



Database Foundations

Fysiek datamodel

Deel 1

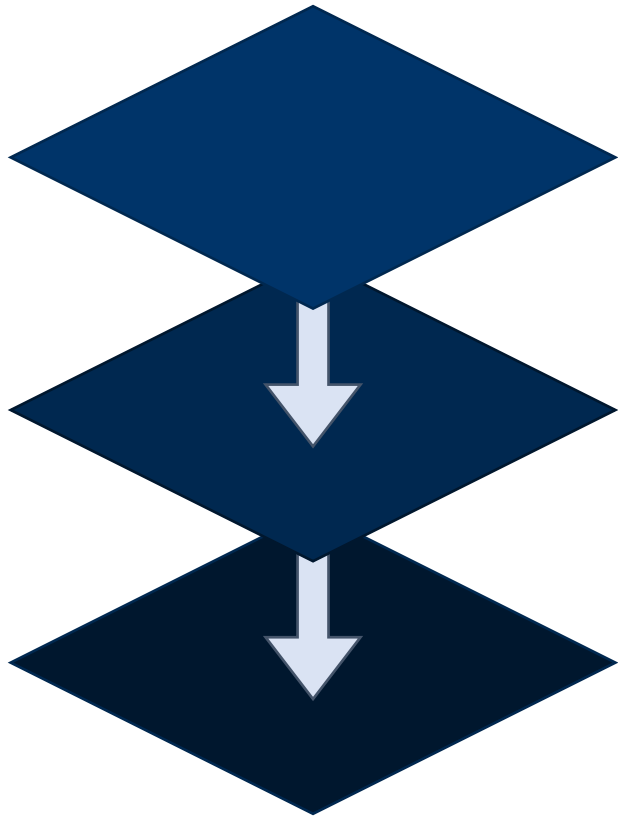
MBI10h | Academiejaar 2022 – 2023

Doelstellingen

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

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 datamodel

Kernvraag

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



Wat bevat een fysiek datamodel?

Een fysiek datamodel vult het logisch datamodel aan met **technische details**. In bepaalde gevallen wordt de structuur ook aangepast vanuit performantie en usability redenen.

Logisch 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

Logisch datamodel

Kernvraag

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



Wie? architecten, databankbeheerders, ontwikkelaars

Wanneer? Tijdens de **implementatie**-fase waar we na een keuze voor een DBMS vanuit het logisch naar een fysiek datamodel evolueren.

Hoe? Overleg architecten en DB beheerders

Van een logisch naar een fysiek datamodel

Het omzetten van een logisch naar fysiek datamodel verloopt in een aantal stappen:

- 1 Geef alle tabellen, kolommen, ... een duidelijke naam binnen de afgesproken naamgevingsconventies**
- 2 Bepaal voor elke kolom een datatype en (waar nodig) een maximale lengte**
- 3 Geef aan of de waarde in een kolommen al dan niet leeg mag zijn**

Van een logisch naar een fysiek datamodel

Het omzetten van een logisch naar fysiek datamodel verloopt in een aantal stappen:

- 1 Geef alle tabellen, kolommen, ... een duidelijke naam binnen de afgesproken naamgevingsconventies**
- 2 Bepaal voor elke kolom een datatype en (waar nodig) een maximale lengte
- 3 Geef aan of de waarde in een kolommen al dan niet leeg mag zijn

Van een logisch naar een fysiek datamodel

Naamgevingsconventies

Naamgevingsconventies zijn een **reeks van afspraken** die bepalen hoe we componenten (tabellen, kolommen, primaire sleutels, vreemde sleutels, ...) in onze databank een naam geven.

Het geheel van deze afspraken wordt **consistent** toegepast op de verschillende om de verschillende gebruikers eenvoudig doorheen de databank te laten navigeren.

