Samenvatting Database Design

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.
   - N:M kun je alleen via een koppeltabel maken.

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.

Video Workshop

De volgende video is een samenvatting van de Workshop Databases (1)

https://www.youtube.com/embed/MT-gW0Uo_IQ

https://youtu.be/MT-gW0Uo_IQ

Datatypes

De meest voorkomende datatypes zijn:

Datatype	Voorbeeld
int	-2 147 648 - 2 147 649
varchar(), bijv. varchar(20)	"Big Boss 12"
date	2022-04-01
datetime	2022-04-01 18:43:12
decimal(6,2)	1250,95

Relaties (1:N) tussen entiteiten

Een relatie tussen twee entiteiten is vrijwel altijd een 1-op-meer relatie en heeft daardoor aan één kant een 'harkje'.

> image-1643803461861.png
> ('harkje')

Voorbeelden

- één persoon bezit meer (één of meer) auto's en niet anders om een auto heeft geen twee eigenaren
- één klas bestaat uit meer studenten en niet een student zit in meer klassen
- één huiswerkopdracht is meerdere keren beantwoord (door één student, denk aan Canvas) en niet één antwoord hoort bij meerdere huiswerkopdrachten.
- één voetbalclub heeft meer spelers en niet één speler hoort bij meerdere voetbalclub
- één school heeft meerdere studenten en niet één student zit op meerdere scholen

Het gaat bij al deze zaken niet of het echt niet kan. Natuurlijk zou een student op meerdere scholen kunnen zitten, maar dat is niet de regel. Zou dit wel 'normaal' zijn, dan zouden we de relatie ook als meer-meer kunnen zien. Eén school heeft meer studenten en één student zit op meerdere scholen. Een meer-op-meer relatie wordt in DB Design Level 2 besproken.

Neem student en studiecoach. Welke regel denk je dat van toepassing is?

één studiecoach	meer-studenten
eén student	meer studiecoaches

OK in dit geval is het dus de eerste regel. Dat betekent dat de meer kant het 'harkje' krijgt. De lijn tussen de twee entiteiten in het ERD heeft dus het harkje aan de meer-kant. Bij het harkje hoort ook de FK.

	één - kant	meer - kant
PK	één unieke PK	één unieke PK
Lijntje	geen 'harkje' (alleen streep)	'harkje'
FK	geen FK	één FK verwijst naar de PK van de één kant

Het ERD wordt dan.

image-1643791334313.png

Beide entiteiten hebben een unieke PK en FK en harkje staan aan dezelfde kant.

Samengevat: het harkje staat aan de meer kant en bij elk harkje hoort een FK. Of nog korter:

> **HARKJE = MEER = FK**

N:M (veel-op-veel)

Stel je voor je hebt de entiteit, *product* en *klant*. De relatie tussen deze twee entiteiten is N:M, veel-op-veel. Eén klant kan immers meer producten kopen en een product kan door meerdere klanten worden gekocht.

Om dit in een database te zetten moet je een koppeltabel aanmaken. Dit is een extra entiteit. Dat ziet er zo uit:

image-1664052926462.png

(voor het gemak zijn in dit ERD de datatypen even weggelaten).

De entiteit product_klant is de koppeltabel en deze verbind het product en de klant aan elkaar zodat er een N:M relatie ontstaat.

Via de combinatie van de FK's worden producten aan klanten gekoppeld. Stel er is een klant met het id 101 en een klant met id 102 en stel je hebt een product met id 10 en 11. Stel klant 101 en klant 102 hebben allebei product 10 en 11 gekocht. Dan staat er in de koppel tabel de volgende informatie:

id (PK)	klant_id (FK)	product_id (FK)
1	101	10
2	102	11

3	101	10
4	102	11

Stappenplan maken ERD

Nog even alle stappen die je moet uitvoeren om een ERD te maken op een rijtje:

1. Bepaal alle entiteiten. Dit zijn personen, dingen of gebeurtenissen waar je gegevens over wilt vastleggen.

2. Bepaal van alle entiteiten de attributen (wat je vastleggen).

3. Bepaal de datatypes van alle attributen.

4. Zorg ervoor dat elke entiteit een PK krijgt (bij twijfel gebruik je 'id').

5. Bepaal de relaties tussen de entiteiten. Bij een N:M relatie heb je een koppeltabel nodig.

6. Teken de relaties. Het harkje staat aan de meer-kant.

7. Bij elk harkje hoort een FK, de FK verwijst naar de PK met de entiteit waarmee de relatie bestaat.

8. Lees het verhaal nog een keer door en controleer of je alles wat er in het verhaal staat kunt vastleggen in je databaseontwerp.

Checklist, wat gaat er vaak fout?

1. Heeft elke entiteit precies één PK (*Primar yKey*)?

2. Is elke PK uniek, dus kan er maar één van voorkomen?
   (Achternaam kan bijvoorbeeld *geen* PK zijn, omdat er meer mensen zijn met dezelfde achternaam).
   Tip: meestal zijn PK's int en vaak heten ze gewoon id.

3. Heeft elke attribuut een datatype?

4. Telefoonnummer is geen int want dan valt de eerst 0 weg, immers 0612341234 wordt 612341234

5. Datum is altijd datatype date.

6. Datum plus tijd is datatype datetime.

7. De relatie heeft maximaal één 'harkje'. Het harkje staat aan de 'meer' kant. Dus een *student* 'hoort' bij één *klas* en bij een *klas* 'horen' meerdere *studenten*. Het harkje staat in dit voorbeeld dan aan de student kant.

8. Bij elk 'harkje' hoort precies één FK. De FK verwijst naar de PK van de table waarmee deze is verbonden.

9. De PK en FK die bij elkaar horen hebben hetzelfde datatype.

10. int heeft een vaste lengte, het is dus int en niet int(11).

11. varchar heeft altijd een lengte, dit is de *maximaal* lengte die kan voorkomen. varchar(5) als plaatsnaam is dus fout.

12. String datatypes zijn er in verschillende vormen in Database-land. De meest gebruikte zijn: char en varchar; char heeft een vaste lengte, varchar heeft een maximale lengte. *String* bestaat niet in MySQL.

Powerpoint

Twee powerpoints:

1. Database I
2. Database II

--

Perma-link: https://www.roc.ovh/link/459

Revision #29
Created 2022-01-19 17:46:50 UTC by Max
Updated 2023-10-05 17:30:54 UTC by Max