digitale plek waar zulke gegevens netjes georganiseerd worden bewaard. Je verdeelt de informatie over verschillende tabellen, waarbij elke tabel over één onderwerp gaat, zoals:

- **Studenten**: naam, klas, geboortedatum
- **Docenten**: naam, vak, afdeling
- **Opleidingen**: naam, niveau, duur

Een database lijkt een beetje op een Excel-bestand, maar is veel krachtiger en beter gestructureerd. Je wilt geen dubbele gegevens en alles moet logisch met elkaar verbonden zijn.

Voorbeeld: Dit is géén goede database:

```
Naam         | Klas         | Opleiding
------------ | ------------ | -----------------
Fatima       | SD1A         | Software Developer
Ali          | SD1A         | Software Developer
Robin        | SD2B         | Software Developer
Jente        | SD1A         | Software Developer
```

```
Steven      | SD1A        | Recht & HR
```

→ Hier zie je dat de opleiding meerdere keren herhaald wordt. Dat is zonde en foutgevoelig.

◻◻ Wat zou beter zijn?

Je zou in plaats daarvan een aparte tabel 'Studenten' maken en een aparte tabel 'Opleidingen'. Studenten krijgen dan een *verwijzing* naar hun opleiding (dat komt in de volgende lessen aan bod).

?? Opdracht 1 – Gegevens analyseren

1. Bekijk onderstaande lijst met gegevens:

```
Naam: Esra
Klas: SD1A
Opleiding: Software Developer
Docent: De Jong
Vak: Webontwikkeling
Lesdag: Woensdag
```

2. Welke verschillende **onderwerpen** zie je hierin? Probeer per onderwerp de eigenschappen op te schrijven.
   - **Student**: naam, klas
   - ... (jij vult aan)
3. Zet je antwoorden in een tabelvorm: welke tabellen zou je nodig hebben? Welke kolommen zouden erin staan?

? Reflectie

- Waarom is het vastleggen van gegevens op meerdere plaatsen foutgevoelig? Beschrijf een situatie waarin dat fout kan gaan.
- Waarom is het niet handig om alle informatie in één grote tabel te zetten?
- Wat denk je dat het voordeel is van losse tabellen met verbindingen?

? Inleveren

- Lever je tabellenindeling in (.txt, .pdf of screenshot).

- Voeg je antwoorden toe op de reflectievragen.

2 Entiteiten en Attributen

? Leerdoelen

- Je weet wat een entiteit is en wat een attribuut is.
- Je kunt entiteiten en hun attributen herkennen in een realistisch scenario.
- Je kunt een eerste versie van een ERD tekenen met entiteiten en attributen.

? Uitleg

◻◻ Entiteit

In de vorige opdracht heb je '**onderwerpen**' en '**eigenschappen**' bepaald, weet je het nog? Een onderwerp was 'student' en een eigenschap was 'naam' en 'klas'.

Bij het ontwerpen van een database heet een onderwerp een 'entiteit', en een eigenschap is een ' attribuut'

> Een **entiteit** is een "***ding***", "***persoon***" of "***gebeurtenis***" in de echte wereld waarover je gegevens wilt opslaan.

Bijvoorbeeld:

- `Student` -> persoon
- `Cursus` -> gebeurtenis
- `Docent` -> persoon
- `Opleiding` -> gebeurtenis
- `Laptop` -> ding

◻◻ Attribuut

Een **attribuut** is een eigenschap van een entiteit. Bijvoorbeeld:

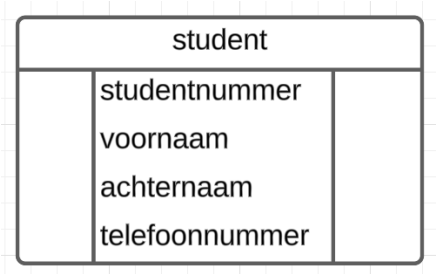
- De entiteit `Student` heeft de attributen: `studentnummer` `voornaam`, `achternaam`, `telefoonnummer`.
- De entiteit `Cursus` heeft de attributen: `naam`, `duur`, `startdatum`.