Van een logisch naar een fysiek datamodel

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

Het omzetten van een logisch naar fysiek datamodel verloopt in een aantal stappen:

- 1 Geef alle tabellen, kolommen, ... een duidelijke naam binnen de afgesproken naamgevingsconventies
- 2 **Bepaal voor elke kolom een datatype en (waar nodig) een maximale lengte**
- 3 Geef aan of de waarde in een kolommen al dan niet leeg mag zijn

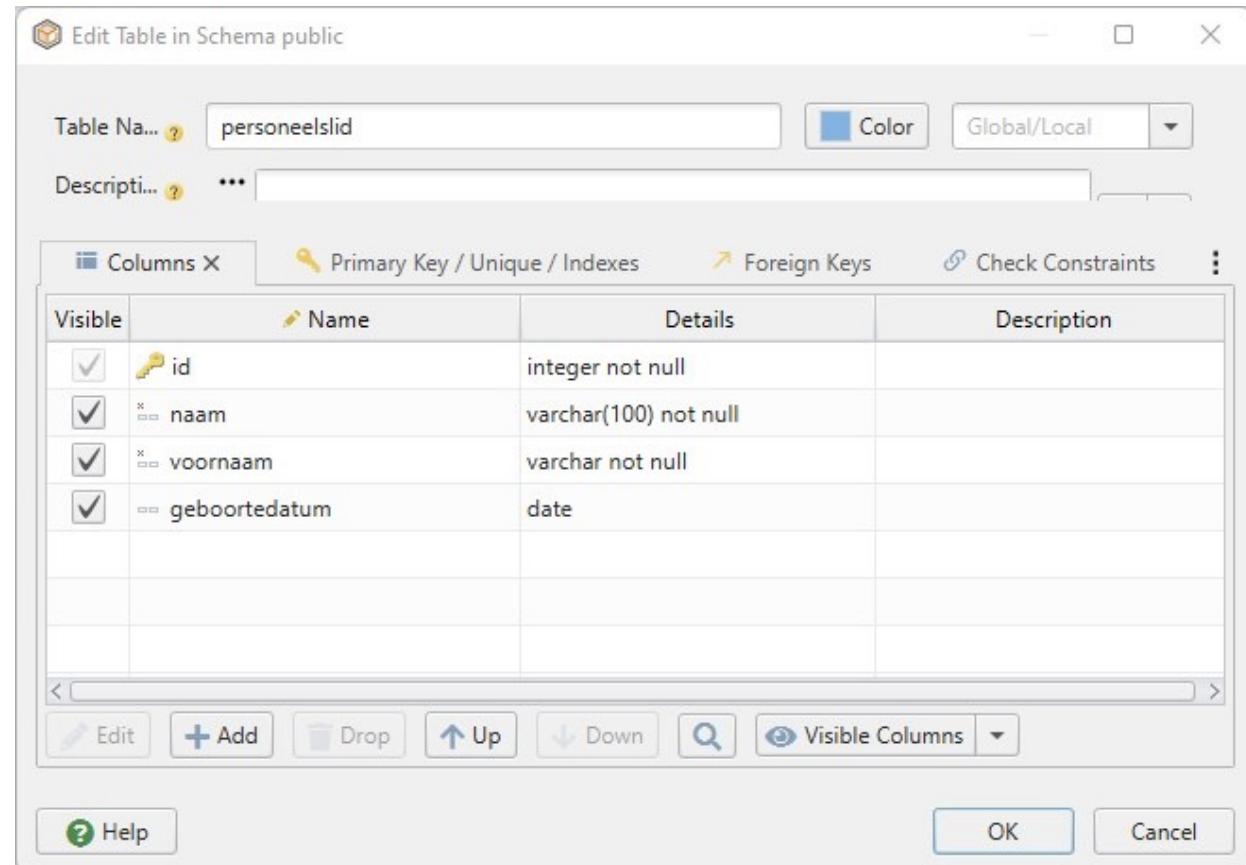
Van een logisch naar een fysiek datamodel

Datatypes

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)



A screenshot of the 'Edit Table in Schema public' dialog box. The table name is 'personeelslid'. The description is empty. The 'Columns' tab is selected, showing a table with columns: Visible, Name, Details, and Description. The columns are: id (Primary Key, integer not null), naam (varchar(100) not null), voornaam (varchar not null), and geboortedatum (date). The 'Visible' column has checkboxes for each attribute, all of which are checked. The 'Details' column shows the data type and constraints. The 'Description' column is empty. The dialog box has buttons for Edit, Add, Drop, Up, Down, Search, and Visible Columns. There are also OK and Cancel buttons at the bottom right.

Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	

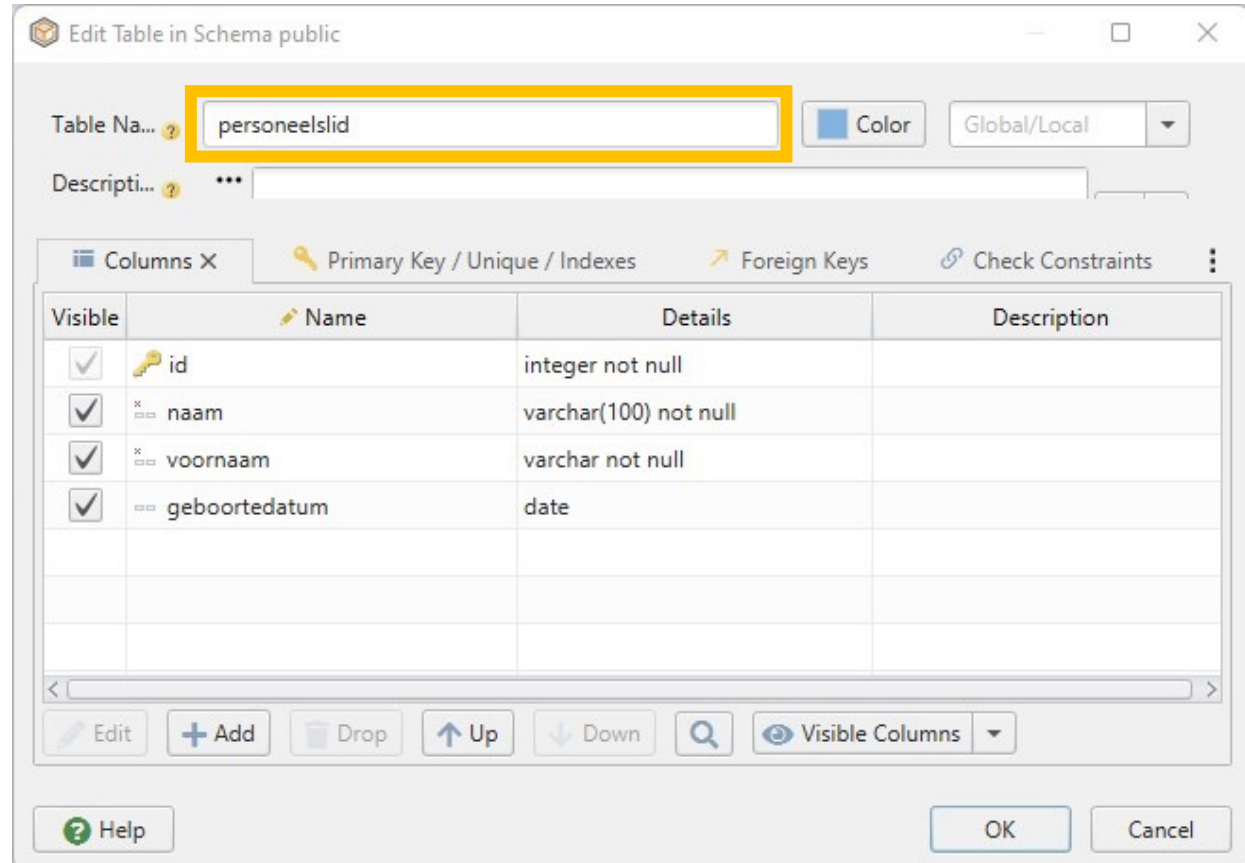
Van een logisch naar een fysiek datamodel

Datatypes

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)



A screenshot of the 'Edit Table in Schema public' window in the DBSchema application. The window title is 'Edit Table in Schema public'. The 'Table Na...' field contains 'personeelslid'. The 'Descripti...' field is empty. Below the fields are tabs for 'Columns X', 'Primary Key / Unique / Indexes', 'Foreign Keys', and 'Check Constraints'. The 'Columns X' tab is active, showing a table with columns: 'Visible', 'Name', 'Details', and 'Description'. The table contains four rows of data:

Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	

At the bottom of the window are buttons for 'Edit', '+ Add', 'Drop', 'Up', 'Down', 'Visible Columns', 'Help', 'OK', and 'Cancel'.

