COLLEGE 1

Relationele databases

Eenvoudige queries met 1 tabel
SELECT … FROM … WHERE …

LES 1

LES 1

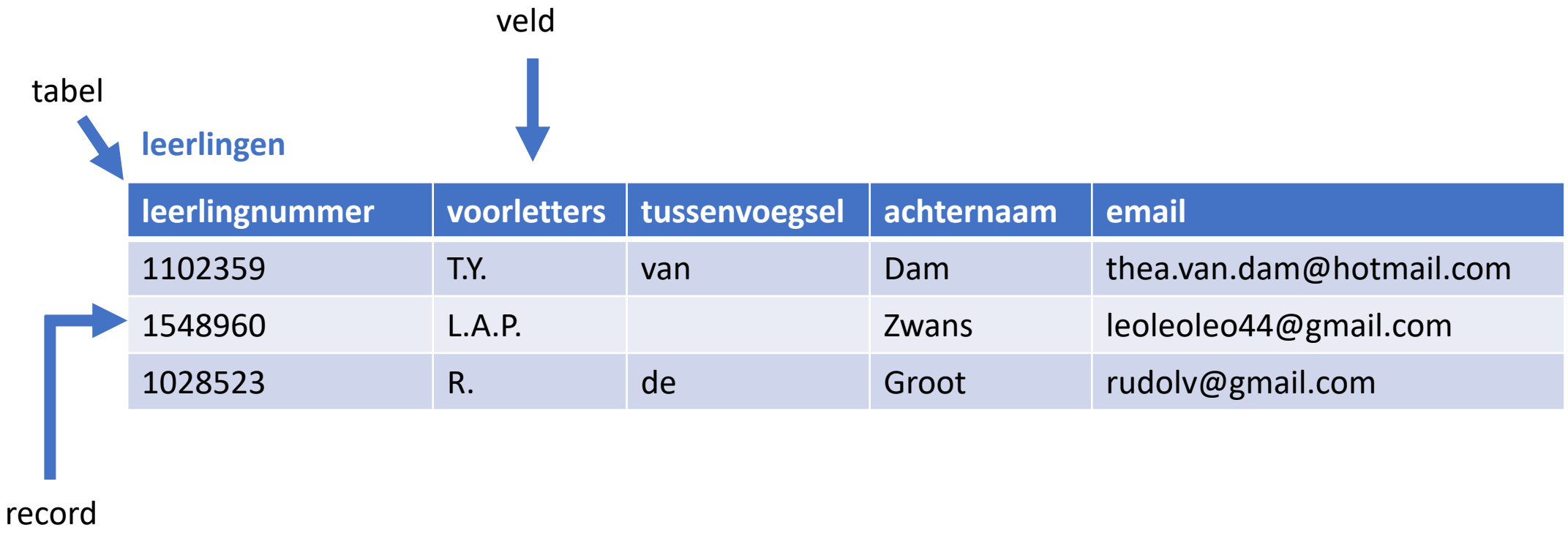
In databases kun je heel veel gegevens opslaan

Ken je voorbeelden waarbij databases gebruikt worden?

Magister, webwinkels, routeplanner, banken, Funda …

Relationele databases

- Informatie is geordend in tabellen



leerlingnummer	voorletters	tussenvoegsel	achternaam	email
1102359	T.Y.	van	Dam	thea.van.dam@hotmail.com
1548960	L.A.P.		Zwans	leoleoleo44@gmail.com
1028523	R.	de	Groot	rudolv@gmail.com

LES 1

Relationele databases

- Alle opgevraagde informatie komt als tabel

leerlingnummer	voorletters	tussenvoegsel	achternaam	email
1102359	T.Y.	van	Dam	thea.van.dam@gmail.com
1548960	L.A.P.		Zwans	leoleoleo44@outlook.com
1028523	R.	de	Groot	rudolv@gmail.com

"geef de velden voorletters, tussenvoegsel en achternaam van alle records waarvan het veld email eindigt op gmail.com"

voorletters	tussenvoegsel	achternaam
T.Y.	van	Dam
R.		Groot

LES 1

Relationele databases

- **Alle** opgevraagde informatie komt als tabel, ook als het één waarde is.

leerlingnummer	voorletters	tussenvoegsel	achternaam	email
1102359	T.Y.	van	Dam	thea.van.dam@hotmail.com
1548960	L.A.P.		Zwans	leoleoleo44@gmail.com
1028523	R.	de	Groot	rudolv@yahoo.com

“geef het veld leerlingnummer
van het record dat het hoogste leerlingnummer
heeft van alle records in de tabel”

leerlingnummer
1548960

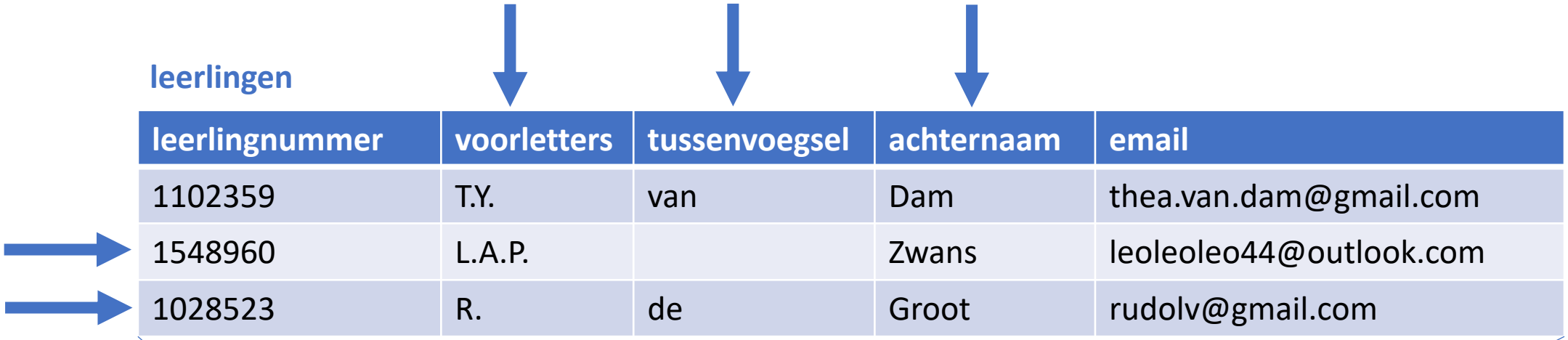
LES 1

SQL (Structured Query Language)

- Dé taal voor het werken met relationele databases.
- Aantal SQL-dialecten, maar de basis is hetzelfde.
  - `SELECT` -> opvragen van informatie
  - `UPDATE` -> wijzigen van velden
  - `DELETE` -> verwijderen van records
  - `INSERT` -> toevoegen van records
  - `CREATE` -> tabel maken