□□ ERD

Een entiteit met de attributen zet je in een bepaald formaat in een database ontwerp. Dat heet een ERD.

Op de eerste regel staat de **naam van de entiteit** en in de middelste kolom zet je alle **attributen** .

Voorbeeld:



□□ In een ERD (Entity Relationship Diagram) teken je entiteiten als rechthoeken en attributen als ovale of gelabelde velden ernaast.

> In een ERD heb je **drie kolommen**
> de eerste voor de **keys**, de tweede voor de **attribuutnamen** en de derde voor de **datatypes**.

?? Opdracht 1 – Entiteiten en attributen

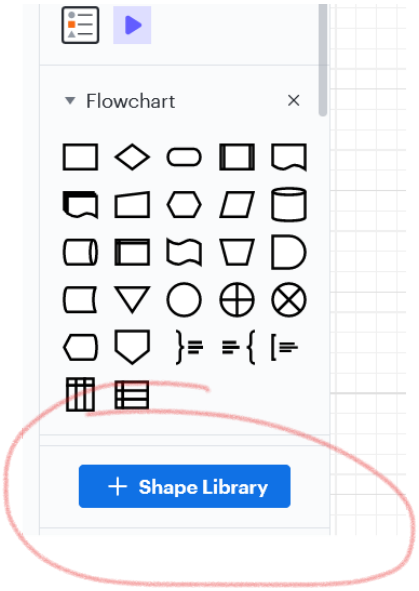
1. Je werkt voor een evenementenbureau. Zij willen bijhouden:
   - Welke klanten boekingen doen
   - Welke evenementen er zijn
   - Welke locaties beschikbaar zijn
2. Maak een lijstje van minstens 3 entiteiten uit deze situatie. Bijvoorbeeld:
   - `Klant`
   - ...
   - ...

3. Geef per entiteit minstens 3 bijpassende attributen. Bijv.:
    - Klant → voornaam, e-mailadres, ...
    - Locatie → naam, ...
4. Teken jouw eerste ERD in Lucidchart.

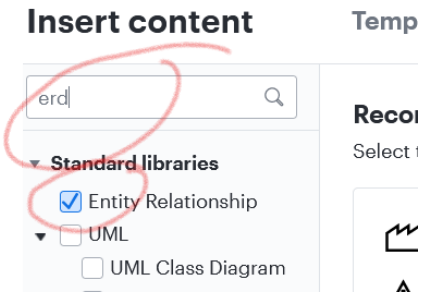
Lucichart

Registreer je voor Lucichart en maak een gratis account.

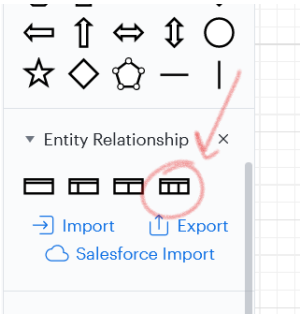
Kies toevoegen library



Zoek naar ERD en selecteer "Entity Relationship"



Gebruik vervolgens dit figuur om een entiteit te maken.



? Reflectie

- Hoe weet je of iets een entiteit was of gewoon een attribuut?
- Heb je misschien dingen dubbel in verschillende entiteiten? Kun je iets beter loskoppelen?

? Inleveren

- Maak je ERD in Lucichart en maak een screenshot.

3 Primary Keys

? Leerdoelen

- Je weet wat een **primary key** is.
- Je begrijpt waarom een primary key verplicht is in elke tabel.
- Je kunt per entiteit een geschikte primary key kiezen.

? Uitleg

□□ Wat is een Primary Key (PK)?

Een primary key is een uniek gegeven waarmee je één rij uit een tabel kunt identificeren. Elke tabel in een database **moet** een primary key hebben.