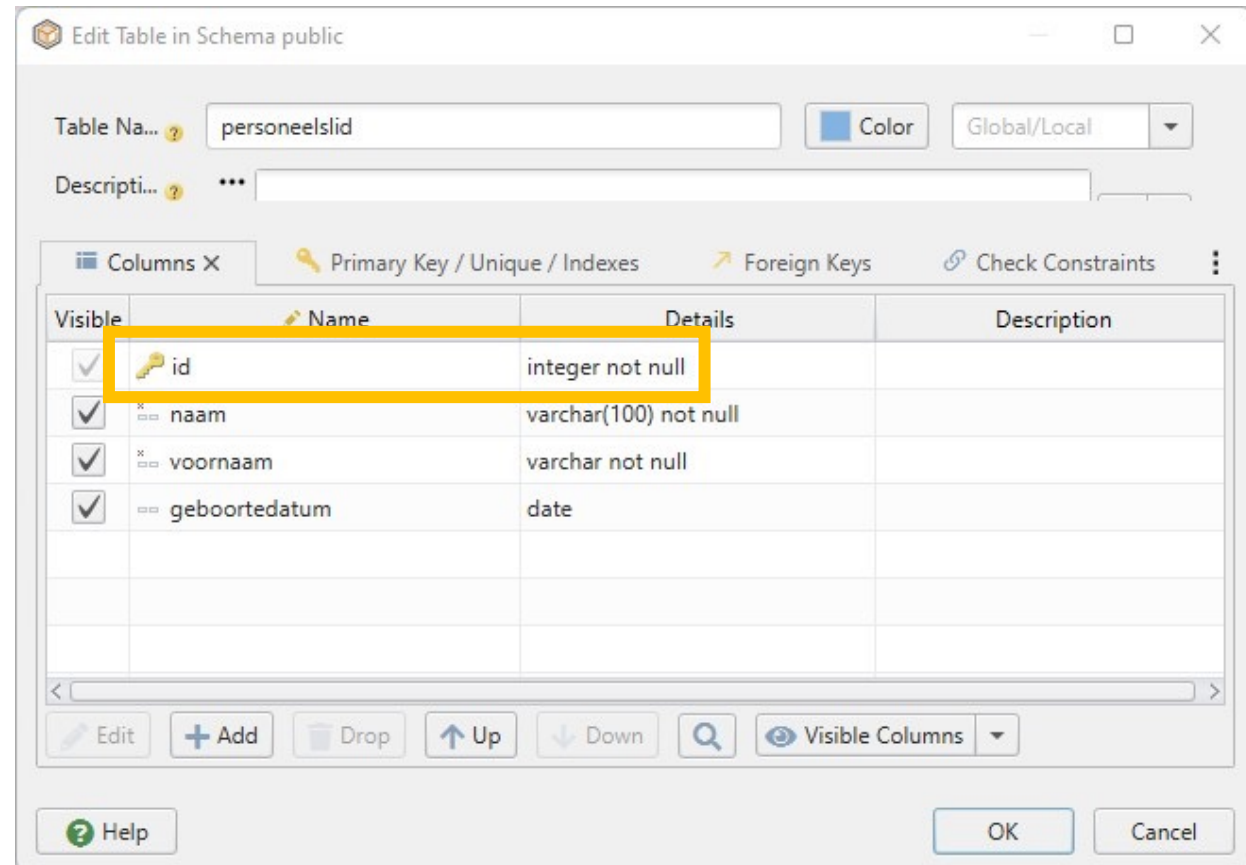
Van een logisch naar een fysiek datamodel

Datatypes

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)



The screenshot shows the 'Edit Table in Schema public' window in the DBSchema application. The table name is 'personeelslid'. The 'Columns' tab is selected, showing a list of columns with their data types and constraints. The 'id' column is highlighted with a yellow box, showing it is an 'integer not null' and is the primary key. Other columns include 'naam' (varchar(100) not null), 'voornaam' (varchar not null), and 'geboortedatum' (date).

Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	

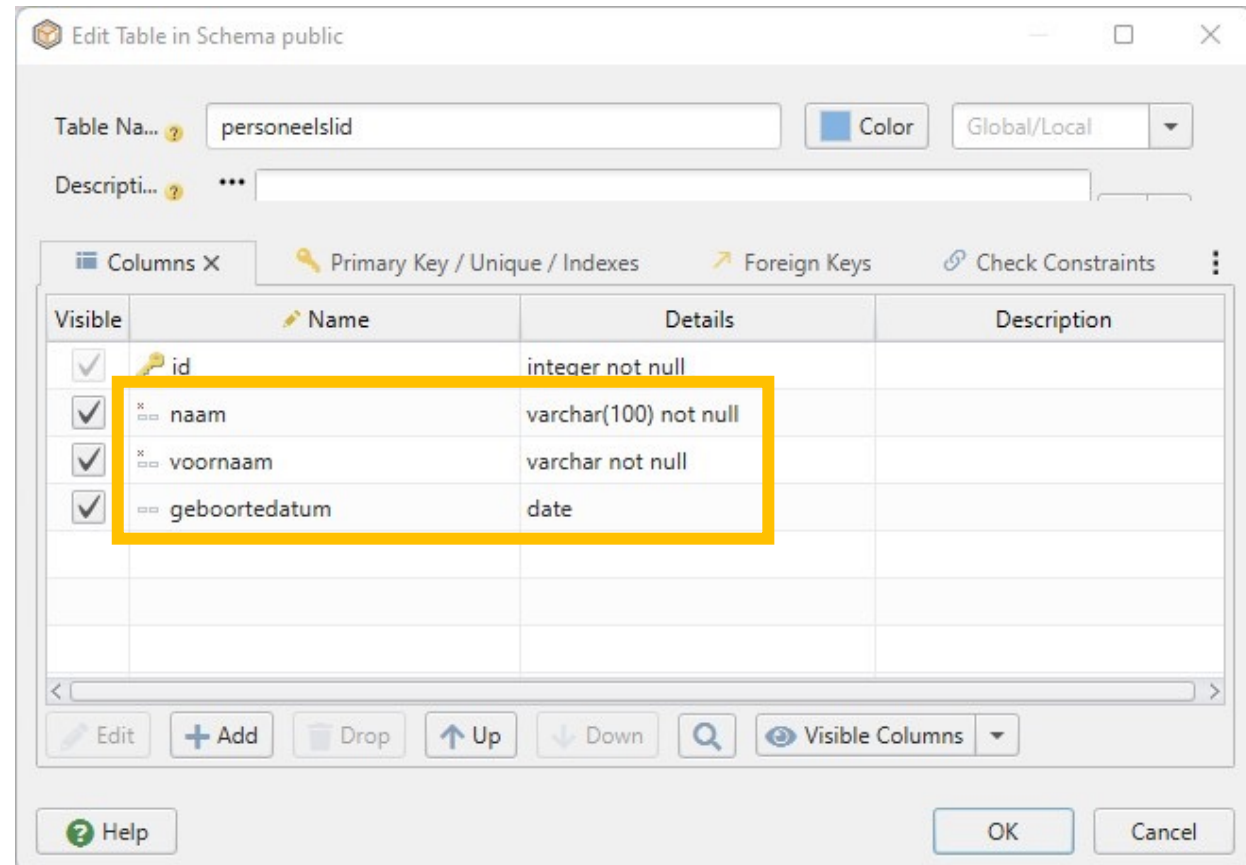
Van een logisch naar een fysiek datamodel

Datatypes

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)



Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	

Van een logisch naar een fysiek datamodel

Het omzetten van een logisch naar fysiek datamodel verloopt in een aantal stappen:

- 1 Geef alle tabellen, kolommen, ... een duidelijke naam binnen de afgesproken naamgevingsconventies
- 2 Bepaal voor elke kolom een datatype en (waar nodig) een maximale lengte
- 3 **Geef aan of de waarde in een kolommen al dan niet leeg mag zijn**

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.

We willen op deze manier 2 situaties afdekken:

- De klant wilt dat een bepaalde kolom altijd ingevuld wordt, omdat een ontbrekende waarde **niet zinvol** is.
- Omwille van de relaties en de **kardinaliteiten** die hierbij verbonden zijn, willen we zeker zijn dat een bepaalde verwijzing naar een andere tabel, altijd ingevuld is.

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

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
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	

Edit
Add
Drop
Up
Down
Visible Columns

Help
OK
Cancel

Een primaire sleutel mag nooit leeg zijn

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

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

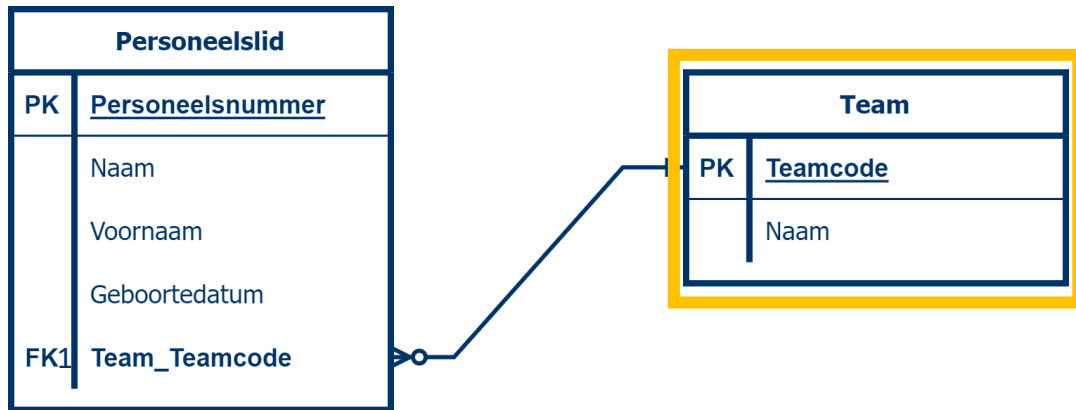
De waarde **NULL** laat ook toe om de **minimumkardinaliteit** van een relatie af te dwingen.

