**Extra oefeningen BIS deel 2 / ter voorbereiding op de toets**

**Versie:** 22-11-2018

Totaal 4 delen, gelijk aan de echte toets, qua puntenaantal zijn deze in de echte toets als volgt verdeeld:

Vraag 1 (NPI en conceptueel model) -> 20 punten

Vraag 2 (ER model) -> 20 punten

Vraag 3 (Normalisatie) -> 30 punten

Vraag 4 (DDL) -> 20 punten

*Deel 1: NPI*

Het bedrijf FietsTotaal met vestigingen over heel Nederland wil een centrale administratie aanleggen. Hierin kunnen ze bijhouden welke klant welke order heeft geplaatst. Omdat klanten over heel Nederland verspreid zijn, legt Fietstotaal per klant een uniek klantnummer vast. Het bedrijf heet FietsTotaal, maar er is sprake van branchevervaging: er worden allerlei producten geleverd, van verschillende merken en types. Een order bestaat uit meerdere producten, en voor Fietstotaal is van belang dat orderdatum en aantal per product wordt vastgelegd. Een order kan in 1 keer worden betaald, maar het kan ook zijn dat meerdere betalingen worden uitgevoerd voor 1 order.

Vraag 1a: voer de noun phrase identification uit op bovenstaande tekst. Benoem concepten, attributen en relaties, en wanneer je er voor kiest iets niet op te nemen, geef dan een valide reden op.

Uitwerking

Het bedrijf FietsTotaal met vestigingen over heel Nederland wil een centrale administratie aanleggen. Hierin kunnen ze bijhouden welke klant welke order heeft geplaatst (kopers kunnen meerdere orders plaatsen, en kunnen zich al registreren voordat ze een order geplaatst hebben). Omdat klanten over heel Nederland verspreid zijn, legt Fietstotaal per klant een uniek klantnummer vast. Het bedrijf heet FietsTotaal, maar er is sprake van branchevervaging: er worden allerlei producten geleverd, van verschillende merken en types, met ieder een eigen productnummer. Een order bestaat uit één of meerdere producten, en voor Fietstotaal is van belang dat ordernummer en aantal per product wordt vastgelegd. Een order kan in 1 keer worden betaald, maar het kan ook zijn dat meerdere betalingen worden gedaan voor 1 order. Daarbij wordt in ieder geval betaaldatum en betaalwijze vastgelegd, en een uniek id.

|  |  |  |  |
| --- | --- | --- | --- |
| **Zelfstandig naamwoorden (long list)** | **Valt af wegens** | **Kandidaat concepten** | **Kandidaat attributen** |
| Bedrijf | Scope |  |  |
| Vestigingen | Scope |  |  |
| Nederland | Scope |  |  |
| Administratie | Containerbegrip |  |  |
| Klant |  | Klant |  |
| Order |  | Order |  |
| Kopers | Dubbel |  |  |
| Producten |  | Product |  |
| Klantnummer |  |  | Klantnummer (Klant) |
| Merk |  |  | Merk (product) |
| Type |  |  | Type (product) |
| Productnummer |  |  | Productnummer(product) |
| Ordernummer |  |  | Orderdatum (order) |
| Aantal |  |  | Aantal (order) |
| Betaling |  | Betaling |  |
| Betaaldatum |  |  | Betaaldatum(betaling) |
| Betaalwijze |  |  | Betaalwijze(betaling) |
| Id |  |  | Id(Betaling) |
| Branchevervaging | Abstract |  |  |

Relaties:

Klant plaatst Order

Order bestaat uit Producten

Betaling wordt gedaan voor Order

Toelichting: let op enkelvoud / zelfstandige naamwoord bij concept en attribuut. Bij op zoek gaan naar relaties kun je als in de tekst te weinig houvast is ook andere werkwoorden gebruiken, maar als het in de tekst staat, neem die dan over.

Een uitwerking met vestiging als concept is ook mogelijk, en evt een relatie naar product (leveren).

Vraag 1b:

Maak op basis van je NPI het conceptuele model.