Regel: Elke entiteit heeft precies één PK. Heb je meerdere opties? Kies er één. Heb je geen goede kandidaat? Gebruik dan een kunstmatige sleutel, zoals een `id`.

Voorbeelden:

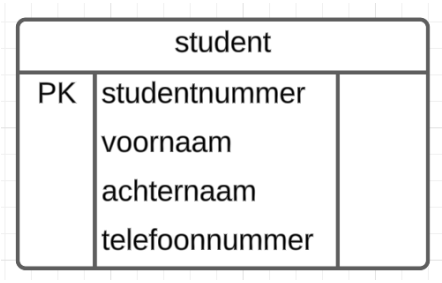
- `Student` → `student_nr`
- `Auto` -> `kenteken`
- `Evenement` → `evenement_id`
- `Klant` → email_adres

Een primary key moet:

- Uniek zijn
- Nooit leeg zijn
- Vast blijven (mag niet wijzigen)

Als je later tabellen met elkaar verbindt, gebruik je de PK om een verbinding te maken van de ene entiteit naar de andere entiteit.

□□ Voorbeeld van een ERD met PK



?? Opdracht 1 – Kies je primary keys

1. Gebruik het ERD van de vorige opdracht met de entiteiten: `evenement`, `klant` en `locatie`.
2. Voeg aan elke entiteit een geschikte primary key toe. Kies daarbij een bestaand aatribuut of voeg zelf een sleutel toe.
3. Zet in de eerste kolom van je ERD de letters PK om aan te geven dat deze regel de PK bevat.
4. Controleer: is jouw PK echt uniek en onveranderlijk?

? Reflectie

- Welke van je gekozen PK’s is natuurlijk (bestaand gegeven), en welke kunstmatig (gegenereerd ID)?

- Wat zou er misgaan als je geen PK kiest?

? Inleveren

- Lever een screenshot van je ERD in met in de eerste kolom de PK's (primary keys).

4 1:N-relaties en Foreign Keys

? Leerdoelen

- Je weet wat een 1:N-relatie is in een database.
- Je begrijpt wat een foreign key (FK) is en waarvoor die dient.
- Je kunt zelf een 1:N-relatie modelleren met een foreign key.

? Uitleg

□□ Wat is een 1:N-relatie?

Bij een **één-op-veel-relatie** (1:N) hoort bij één rij in de ene tabel, meerdere rijen in de andere tabel.

Bijvoorbeeld:

- `1 Klant` → `veel Boeking(en)`
- `1 Docent` → `veel Cursussen`

De **“meer”-kant** krijgt de foreign key (FK). De FK is een kopie van de primary key (PK) van de andere tabel.

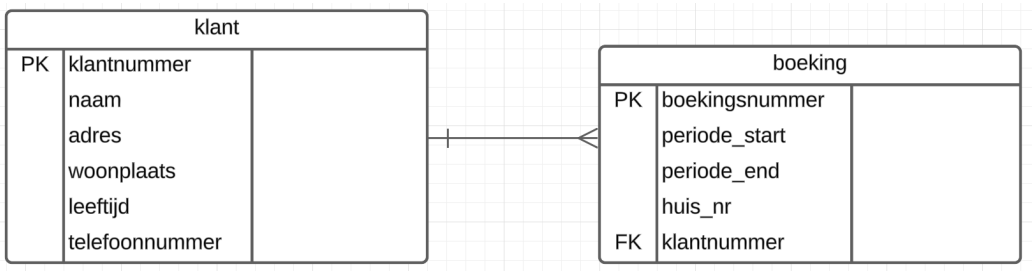
> □□ De regel: **Meer = harkje = FK**

- Teken het harkje aan de kant waar “veel” is
- In die tabel voeg je de foreign key toe
- Voorbeeld: `boekingen` krijgt `klant_id` als FK

◫ Voorbeeld

Een klant kan veel boekingen hebben (andersom kan niet!). Dus het harkje komt aan de kant van de boeking.