- Een opdracht in de taal SQL noem je een **query**

Eenvoudige SQL-opdracht



leerlingen

leerlingnummer	voorletters	tussenvoegsel	achternaam	email
1102359	T.Y.	van	Dam	thea.van.dam@gmail.com
1548960	L.A.P.		Zwans	leoleoleo44@outlook.com
1028523	R.	de	Groot	rudolv@gmail.com

```
SELECT voorletters, tussenvoegsel, achternaam
  FROM leerlingen
  WHERE email LIKE "%gmail.com";
```

voorletters	tussenvoegsel	achternaam
T.Y.	van	Dam
R.		Groot

LES 1

Begrippen

- Een **tabel** is een groep gegevens in een database die in rijen en kolommen geordend zijn. Een tabel heeft een naam.
- Een **record** is een rij met gegevens in een tabel van een database.
- Een **veld** is de kop van een kolom van een tabel. Een veld heeft een naam en een gegevenstype. Het gegevenstype beschrijft wat voor soort gegevens er in het veld staan, bijvoorbeeld tekst of een geheel getal.
- Een **waarde** is een gegeven dat in bepaald veld in een bepaald record is ingevuld.
- Een **primaire sleutel** is een veld waarvan de waarde uniek is voor elk record in de tabel. Een primaire sleutel heeft meestal de naam “id” en de waarde is een geheel getal. De primaire sleutel wordt gebruikt om een record aan te wijzen.
- Een **verwijzende sleutel** is een veld waarvan de waarde verwijst naar een primaire sleutel in een andere tabel. Een verwijzende sleutel heeft dus een waarde die overeenkomt met de primaire sleutel van een andere tabel.

Queries met functies en groepen
GROUP BY … HAVING …

LES 2

Berekeningen uitvoeren in SQL

- Soms wil je niet alleen rauwe gegevens uit een database, maar informatie die met behulp van de losse gegevens kan worden verkregen:
  - het aantal leerlingen in 5V
  - de leerling met het hoogste cijfer
  - het artikel met de laagste prijs
  - de totale reiskosten in maart
  - het gemiddelde cijfer voor het PWS
- SQL heeft hiervoor speciale functies die je in een query kunt gebruiken: COUNT, MIN, MAX, SUM, AVG

LES 2

Rijen samenvoegen

- Met functies kun je rijen samenvoegen

id	naam	voorraad	categorie	prijs	COUNT(*)
~~1~~	~~washandje~~	~~19~~	~~badlinnen~~	~~1,99~~	5
~~2~~	~~handdoek~~	~~23~~	~~badlinnen~~	~~4,99~~	
~~3~~	~~gastendoek~~	~~33~~	~~badlinnen~~	~~2,49~~	
~~4~~	~~wok 23cm~~	~~12~~	~~kookgerei~~	~~29,99~~	
~~5~~	~~USB-kabel 2m~~	~~34~~	~~elektronica~~	~~3,99~~	

```
“tel het aantal rijen (artikelen)”

SELECT COUNT(id) FROM Artikel
```

COUNT(id)
5

LES 2

Rijen samenvoegen

- Sommige queries kunnen maar zijn onzin

id	naam	voorraad	categorie	prijs	COUNT(*)
~~1~~	~~washandje~~	~~19~~	~~badlinnen~~	~~1,99~~	5
~~2~~	~~handdoek~~	~~23~~	~~badlinnen~~	~~4,99~~	
~~3~~	~~gastendoek~~	~~33~~	~~badlinnen~~	~~2,49~~	
~~4~~	~~wok 23cm~~	~~12~~	~~kookgerei~~	~~29,99~~	
5	USB-kabel 2m	~~34~~	~~elektronica~~	~~3,99~~	

```
SELECT naam, COUNT(id) FROM Artikel
```

Een willekeurige rij met naam wordt gegeven

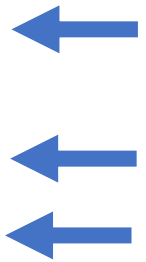
naam	COUNT(id)
USB-kabel 2m	5

LES 2

Rijen groeperen

- Met GOUP BY kun je rijen groeperen

id	naam	voorraad	categorie	prijs	COUNT(*)
1	washandje	19	badlinnen	1,99	3
~~2~~	~~handdoek~~	~~23~~	~~badlinnen~~	~~4,99~~	
~~3~~	~~gastendoek~~	~~33~~	~~badlinnen~~	~~2,49~~	
4	wok 23cm	12	kookgerei	29,99	1
5	USB-kabel 2m	34	elektronica	3,99	1



"tel het aantal rijen (artikelen) per catergorie"

```
SELECT COUNT(id) FROM Artikel GROUP BY categorie
```

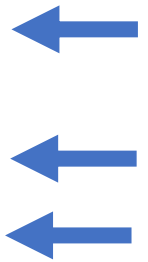
COUNT(id)
3
1
1

LES 2

Gegroepeerde rijen filteren

- Met HAVING kun je gegroepeerde rijen filteren

id	naam	voorraad	categorie	prijs	COUNT(*)
1	washandje	19	badlinnen	1,99	3
~~2~~	~~handdoek~~	~~23~~	~~badlinnen~~	~~4,99~~	
~~3~~	~~gastendoek~~	~~33~~	~~badlinnen~~	~~2,49~~	
~~4~~	~~wok 23cm~~	~~12~~	~~kookgerei~~	~~29,99~~	~~1~~
~~5~~	~~USB-kabel 2m~~	~~34~~	~~elektronica~~	~~3,99~~	~~1~~



“geef de categorie met meer dan 1 artikel per caterogie”

```
SELECT COUNT(id) FROM Artikel GROUP BY categorie HAVING COUNT(id) > 1
```

COUNT(id)
3

LES 2

Samenvatting

Velden (kolommen) en records (rijen) selecteren

- `SELECT <velden> FROM <tabel> WHERE <filter>`

Records groeperen

- `SELECT <velden> FROM <tabel> GROUP BY <veld>`

Records groeperen en berekende velden toevoegen

- `SELECT <velden>, <berekende velden> FROM <tabel> GROUP BY <veld>`

Records groeperen en berekende velden toevoegen en records selecteren

- `SELECT <velden>, <berekende velden> FROM <tabel> GROUP BY <veld> HAVING <filter>`

Onthoud: met GROUP BY gebruik je HAVING (niet WHERE)

LES 2

Relaties tussen tabellen (1:n)
JOIN … ON …

LES 3

Redundantie voorkomen

Als je alle gegevens uit je database in één tabel zet, dan zie je al snel dat dezelfde informatie meerdere keren voorkomt. Dat noemen we **redundantie**.

