



Database Foundations

Fysiek datamodel

- Wat is een fysiek datamodel?
- Hoe zet ik een logisch datamodel om naar een fysiek datamodel?

Logisch model

- Vorige week:
 - logisch model
 - alleen de essentie gezien in de les
 - grote oefening was zelfstudie

- VRAAG: Lukte dat?

Doelstellingen

- Wat is een fysiek datamodel?
- Hoe zet ik een logisch datamodel om naar een fysiek datamodel?
- DBSchema leren gebruiken

Opdracht

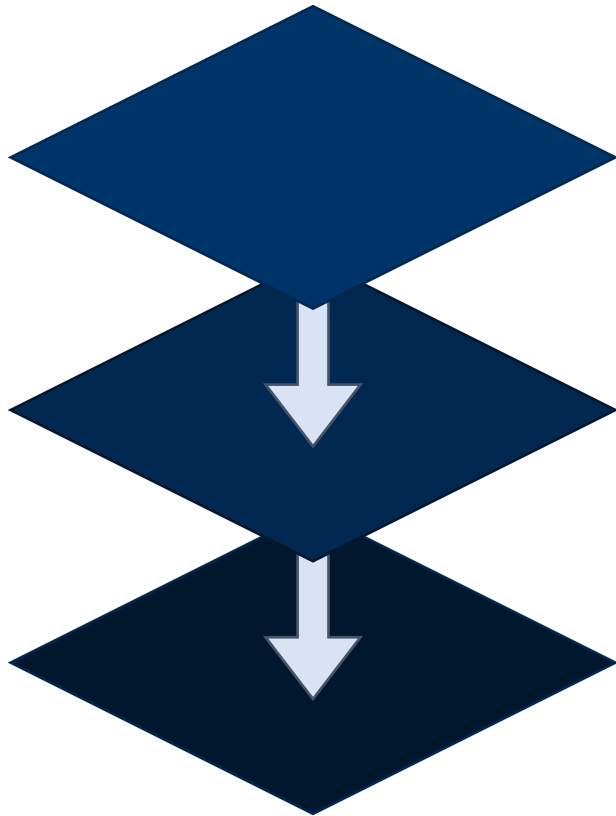
- maak nota's tijdens deze les
- vat voor jezelf samen wat je niet wil vergeten
- herwerk het nog even na de les (max. 1 A4)

- tip: het is heel praktisch, denk aan wat je nodig hebt voor het groepswerk

- volgende week laat ik mijn eigen samenvatting van deze les zien

3 lagen van een datamodel

**Een datamodel kan uit 3 lagen opgebouwd zijn...
...elke laag beantwoordt een vraag.**



Conceptueel datamodel

> Welke **informatie** moeten we in onze databank mee opnemen?

Logisch datamodel

> Hoe moeten we de data structureren volgens ons gekozen **databankmodel**?

Fysiek datamodel

> Hoe moeten we de datastructuur beschrijven volgens ons gekozen **DBMS**?

logisch + technische details

Fysiek datamodel

Technische details die in een fysiek datamodel beschreven worden:

- Het **datatype** van de kolommen
- Maximale **lengte** van een waarde
- Het al dan niet leeg (**NULL**) zijn van een waarde
- Beperkingen (**constraints**) die we aan een bepaalde tabel willen opleggen
- Indexing
- Views
- Procedures
- Gebruik van diskpace

welke datatypes kennen jullie nog?

lengtes van datatypes: voor welke?

welke datatypes kennen jullie nog?

char(x)

varchar(x)

integer / smallint

date / timestamp / time

interval (vermeld in cursus, maar ik ken het zelf niet goed)

boolean (niet in cursus denk ik)

numeric (in cursus?)

zie postgresql documentatie!

<https://www.postgresql.org/docs/current/datatype.html>

Naamgevingsconventies

Naamgevingsconventies... onze visie...

Tabel: we gebruiken een naam in **enkelvoud** die in lowercase wordt gezet. Verschillende woorden worden van elkaar gescheiden door een `\_`.

Kolom: de kolomnaam wordt in lowercase beschreven, met verschillende woorden van elkaar gescheiden door een `\_`'. Probeer de kolomnaam zo kort mogelijk te houden, maar tegelijk wel zo dat het nog mogelijk is om te achterhalen wat het betekent.

Primaire sleutel: indien mogelijk, geven we de naam van de sleutel `id`.