Bij elk harkje hoort een **FK** die verwijst naar de **PK** van de entiteit waarmee die is gekoppeld. In dit geval dus klantnummer.



◫ Het `klantnummer` in boeking is een *foreign key* die verwijst naar klant.

?? Opdracht 1 – Relaties en foreign keys

1. Gebruik het ERD van de vorige opdracht met de entiteiten: `evenement`, `klant` en `locatie`.
2. Bepaal de veel kant van klant en evenement (evenement zou een concert kunnen zijn). Je wilt bijhouden welke evenementen door een klant worden bezocht.
3. Bepaal de veel kant van evenement en locatie
4. Teken de entiteiten en de relaties. Zet de harkjes aan de juiste kant.
5. Voeg foreign keys toe (FK's) en plaats in de eerste kolom een FK bij elke Foreign key.

? Inleveren

- Lever je een screenshot van de ERD's inclusief foreign keys en erelaties in.

5 Datatypes en Validatie

? Leerdoelen

- Je kent de meest gebruikte datatypes in een database.
- Je kunt passende datatypes kiezen voor attributen in je ERD.

- Je weet waarom validatie belangrijk is bij het kiezen van datatypes.

? Uitleg

□□ Wat is een datatype?

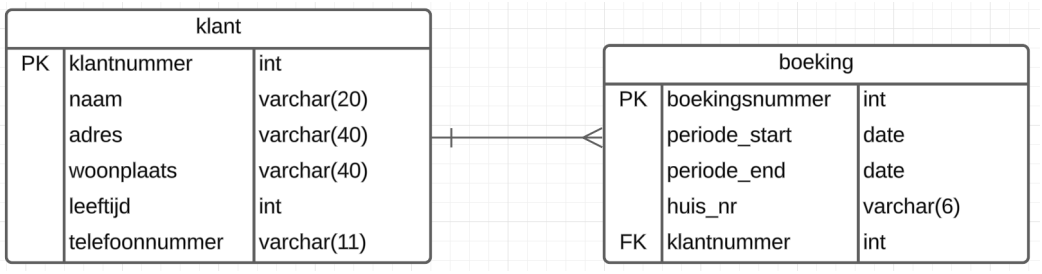
Een datatype bepaalt welk soort informatie je in een kolom/tabel opslaat. Het zorgt ervoor dat je database weet hoe de data eruitziet en wat ermee mag gebeuren.

Niet alle data-typen zijn even snel als je ze in een database gebruikt. Als je één van de volgende veelgebruikte datatypes gebruikt dan zit je meestal goed. Wil je andere gebruiken zoek dan goed uit wat de nadelen zijn.

□□ Veelgebruikte datatypes:

Datatype	Gebruik	Voorbeeld
`INT`	Voor hele getallen (bijv. ID's, aantallen)	`5`, `142`
`VARCHAR(50)`	Voor tekst tot X karakters	`'Jan'`, `'email@example.com'`
`DATE`	Voor datums zonder tijd	`'2025-06-06'`
`DATETIME`	Voor datum én tijd	`'2025-06-06 14:30:00'`
`DECIMAL(6,2)`	Voor getallen met komma (bijv. prijs)	`12.99`, `9999.00`

□□ Voorbeeld



□□ Tips voor goede keuzes:

- Let op lengte bij `VARCHAR`: hoe langer, hoe trager, maar te kort is ook niet goed.
- Een telefoonnummer is een `VARCHAR()` en geen `INT`, waarom?
- Stel je wil een bedrag van maximaal 9999,99 opslaan dan gebruik je decimal(6,2).