Redundantie is niet wenselijk in een database, want:

- Het kost veel opslagruimte
- Er is meer kans op fouten in je database, bijvoorbeeld als iets twee keer in de database staat en je maakt op één plek een typefout, dan zie je niet meer dat beide gegevens eigenlijk hetzelfde zijn.
- Bij het wijzigen van gegevens is het onhandig dat je dat op meerdere plaatsen moet doen

Redundantie kun je voorkomen door gegevens in meerdere tabellen op te slaan. Met verwijzende sleutels aan te geven welke gegevens uit verschillende tabellen bij elkaar horen. Dit noemen we **normaliseren**.

Bij het opvragen van gegevens uit een database met een SQL-opdracht (query), kun je tabellen tijdelijk samenvoegen met JOIN.

Met JOIN kun je uit een genormaliseerde database (met meerdere tabellen) dus dezelfde informatie halen als uit een database waarin redundantie zit (met één tabel).

Normeren

dubbele informatie (redundantie)

- **Normeren** is het splitsen van een tabel in meerdere tabellen, zodat dubbele informatie maar één keer wordt opgeslagen.

leerlingnummer	voornaam	code	cijfers
101279	Joop	ak-T01	8.4
139002	Andrea	in-P03	7.2
139002	Andrea	bi-T02	5.1
101279	Joop	in-P03	4.8
…	…	…	…

1. Splits tabel
2. Voeg primaire sleutels toe (id)
3. Voeg verwijzende sleutel toe (leerling_id)

leerlingen

id	leerlingnummer	voornaam
1	101279	Joop
2	139002	Andrea

cijfers

id	code	cijfer	leerling_id
1	ak-T01	8.4	1
2	in-P03	7.2	2
3	bi-T02	5.1	2
4	in-P03	4.8	1

LES 3

Tabellen weer combineren met JOIN

leerlingen

id	leerlingnummer	voornaam
1	101279	Joop
2	139002	Andrea

cijfers

id	code	cijfer	leerling_id
1	ak-T01	8.4	1
2	in-P03	7.2	2
3	bi-T02	5.1	2
4	in-P03	4.8	1

uitgebreide tabel

SELECT * FROM leerlingen JOIN cijfers

id	leerlingnummer	voornaam	id	code	cijfer	leerling_id
1	101279	Joop	1	ak-T01	8.4	1
1	101279	Joop	2	in-P03	7.2	2
1	101279	Joop	3	bi-T02	5.1	2
1	101279	Joop	4	in-P03	4.8	1
2	139002	Andrea	1	ak-T01	8.4	1
2	139002	Andrea	2	in-P03	7.2	2
2	139002	Andrea	3	bi-T02	5.1	2
2	139002	Andrea	4	in-P03	4.8	1

JOIN combineert alle rijen van de ene tabel met alle rijen van de andere tabel,
maar ik heb niets aan rijen met leerlinginformatie van Joop met cijfers van Andrea of andersom

Tabellen weer combineren met JOIN

leerlingen

id	leerlingnummer	voornaam
1	101279	Joop
2	139002	Andrea

cijfers

id	code	cijfer	leerling_id
1	ak-T01	8.4	1
2	in-P03	7.2	2
3	bi-T02	5.1	2
4	in-P03	4.8	1

SELECT * FROM leerlingen JOIN cijfers ON leerlingen.id = cijfers.leerling_id

id	leerlingnummer	voornaam	id	code	cijfer	leerling_id
1	101279	Joop	1	ak-T01	8.4	1
~~1~~	~~101279~~	~~Joop~~	~~2~~	~~in-P03~~	~~7.2~~	~~2~~
~~1~~	~~101279~~	~~Joop~~	~~3~~	~~bi-T02~~	~~5.1~~	~~2~~
1	101279	Joop	4	in-P03	4.8	1
~~2~~	~~139002~~	~~Andrea~~	~~1~~	~~ak-T01~~	~~8.4~~	~~1~~
2	139002	Andrea	2	in-P03	7.2	2
2	139002	Andrea	3	bi-T02	5.1	2
~~2~~	~~139002~~	~~Andrea~~	~~4~~	~~in-P03~~	~~4.8~~	~~1~~

JOIN … ON combineert alle rijen van de ene tabel met alle rijen van de andere tabel EN filtert rijen eruit waarvan de gegevens uit de ene en andere tabel bij elkaar passen

Tabellen weer combineren met JOIN

leerlingen

id	leerlingnummer	voornaam
1	101279	Joop
2	139002	Andrea

cijfers

id	code	cijfer	leerling_id
1	ak-T01	8.4	1
2	in-P03	7.2	2
3	bi-T02	5.1	2
4	in-P03	4.8	1

SELECT voornaam, cijfer FROM leerlingen JOIN cijfers ON leerlingen.id = cijfers.leerling_id

voornaam	cijfer
Joop	8.4
Andrea	8.4
Andrea	5.1
Joop	4.8

JOIN … ON combineert alle rijen van de ene tabel met alle rijen van de andere tabel EN filtert rijen eruit waarvan de gegevens uit de ene en andere tabel bij elkaar passen

LES 4

Relaties tussen tabellen (n:m)
JOIN … ON …

LES 3

Vakkenpakket database

id	leerlingnummer	voornaam
1	101279	Joop
2	139002	Andrea

id	vak	vakbeschrijving
1	ak	nuttig
2	in	Het mooiste vak
3	bi	leuk

SELECT * FROM leerlingen JOIN gekozen
ON leerlingen.id = gekozen.leerling_id
JOIN vakken
ON vakken.id = gekozen.vakken_id