Vreemde sleutel: vreemde sleutels zijn verwijzingen naar andere tabellen, dus trachten we de naam van de oorspronkelijke tabel mee in de benaming op te nemen. Bijvoorbeeld de primaire sleutel `id` uit de tabel `lector` zou in andere tabellen als vreemde sleutel `lector\_id` genoemd kunnen worden.

Van een logisch naar een fysiek datamodel

Datatypes

Logisch datamodel (Draw.io)

Personeelslid	
PK	<u>Personeelsnummer</u>
	Naam
	Voornaam
	Geboortedatum



Fysiek datamodel (DBSchema)

Edit Table in Schema public

Table Na... ? personeelslid Color Global/Local

Descripti... ? ...

Columns X Primary Key / Unique / Indexes Foreign Keys Check Constraints

Visible	Name	Details	Description
☒	id	integer not null	
☒	naam	varchar(100) not null	
☒	voornaam	varchar not null	
☒	geboortedatum	date	
☐			
☐			
☐			
☐			

Edit + Add Drop Up Down Visible Columns

Help OK Cancel

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

De waarde **NULL** geeft in een databank twee mogelijke situatie aan:

- De waarde voor deze rij is **onbekend**, maar kan op een later tijdstip ingevuld worden.
- De kolom is **niet van toepassing** voor deze rij, er is geen mogelijke waarde die hier ingevuld kan worden.

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

We geven voor elke kolom die altijd ingevuld moet zijn, expliciet aan dat een kolom NOT NULL is.

- PK kolom: is **altijd** NOT NULL
- gewone kolom: verplicht ingevuld (leeg **niet zinvol**)
- FK kolom: de **kardinaliteiten**, een bepaalde verwijzing naar een andere tabel, altijd ingevuld is

NOT NULL voor PK / gewone kolommen:

Personeelslid	
PK	<u>Personeelsnummer</u>
	Naam
	Voornaam
	Geboortedatum



Edit Table in Schema public

Table Na... Global/Local

Descripti...

Columns X Primary Key / Unique / Indexes Foreign Keys Check Constraints

Visible	Name	Details	Description
☒	 id	integer not null	
☒	* naam	varchar(100) not null	
☒	* voornaam	varchar not null	
☒	= geboortedatum	date	

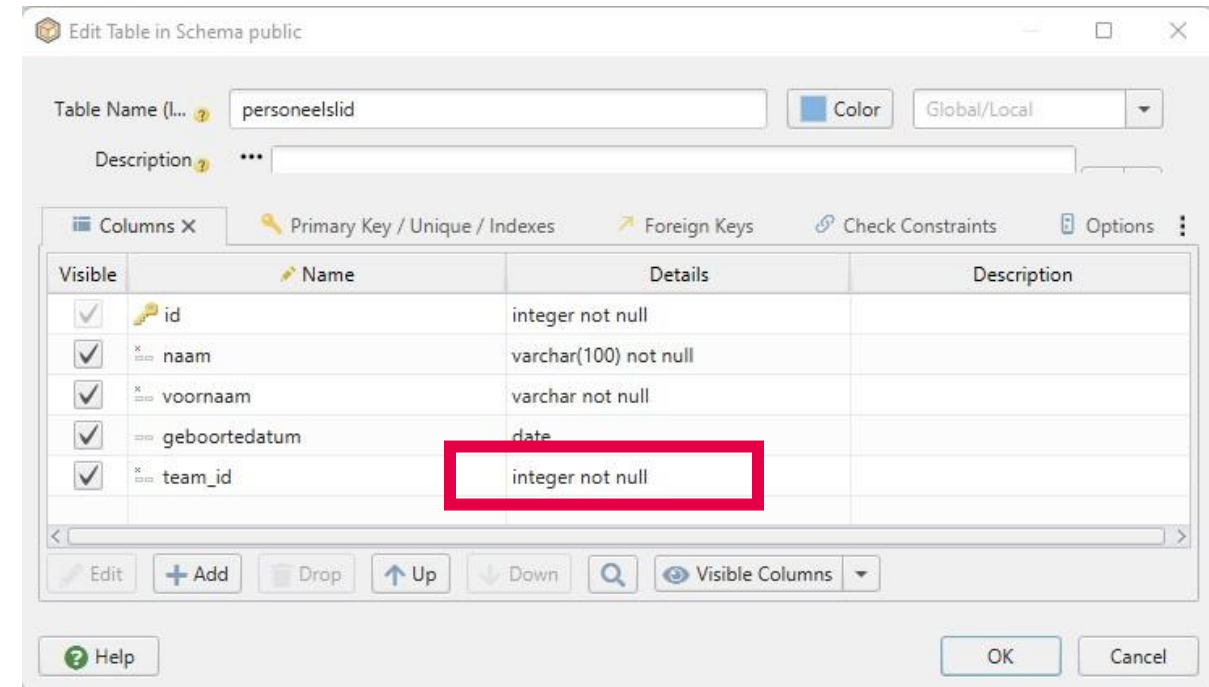
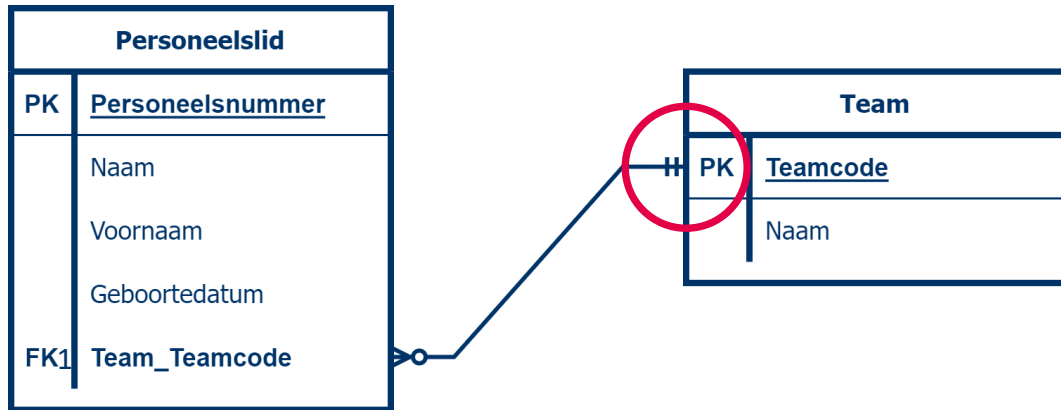
Een primaire sleutel mag nooit leeg zijn