?? Opdracht 1 – Pas je datatypes aan

1. Gebruik het ERD van de vorige opdracht met de entiteiten: `evenement`, `klant` en `locatie`.
2. Voeg de volgende attributen toe:
   - evenement -> toegangsprijs, aanvangsdatum en aanvangstijd
3. Kies voor elk attribuut een passend datatype uit de tabel hierboven.
4. Schrijf de datatypes in de derde kolom van de ERD's..
5. Gebruik minimaal 3 verschillende soorten datatypes; je mag er zelf attributen bij bedenken.

? Inleveren

- Lever je bijgewerkte ERD in met daarin duidelijk per attribuut het datatype genoteerd en de FK.

6 Case – Modelleer een realistisch scenario

? Leerdoelen

- Je kunt zelfstandig entiteiten en relaties herkennen in een realistisch scenario.
- Je past de 5 stappen toe om een ERD te maken.
- Je maakt een logisch en technisch correct ERD met PK’s, FK’s en datatypes.

Herhaling, de 5 basisregels

1. Een **entiteit** is een persoon, ding of gebeurtenis. Een getal of bedrag (bijvoorbeeld gewicht) is nooit een entiteit, maar altijd een attribuut (=eigenschap) van een entiteit.
2. Elke entiteit heeft precies één **PK (primary key)**. De primary key maakt de entiteit uniek (bijvoorbeeld kenteken van een auto).
3. Entiteiten hebben de volgende **relaties** 1:1, 1:N, N:1 of N:M.
   - 1:1 relaties bestaan bijna niet, als ze voorkomen dan kun je de relaties samenvoegen.

- 1:N en N:1 is eigenlijk hetzelfde en komen het meest voor.

4. Een **1:N** relatie verbind je met een lijntje met een 'harkje'. Het **lijntje** staat aan de 1-kant en het '**harkje**' staat aan de meer-kant.
5. Bij elk 'harkje' hoort precies één FK. De FK verwijst naar de PK van de table waarmee deze is verbonden.

? Uitleg

Tot nu toe heb je geleerd wat entiteiten, attributen, PK's, FK's, datatypes en 1:N-relaties zijn. Nu pas je alles toe op een echte situatie.

□□ Scenario: Fietsenmaker Snelle Jelle

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

Omdat de 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.

□□ Je hebt de volgende entiteiten

- **reparatie**: ....
- **klant**: ....
- **fiets**: ....

?? Opdracht – Maak het ERD

1. Bepaal per entiteit eerst alle attributen, lees daarvoor goed het scenario door!
2. Teken de drie entiteiten in Lucichart
3. Zet alle attributen in de entiteiten
4. Bepaal de Primary Key (PK).
5. Bepaal de data-types.

6. Bedenk wat de relaties zijn en teken die met het hartje aan de goede kant.
7. Bepaal de Foreign Key (FK)

? Inleveren

- Lever een screenshot in van je ERD gemaakt in Lucichart. Zorg dat alles goed leesbaar is.

7 Meerdere relaties en N:N

? Leerdoelen

- Je begrijpt wat een N:N-relatie is en wanneer die voorkomt.
- Je kunt een N:N-relatie correct omzetten naar aparte tabellen met FK’s.
- Je kunt meerdere relaties per entiteit modelleren in een ERD.