Gekozen

id	Leerling_id	Vak_id
1	1	1
2	1	2
3	2	2
4	2	3

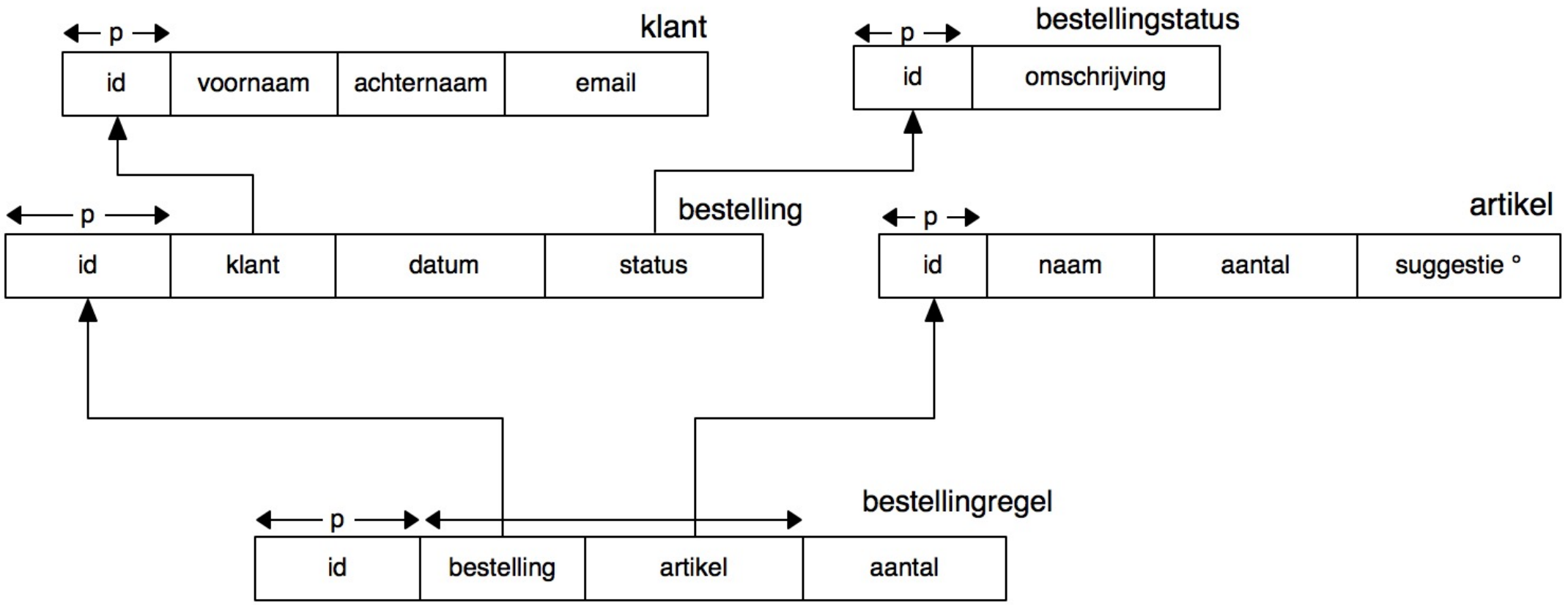
id	leerlingnummer	voornaam	id	vak	vakbeschrijving
1	101279	Joop	1	ak	nuttig
1	101279	Joop	2	in	Het mooiste vak
2	139002	Andrea	2	in	Het mooiste vak
2	139002	Andrea	3	bi	leuk

Voor een n:m relatie heb je een extra tabel nodig

LES 4

Vraag: welke relaties zijn er?

Bestelling:klant = n:1
Bestelling:bestellingstatus = n:1
Klant:bestellingstatus: n:m

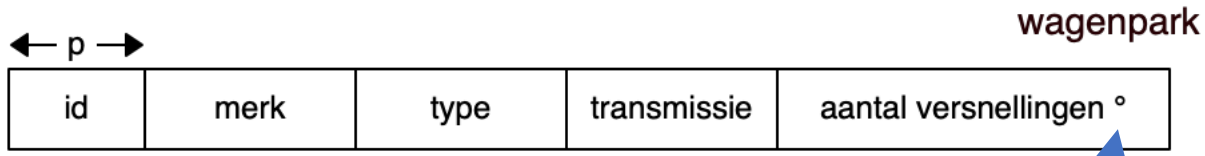


Aandachtspunten
NULL

LES 7

Optionele velden

Soms ontkom je niet aan het feit dat een veld leeg blijft.



Optioneel veld geef je aan met een °

id	merk	type	transmissie	aantal versnellingen
1	Opel	Astra 1.6	handmatig	6
2	Opel	Zafira 1.8	handmatig	5
3	Tesla	Model 3	automaat	NULL
…	…	…	…	…

NULL betekent “geen waarde” of “onbekend”

Is dus niet hetzelfde als 0 of “” (lege string)

LES 7

Even opfrissen…

SELECT <velden> FROM <tabel>	Informatie opvragen
JOIN <tabel> ON …	Tabel uitbreiden met informatie uit andere tabel
WHERE <voorwaarde>	Rijen filteren met behulp van een voorwaarde
GROUP BY <veld>	Bij geaggregeerde functies (zoals MAX(…)) groeperen van resultaten
HAVING <voorwaarde>	Groepen filteren
ORDER BY <veld>	Rijen sorteren op basis van een veld

LES 7

Stappenplan Queries maken

1. Bekijk de database
   A. bekijk de gegevens in alle tabellen,
   B. lees de tekst die erbij staat zodat je snapt hoe de tabellen aan elkaar gerelateerd zijn.
2. Welke velden willen we zien?
   A. Welke velden uit welke tabel?
   B. Welke velden die ik moet berekenen?
   C. Tip: gebruik sum, count enzovoort om te berekenen
3. Welke tabellen heb ik nodig?
   A. Alle tabellen waaruit ik velden gebruik EN
   B. Tabellen die nodig zijn om “te verbinden”
   C. Tip: Gebruik JOIN en ON bij meer dan 1 tabel
4. Moet ik rijen groeperen?
   A. Tip: vaak (niet altijd) nodig als er “per” in de vraag staat en als er berekende velden zijn
   B. Gebruik GROUP BY
5. Moet ik rijen filteren
   A. Gebruik HAVING (met GROUP BY) of WHERE
6. Moet ik sorteren?
   A. Gebruik ORDER BY