Indien een rij in een tabel **ALTIJD** een verwijzing moet bevatten naar een andere tabel (door middel van een vreemde sleutel), kunnen we voor die sleutel aangeven dat die nooit NULL mag zijn...

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)

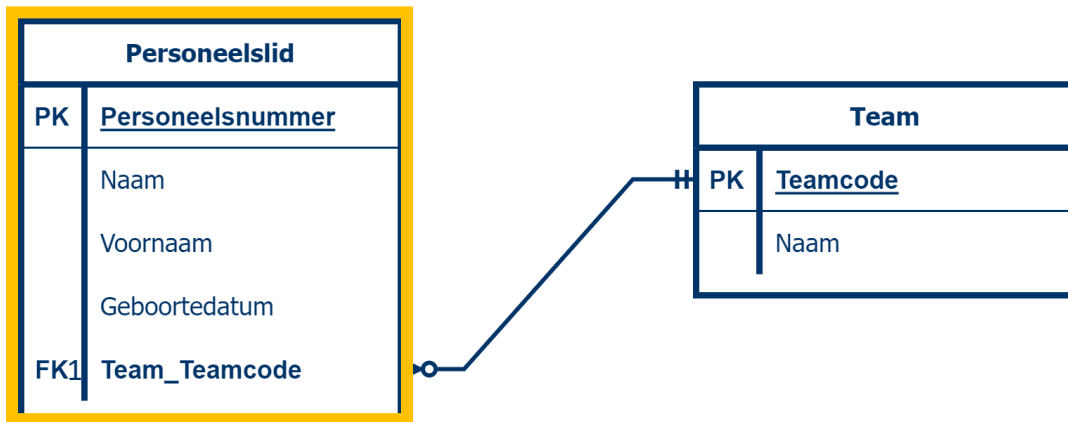
The screenshot shows the 'Edit Table in Schema public' window for a table named 'team'. The 'Columns' tab is selected, showing a table with columns 'id' and 'team_naam'. The 'id' column is marked as the primary key (integer not null) and 'team_naam' is marked as not null (varchar(100) not null). Both columns are checked in the 'Visible' column.

Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	team_naam	varchar(100) not null	

Van een logisch naar een fysiek datamodel

NULL of NOT NULL

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)

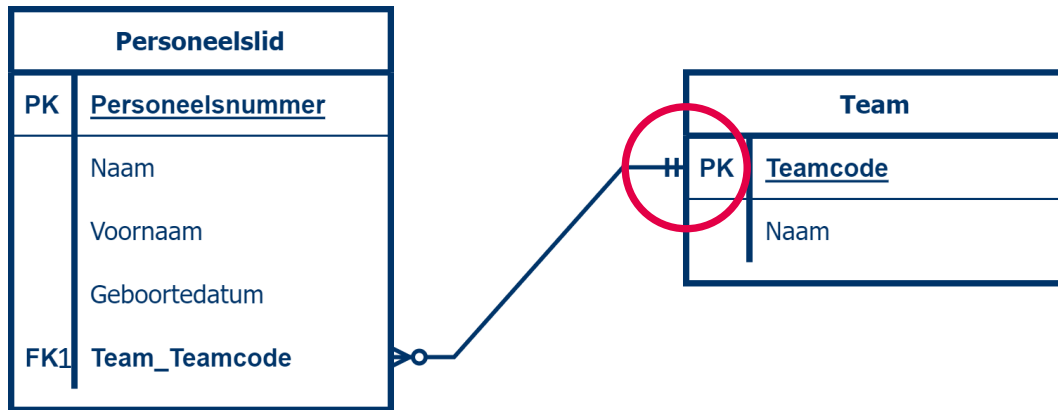
The screenshot shows the 'Edit Table in Schema public' window for the table 'personeelslid'. The table name and description are at the top. Below are tabs for 'Columns', 'Primary Key / Unique / Indexes', 'Foreign Keys', 'Check Constraints', and 'Options'. The 'Columns' tab is active, showing a table with columns: 'id' (integer not null, primary key), 'naam' (varchar(100) not null), 'voornaam' (varchar not null), 'geboortedatum' (date), and 'team_id' (integer not null). A yellow box highlights the column list.

Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	
<input checked="" type="checkbox"/>	team_id	integer not null	

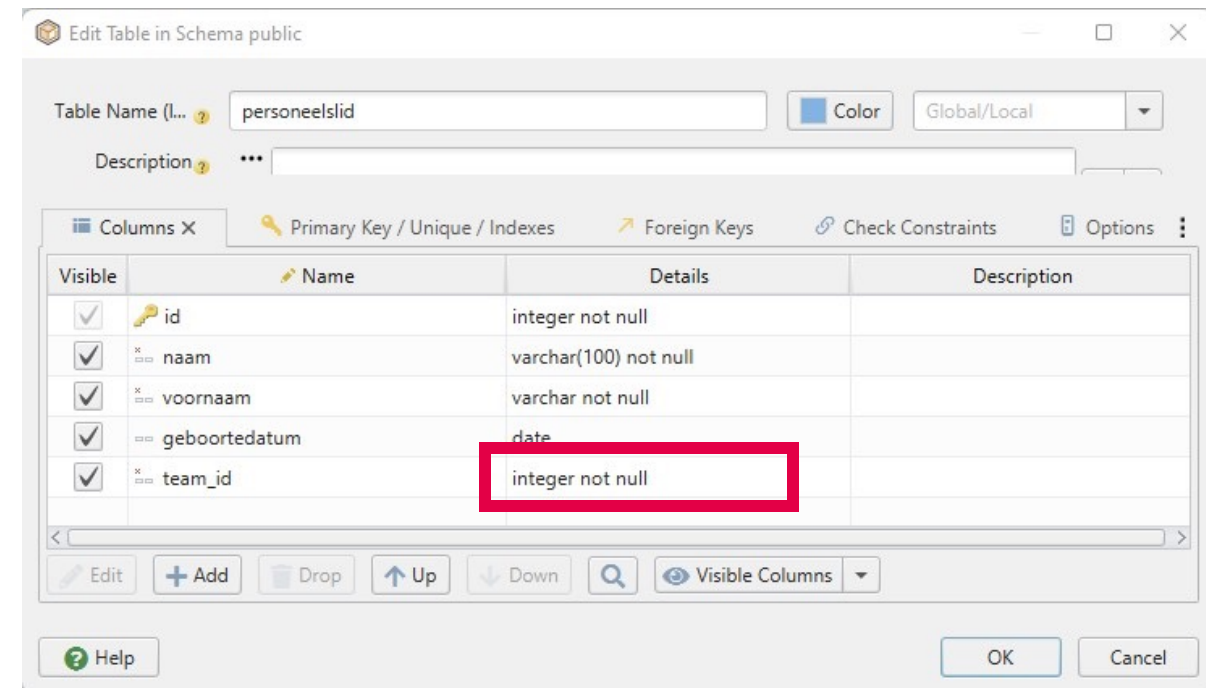
Van een logisch naar een fysiek datamodel

NULL of NOT NULL

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)