Uitwerking



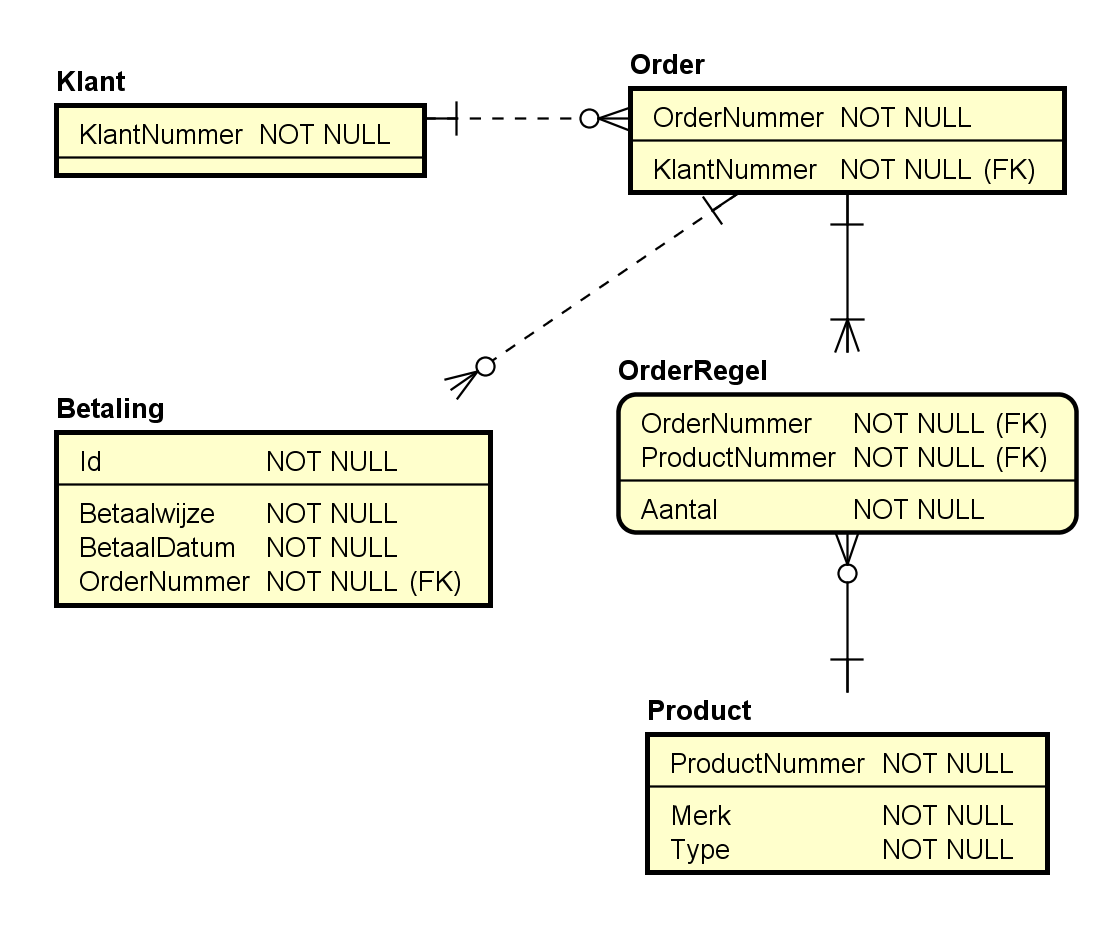
Zorg voor consistentie tussen npi en model. En eventueel attributen opnemen, maar dat hoeft niet.

*Deel 2: ER model*

(in de toets zal een aparte opgave worden opgenomen voor het ER model, in deze extra oefeningen werk je verder met de gegevens uit deel 1)

Vraag 2a: Werk op basis van de informatie uit deel 1 ook het ER model uit. Gebruik de Martin notatiewijze, en neem de attributen op.

Uitwerking:



Belangrijk bij deze uitwerking is dat je onderkent dat aantal iets zegt over de relatie tussen product en order. En dat je daarom de veel op veel relatie al in het logische model uitwerkt met een koppeltabel met een zwakke entiteit Orderregel.

Qua notatie: van belang om verplicht – optioneel per veld vast te leggen, in deze oefenopgave heb je wat weinig informatie gekregen, dus daarom zijn alle attributen als verplicht aangemerkt.

Voor deze uitwerking is Astah gebruikt, deze tool geeft bij een relatie direct het foreign key attribuut in de child-tabel weer, dit hoeft niet (is eigenlijk mooier als je het weglaat, het is niet nodig, de relatie tussen de entiteiten tekenen is voldoende).

*Deel 3 - Normalisatie*

Onderstaand overzicht geeft de administratie van een dierenarts weer, die bijhoudt welke behandelingen hij verricht bij welk dier. Het kan voorkomen dat de dierenarts een bepaalde behandeling (procedure) vaker uitvoert bij een dier.