COLLEGE 1

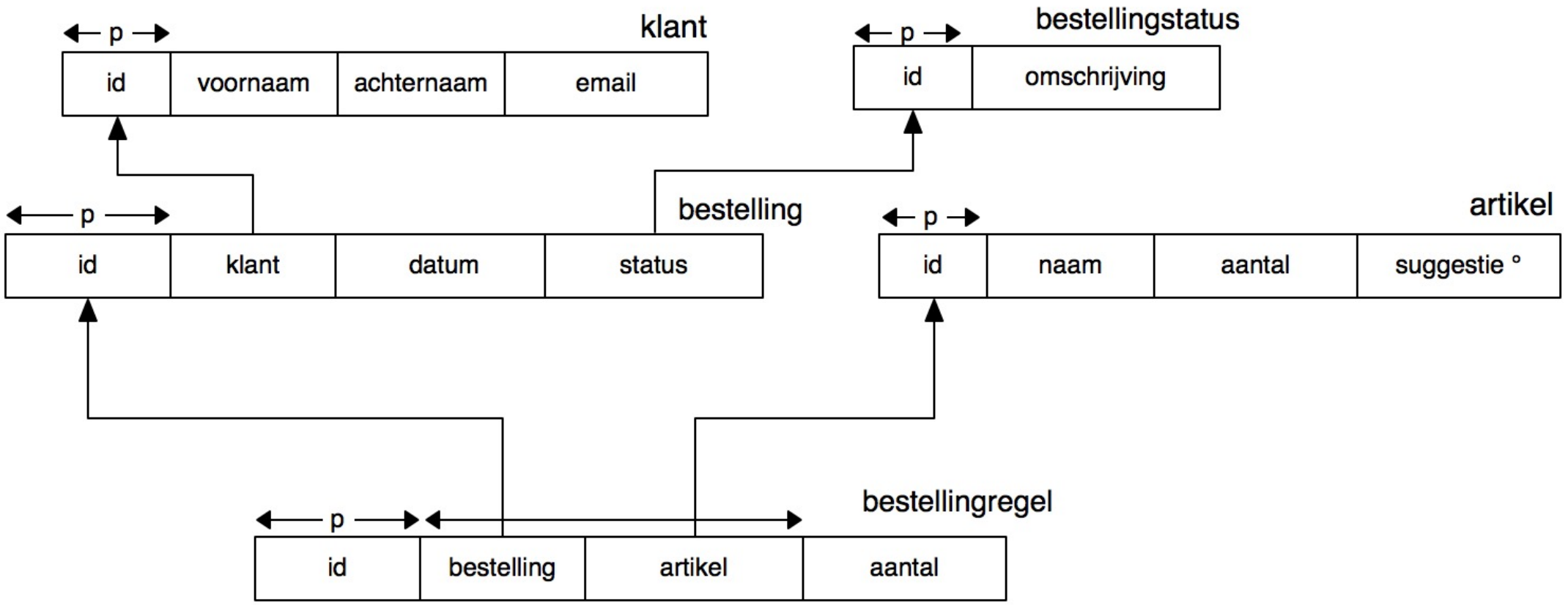
Relationele databases
OEFENINGEN

Het gebruik van JOIN VOORBEELD met 2 tabellen

LES 4

Vraag: welke tabellen heb ik nodig?

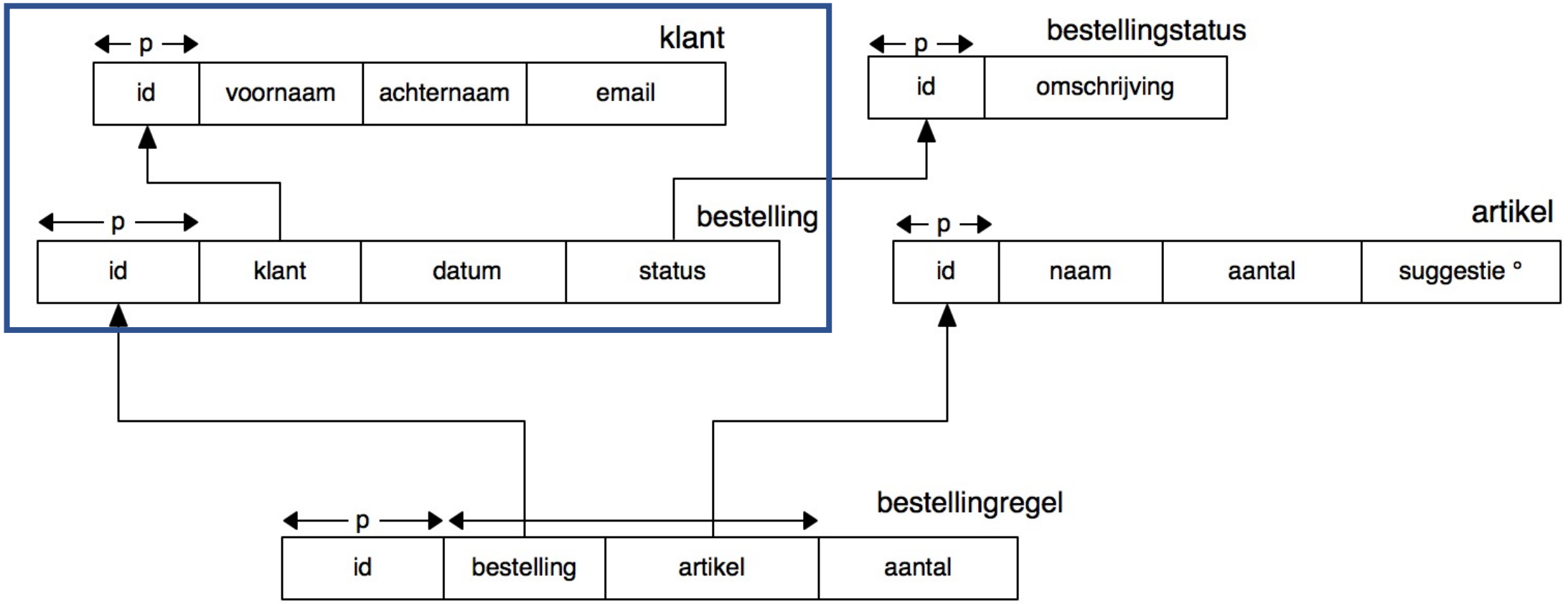
Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.



Vraag: wat komt er na ON?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

SELECT…….. FROM bestelling JOIN klant ON …….