We geven aan dat de geboortedatum van een personeelslid leeg mag zijn

Edit + Add Drop Up Down Visible Columns

Help OK Cancel

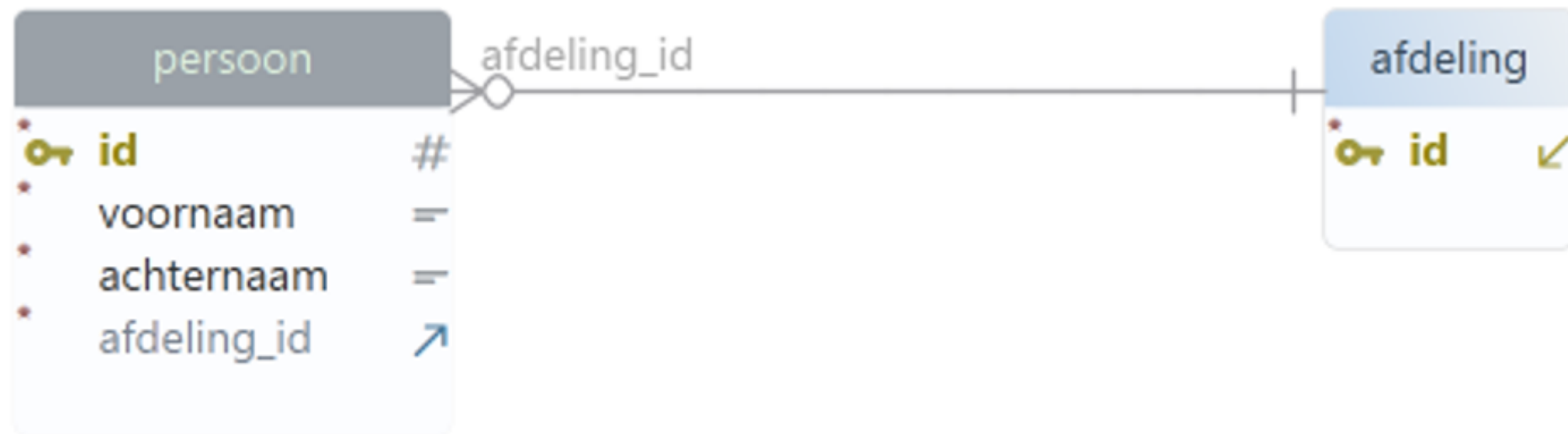
NOT NULL voor FK kolommen:



LET OP: FK NOT NULL geeft ander symbool aan PK kant

DBschema demo

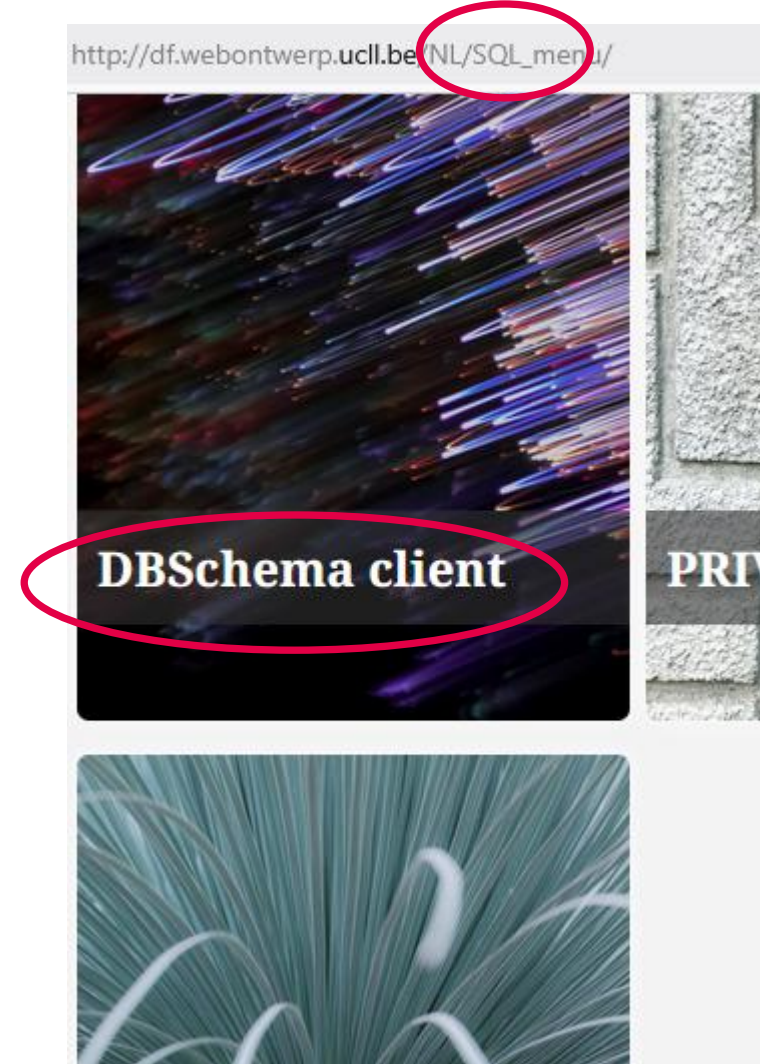
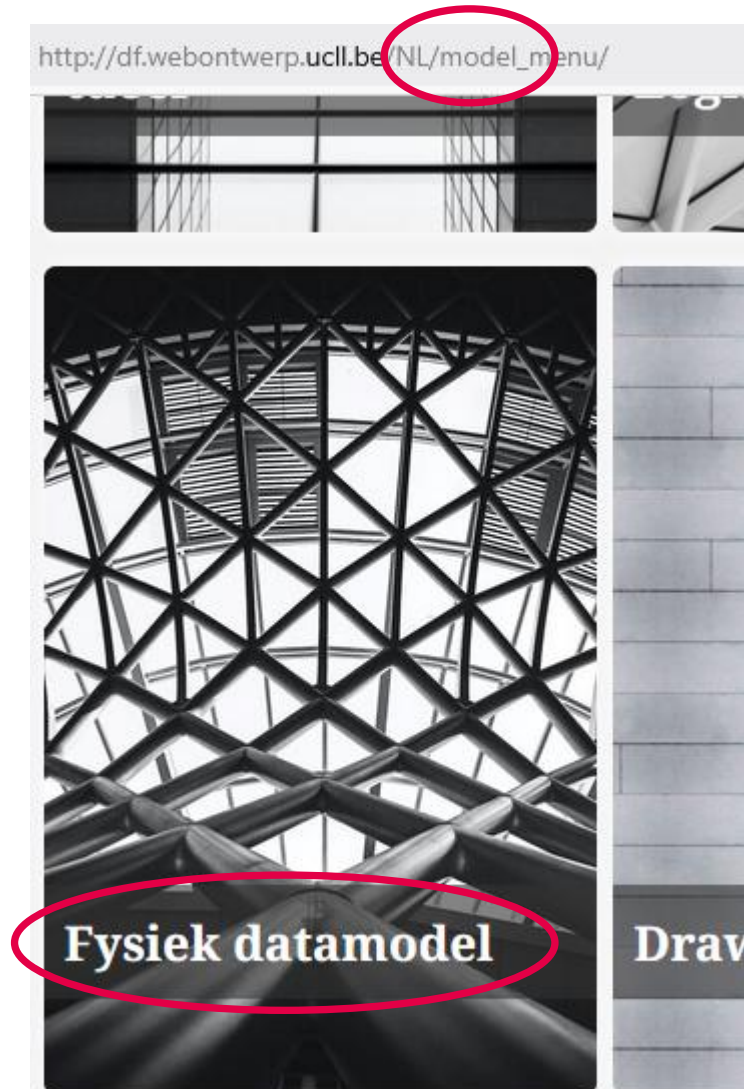
Fysiek datamodel (DBSchema)