DierNr Naam Soort Leeftijd Eigenaar Datum Bezoek Procedure Behandelaar

246 ROVER HOND 12 SAM COOK JAN 13/2002 01 - RABIES VACCINATIE JANSEN

246 ROVER HOND 12 SAM COOK MAR 27/2002 10 – ONDERZOEK WOND JANSEN

246 ROVER HOND 12 SAM COOK APR 02/2002 05 – ONTWORMEN PIETERS

298 SPOT HOND 3 TERRY KIM JAN 21/2002 08 - TETANUS VACCINATIE PIETERS

298 SPOT HOND 3 TERRY KIM MAR 10/2002 05 – ONTWORMEN PIETERS

341 MORRIS KAT 4 SAM COOK JAN 23/2001 01 - RABIES VACCINATIE PIETERS

341 MORRIS KAT 4 SAM COOK JAN 13/2002 01 - RABIES VACCINATIE KLAASSEN

519 TWEEDY VOGEL 2 TERRY KIM APR 30/2002 20 – JAARLIJKSE CONTROLE KLAASSEN

519 TWEEDY VOGEL 2 TERRY KIM APR 30/2002 12 – OOG SPOELING JANSEN

Vraag 3a: wat is de meest geschikte primary key van deze gegevensverzameling?

Uitwerking: DierNr, Datum, Procedure. We kiezen voor diernr en niet voor naam, als er een nummer beschikbaar is, is dat veld meer geschikt als sleutel, grotere kans op uniciteit.

Als de vraag was geweest: wat zijn mogelijke candidate keys dan had je ook kunnen komen met de combinatie leeftijd, datum, procedure. Of met diernr, datum, behandelaar. Of datum, procedure, behandelaar.

Verder is er eigenlijk nog een ‘technisch’ betere sleutel, namelijk Datum + behandelaar, dat is een sleutel over maar 2 kolommen, en een smallere sleutel heeft voordelen. Tegelijk: wil je hier dingen gaan vastleggen over behandelaars en wat ze allemaal doen, of over dieren en hun behandelingen? Deze sleutel levert een andere uitwerking van tabellen op, die we hier verder niet opnemen, maar die bij consequent uitwerken op een toets zeker goed gerekend zouden worden.

Vraag 3b: in welke normaalvorm staat deze tabel en waarom niet in een hogere normaalvorm?

Uitwerking: in 0NV, het veld procedure bevat geen atomair feit, maar de combinatie van een nummer en een naam. (je hoeft vervolgens niet uit te leggen dat het ook geen 2NV of 3NV is).

Vraag 3c: wat zijn de afhankelijkheden in deze tabel?

DierNr, Datum, Procedure -> Behandelaar

DierNr -> Naam, Soort, Leeftijd, Eigenaar

ProcedureNr -> ProcedureNaam

(let op: als je constateert dat een tabel geen 1NV is maar in 1 veld meerdere waarden bevat, dan moet je die meenemen bij de afhankelijkheden)

Vraag 3d: zet de stappen om deze tabel in 3NV te zetten. Neem alle tussenstappen die volgens jou nodig zijn op. In dit voorbeeld hoef je de ***populatie niet mee te nemen***, geef alleen de kolomnamen aan, en zorg dat iedere tabel die je opneemt voldoet aan de eisen van 1NV.

Uitwerking: (waarbij onderstreepte attributen de primary key van een tabel vormen)