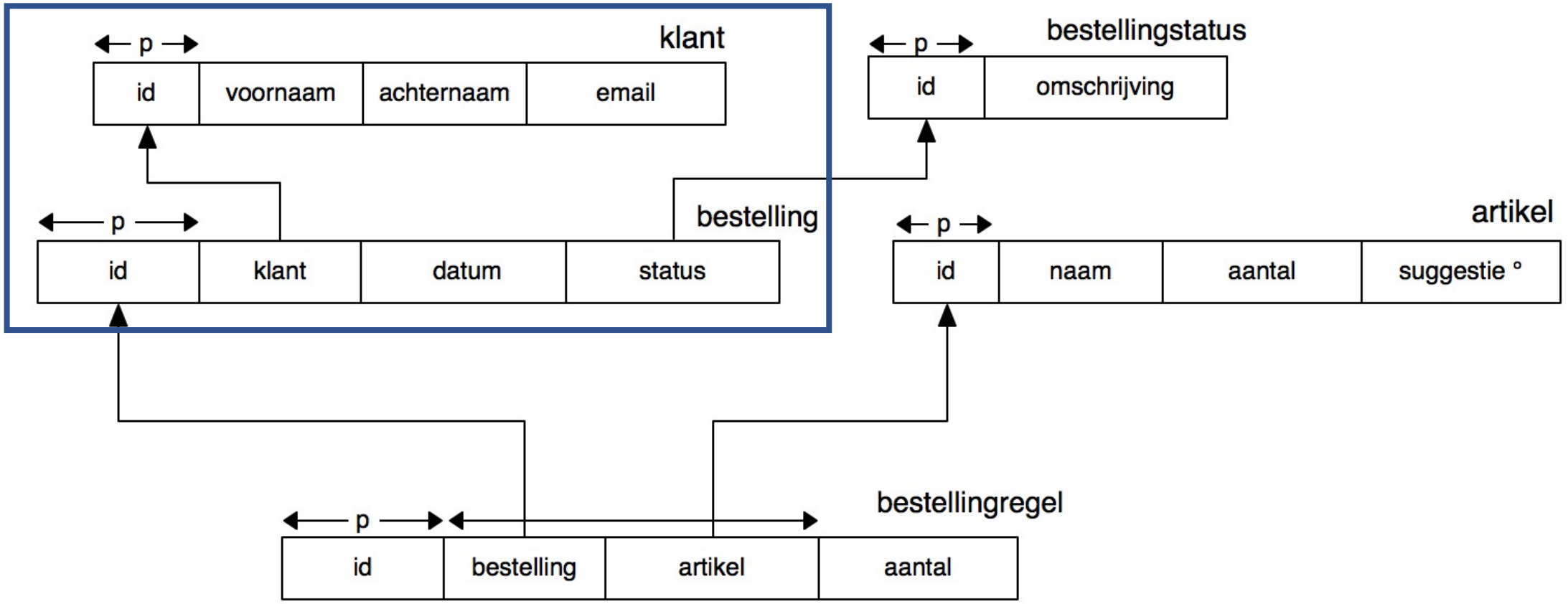


LES 4

Vraag: wat komt er na ON?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

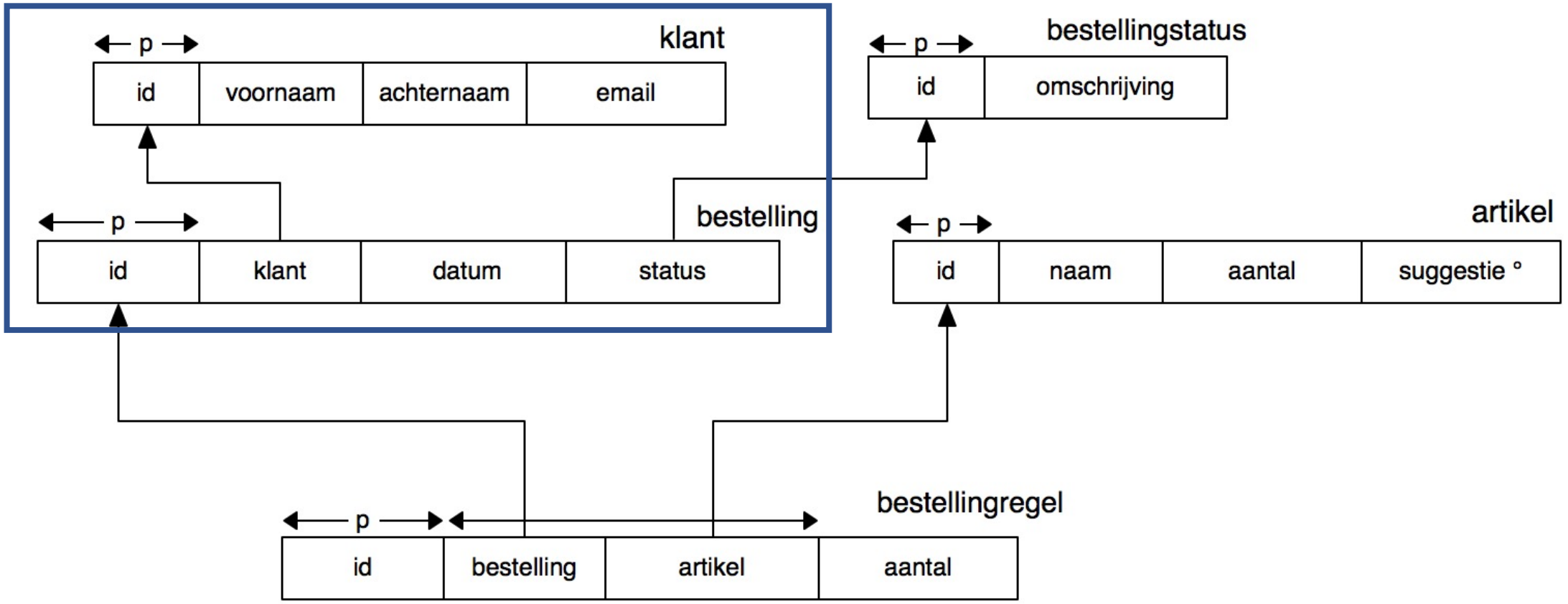
SELECT…….. FROM bestelling JOIN klant ON "alleen die regels waar bestelling.klant en klant.id gelijk zijn"



Vraag: wat komt er na ON?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

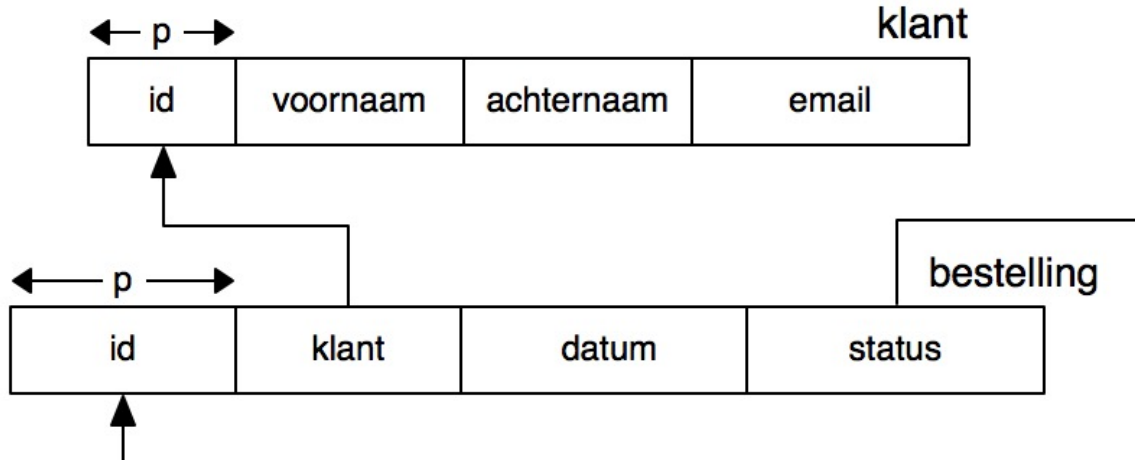
SELECT…….. FROM bestelling JOIN klant ON bestelling.klant = klant.id



Vraag: welke kolommen selecteer je?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