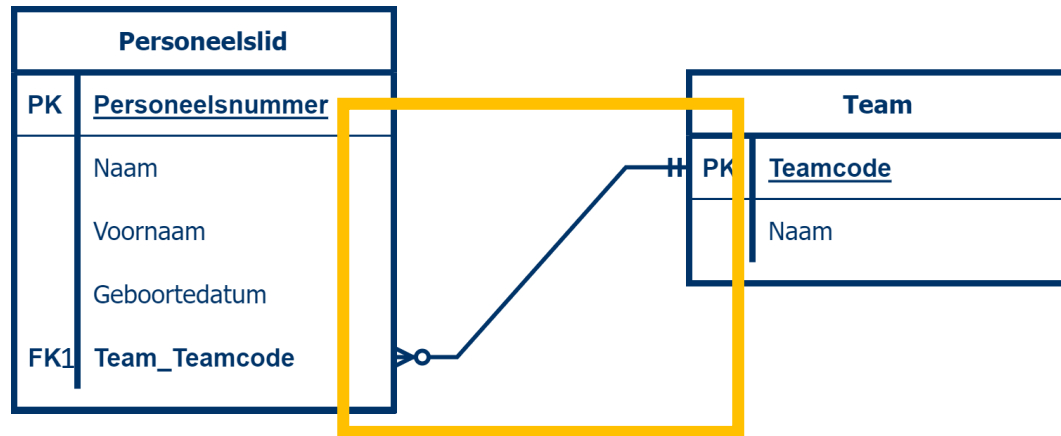


Visible	Name	Details	Description
<input checked="" type="checkbox"/>	id	integer not null	
<input checked="" type="checkbox"/>	naam	varchar(100) not null	
<input checked="" type="checkbox"/>	voornaam	varchar not null	
<input checked="" type="checkbox"/>	geboortedatum	date	
<input checked="" type="checkbox"/>	team_id	integer not null	

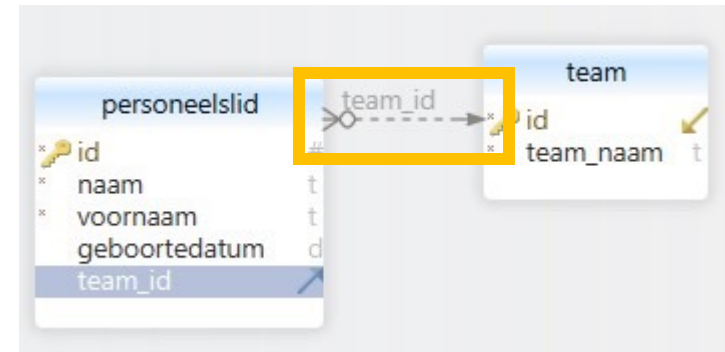
Van een logisch naar een fysiek datamodel

NULL of NOT NULL

Logisch datamodel (Draw.io)



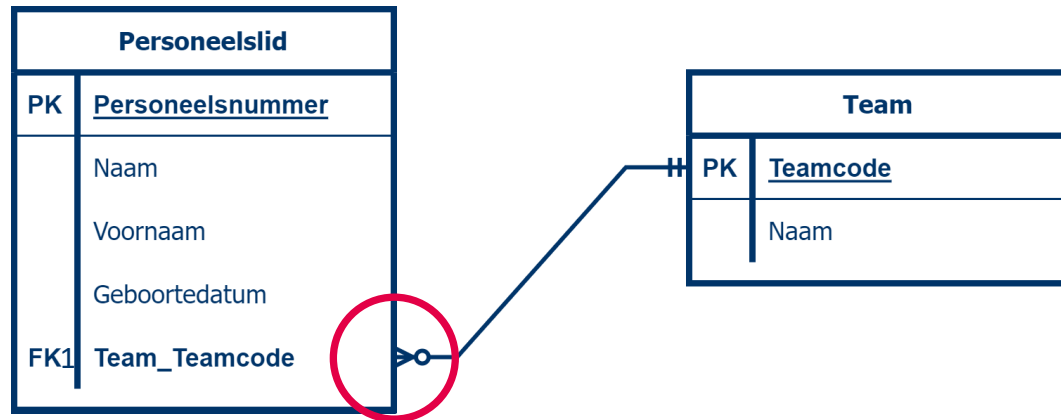
Fysiek datamodel (DBSchema)



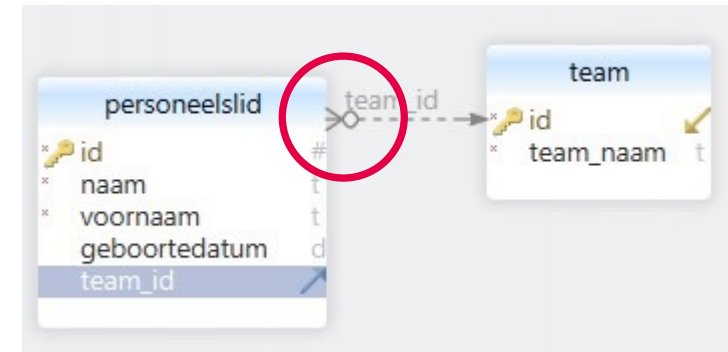
Van een logisch naar een fysiek datamodel

NULL of NOT NULL

Logisch datamodel (Draw.io)



Fysiek datamodel (DBSchema)



Vragen?



**Gebruik de contactmomenten om
eventuele vragen te stellen**