? Uitleg

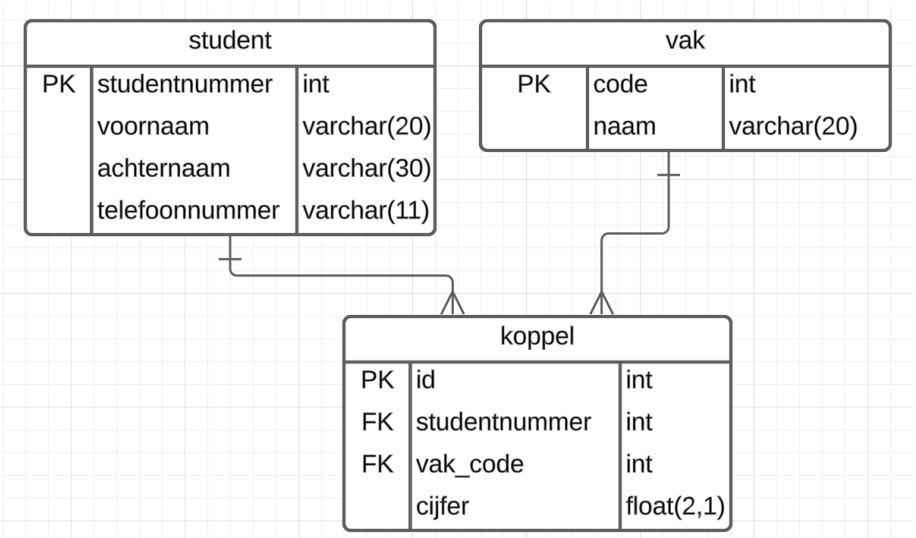
□□ Wat is een N:N-relatie?

Een **veel-op-veel** (N:N) relatie komt voor wanneer meerdere records uit entiteit A gekoppeld kunnen zijn aan meerdere records uit entiteit B.

Voorbeeld: Studenten kunnen zich inschrijven voor meerdere vakken. Elk vak kan meerdere studenten hebben.

□□ Hoe modelleer je dit?

Je maakt een extra tabel tussen de twee entiteiten. Deze bevat alleen de foreign keys van beide kanten.



De tussentabel (ook koppeltabel genoemd) bevat meestal ook extra informatie, zoals de inschrijfdatum of het cijfer. Deze extra informatie gaat over de combinatie student-vak. Een student heeft geen cijfer, een vak heeft geen cijfer, maar de combinatie student-vak heeft wel een cijfer.

□□ Meerdere relaties op één entiteit?

Soms is een entiteit aan meerdere andere entiteiten verbonden.

Voorbeeld: Een **medewerker** werkt in een afdeling, maar ook aan meerdere projecten.

- 1 medewerker ↔ 1 afdeling → 1:N
- 1 medewerker ↔ meerdere projecten ↔ N:N

Gebruik verschillende relaties als het logisch is dat een entiteit meerdere rollen vervult.

?? Opdracht – N:N-model

1. Maak een ERD voor dit scenario:
   Een muziekschool organiseert **lessen**. **Leerlingen** kunnen zich inschrijven op meerdere lessen. Elke **les** wordt gevolgd door meerdere leerlingen.
2. Bedenk zelf voor elke entiteit minimaal 4 attributen.
3. Modelleer de juiste entiteiten, attributen, PK’s, FK’s en datatypes.
4. Voeg een tussentabel toe om de N:N-relatie correct te verwerken.

? Reflectie

- Waarom is een tussentabel nodig bij N:N?
- Welke extra informatie kun je in de tussentabel kwijt?
- Waar moet je op letten bij het toevoegen van meerdere relaties in je ERD?

? Inleveren

- Lever een screenshot in van je ERD gemaakt in Lucichart. Zorg dat alles goed leesbaar is.
- Beantwoord de reflectievragen (pdf of txt bestand)

8 Bibliotheek

□□ Scenario: Bibliotheek

In een bibliotheek wil men bijhouden welke klanten welk boek van welke periode tot periode hebben geleend.

Elke klant kan meerdere boeken gelijktijdig lenen. Verder wil men de klant een whatsapp kunnen sturen twee dagen voor het verstrijken van de inleverdatum.

- Maak een databaseontwerp (ERD).

? Inleveren

- Lever een screenshot in van je ERD gemaakt in Lucichart. Zorg dat alles goed leesbaar is.

--

Aanpassen 2026, het voorbeeld klant booking locatie is eigenlijk een N:M relatie.
- neem ander voorbeeld: klant - order - product
- vermeld de 5 stappen om een ERD te maken duidelijker, maak die structuur helderde.

Revision #19
Created 2025-06-06 13:55:57 UTC by Max
Updated 2026-02-16 11:02:59 UTC by Max