SELECT…….. FROM bestelling JOIN klant ON bestelling.klant = klant.id



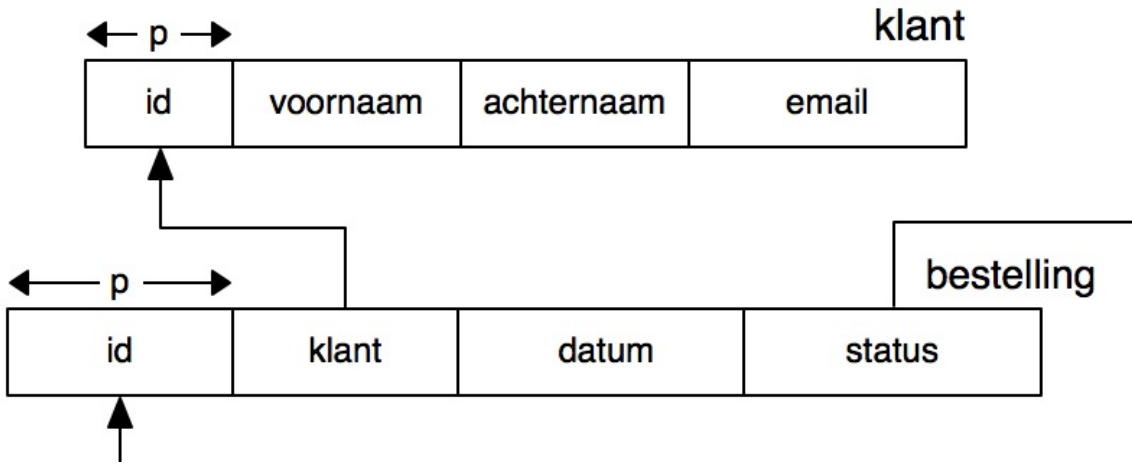
id	klant	datum	status	id	voornaam	achternaam	email
1	1	29/09/2018	4	1	Jan	Fontein	j.fontein@hotmail.com
2	5	30/09/2018	4	5	Laila	d'Hiver	laila@gmail.com
3	2	01/01/2019	4	2	Kasim	Nuberres	knuberres@gmail.com
4	1	03/03/2020	1	1	Jan	Fontein	j.fontein@hotmail.com

LES 4

Vraag: welke kolommen selecteer je?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

```
SELECT bestelling.id, voornaam, achternaam
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id
```



bestelling.id

klant.id

id	klant	datum	status	id	voornaam	achternaam	email
1	1	29/09/2018	4	1	Jan	Fontein	j.fontein@hotmail.com
2	5	30/09/2018	4	5	Laila	d'Hiver	laila@gmail.com
3	2	01/01/2019	4	2	Kasim	Nuberres	knuberres@gmail.com
4	1	03/03/2020	1	1	Jan	Fontein	j.fontein@hotmail.com

LES 4

Vraag: wat komt er na ON?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant.

```
SELECT bestelling.id, voornaam, achternaam
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id;
```

id	voornaam	achternaam
1	Jan	Fontein
2	Laila	d'Hiver
3	Kasim	Nuberres
4	Jan	Fontein

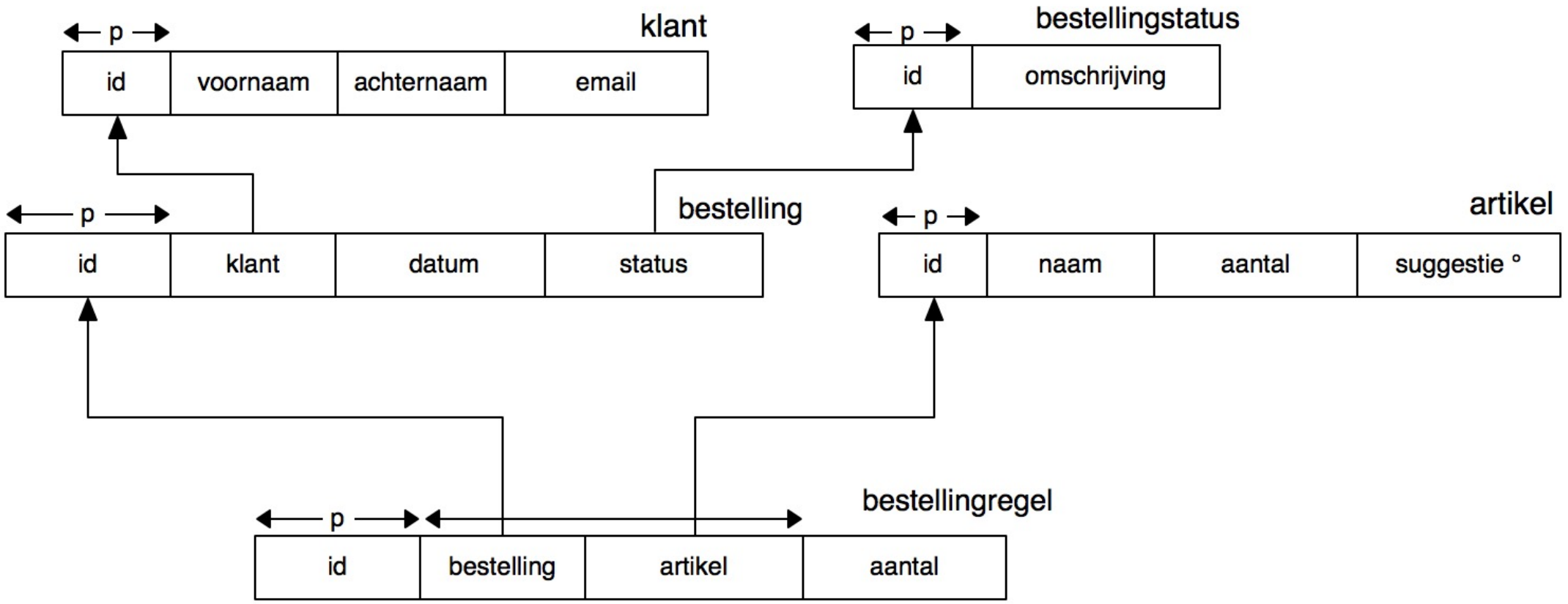
LES 4

Het gebruik van JOIN VOORBEELD met 3 tabellen

LES 4

Vraag: welke tabellen heb ik nodig?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