DBschema settings

- Diagram -> foreign key notation -> "information engineering"
- Diagram -> foreign key notation -> "use strict conventions"
- Diagram -> "Foreign key line point to column"

Cursustekst: modelleren & SQL

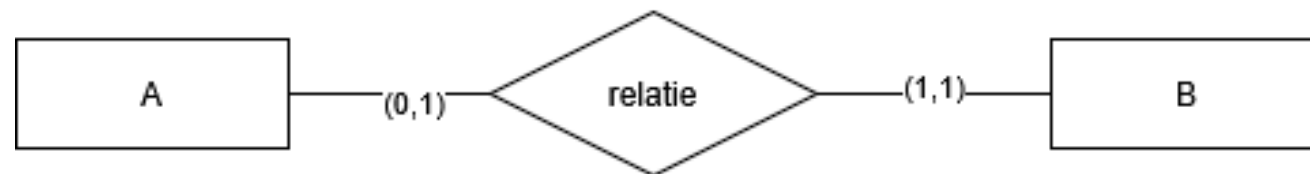
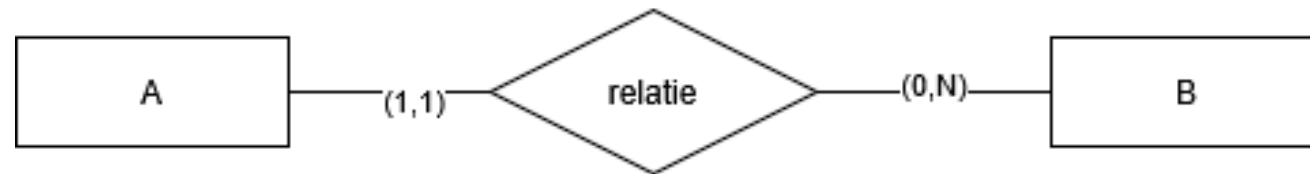
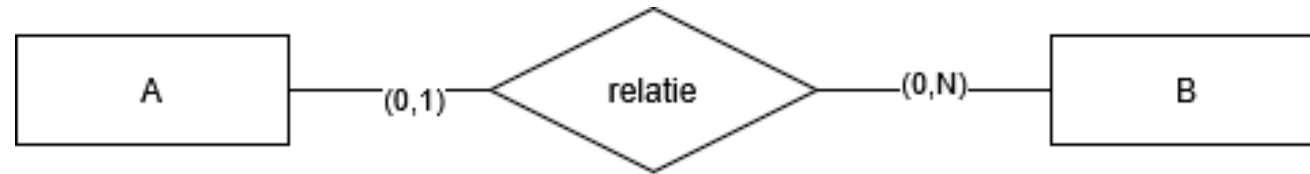


Opdracht (1)

- verzin 3 realistische conceptuele modellen
 - zie templates/kardinaliteiten volgende slide
 - kies minimum 2 attributen per ET
- voor elk: teken logisch model (mag schets op papier)
 - welke kant komt FK?
 - welk symbolen kraaienpootnotatie?
- voor elke: maak dbschema fysieke model
 - krijg ik alle kardinaliteiten goed?
 - welke welk /niet?
 - zie je verschillen in de notatie/symbolen tussen ons logisch model en DBSchema fysiek model notatie?

Opdracht (2): de kardinaliteiten

Voor opdracht van vorige slide,
gebruikt volgende 3 combinaties van kardinaliteiten:



Vragen?