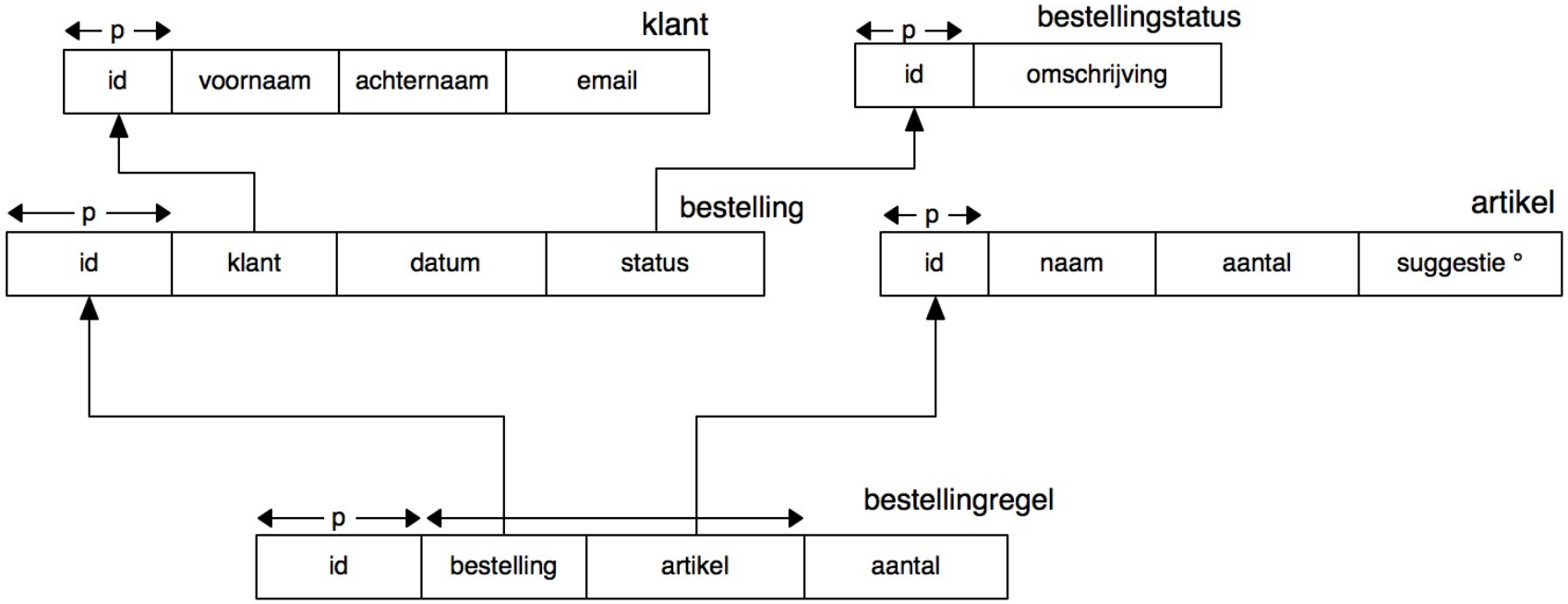


LES 4

Vraag: welke tabellen heb ik nodig?

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

SELECT …
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id
  JOIN bestellingstatus ON bestelling.status = status.id



LES 4

'Tussentabel'

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

SELECT …
FROM bestelling
JOIN klant ON bestelling.klant = klant.id
JOIN bestellingstatus ON bestelling.status = status.id

bestellingstatus

id	voornaam
1	verzonden
2	ingepakt
3	betaald
4	bezorgd

bestelling.id

klant.id

bestellingstatus.id

id	klant	datum	status
1	1	29/09/2018	4
2	5	30/09/2018	4
3	2	01/01/2019	4
4	1	03/03/2020	1

id	voornaam	achternaam	email
1	Jan	Fontein	j.fontein@hotmail.com
5	Laila	d'Hiver	laila@gmail.com
2	Kasim	Nuberres	knuberres@gmail.com
1	Jan	Fontein	j.fontein@hotmail.com

id	omschrijving
4	bezorgd
4	bezorgd
4	bezorgd
1	verzonden

'Tussentabel'

bestellingstatus

id	voornaam
1	verzonden
2	ingepakt
3	betaald
4	bezorgd

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

SELECT …
FROM bestelling
JOIN klant ON bestelling.klant = klant.id
JOIN bestellingstatus ON bestelling.status = status.id

bestelling.id

klant.id

bestellingstatus.id

id	klant	datum	status	id	voornaam	achternaam	email	id	omschrijving
1	1	29/09/2018	4	1	Jan	Fontein	j.fontein@hotmail.com	4	bezorgd
2	5	30/09/2018	4	5	Laila	d'Hiver	laila@gmail.com	4	bezorgd
3	2	01/01/2019	4	2	Kasim	Nuberres	knuberres@gmail.com	4	bezorgd
4	1	03/03/2020	1	1	Jan	Fontein	j.fontein@hotmail.com	1	verzonden

Kolommen selecteren

Alle id's van de bestellingen met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

```
SELECT bestelling.id, voornaam, achternaam, omschrijving
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id
  JOIN bestellingstatus ON bestelling.status = status.id;
```

id	voornaam	achternaam	omschrijving
1	Jan	Fontein	bezorgd
2	Laila	d'Hiver	bezorgd
3	Kasim	Nuberres	bezorgd
4	Jan	Fontein	verzonden

Maar nu…

Alle id's van de bestellingen van Jan met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

SELECT …
FROM bestelling
JOIN klant ON bestelling.klant = klant.id
JOIN bestellingstatus ON bestelling.status = status.id

id	klant	datum	status	id	voornaam	achternaam	email	id	omschrijving
1	1	29/09/2018	4	1	Jan	Fontein	j.fontein@hotmail.com	4	bezorgd
2	5	30/09/2018	4	5	Laila	d'Hiver	laila@gmail.com	4	bezorgd
3	2	01/01/2019	4	2	Kasim	Nuberres	knuberres@gmail.com	4	bezorgd
4	1	03/03/2020	1	1	Jan	Fontein	j.fontein@hotmail.com	1	verzonden

Maar nu…

Alle id's van de bestellingen van Jan met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

```
SELECT …
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id
  JOIN bestellingstatus ON bestelling.status = status.id
  WHERE voornaam = "Jan";
```

id	klant	datum	status	id	voornaam	achternaam	email	id	omschrijving
1	1	29/09/2018	4	1	Jan	Fontein	j.fontein@hotmail.com	4	bezorgd
4	1	03/03/2020	1	1	Jan	Fontein	j.fontein@hotmail.com	1	verzonden

Maar nu…

Alle id's van de bestellingen van Jan met daarbij de voor- en achternaam van de klant en de omschrijving van de bestellingstatus.

```
SELECT bestelling.id, voornaam, achternaam, omschrijving
  FROM bestelling
  JOIN klant ON bestelling.klant = klant.id
  JOIN bestellingstatus ON bestelling.status = status.id
  WHERE voornaam = "Jan";
```

id	voornaam	achternaam	omschrijving
1	Jan	Fontein	bezorgd
4	Jan	Fontein	verzonden