**1NV:**

**Bezoek [ Diernr, DatumBezoek, ProcedureNr, Naam, soort, leeftijd, eigenaar, ProcedureNaam, Behandelaar ]**

**2NV:**

**Dier [ Diernr, Naam, soort, leeftijd, eigenaar ]**

**Bezoek [ Diernr, DatumBezoek, ProcedureNr, Behandelaar ]**

**Procedure [ ProcedureNr, ProcedureNaam]**

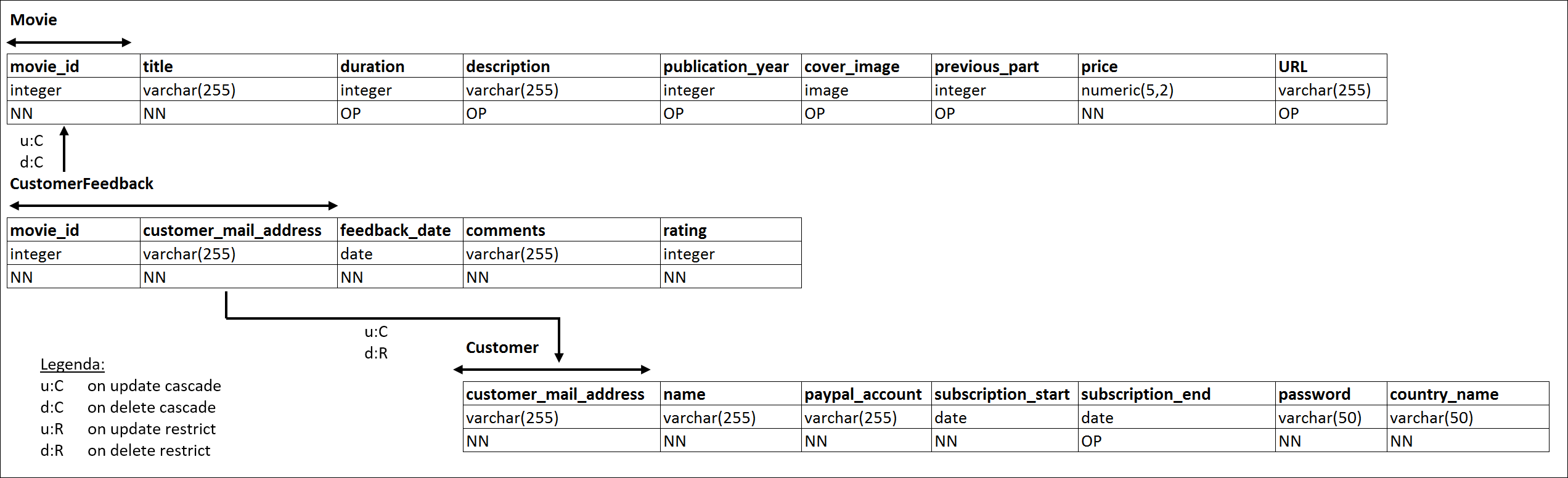
**3NV:**

**Gelijk aan 2NV**

**Bij de uitwerking:** de formulering ‘voldoet aan de eisen van 1NV’ betekent dus dat een tabel een naam heeft en een primary key!

*Deel 4: DDL*

De eigenaar van filmsite wil graag te weten komen wat klanten van bekeken films vinden. Hiervoor is het volgende datamodel bedacht:



In de komende opgave ga je de SQL statements opnemen om deze tabellen te realiseren. Bedenk steeds dat je een uitvoerbare (dus complete) instructie geeft.

(in de echte toets zullen de tabellen minder kolommen tellen zodat het schrijfwerk beperkt is, maar voor deze oefening zijn expres meer kolommen opgenomen)

Vraag 4a: Geef het SQL statement dat de tabel CustomerFeedback aanmaakt zonder foreign key constraints.

Uitwerking:

CREATE TABLE CustomerFeedback(

movie\_Id Integer Not Null,

customer\_mail\_address Varchar(255) Not Null,

feedback\_date Date Not Null,

comments Varchar(255) Not Null,

rating Integer Not Null,

CONSTRAINT PK\_CustomerFeedback PRIMARY KEY (movie\_id,customer\_mail\_address)

)

CREATE TABLE Customer(

Customer\_mail\_address Varchar(255) Not Null,

Name Varchar(255) Not Null,

Paypal\_account Varchar(255) Null,

Subscription\_start Date Not Null,

Subscription\_end Date Null,

Password Varchar(50) Not Null,

Url Varchar(50) Not Null,

CONSTRAINT PK\_Customer PRIMARY KEY (Customer\_mail\_address)

)

CREATE TABLE Movie(

movie\_Id Integer Not Null,

title Varchar(255) Not Null,

duration Integer Null,

Description Varchar(255) Null,

Publication\_year Integer Null,

Cover\_image Image Null,

Previous\_part Integer Null,

Price Numeric(5,2) Not Null,

Url Varchar(255) Null,

CONSTRAINT PK\_Movie PRIMARY KEY (movie\_id)

)

Opmerkingen bij de uitwerking: let op datatypes, mandatory, tabelnamen, consequent zijn in je primary keys conform de opgave / betekenisvolle naam voor constraints.

De tabellen Customer en Movie zijn ook aangemaakt, omdat deze bij de volgende opgave nodig zijn (anders kun je geen foreign key definiëren)

Vraag 4b: Geef het SQL statement dat achteraf (wanneer de tabellen al bestaan) de twee foreign keys inclusief de cascading rules implementeert. Zorg dat de foreign keys een betekenisvolle naam krijgen.

Uitwerking:

ALTER TABLE CustomerFeedback

ADD CONSTRAINT FK1\_Customer FOREIGN KEY(customer\_mail\_address)

REFERENCES Customer (customer\_mail\_address)

ON UPDATE CASCADE

ON DELETE NO ACTION,

CONSTRAINT FK2\_Movie FOREIGN KEY(movie\_id)

REFERENCES Movie (movie\_id)

ON UPDATE CASCADE

ON DELETE CASCADE

Of

alter table Customerfeedback

add constraint fk\_customerfeedback\_customer foreign key (customer\_mail\_address)

references customer (customer\_mail\_address)

ON UPDATE CASCADE ON DELETE NO ACTION

alter table Customerfeedback

add constraint fk\_customerfeedback\_movie foreign key (movie\_id)

references movie (movie\_id)

ON UPDATE CASCADE ON DELETE CASCADE

Tav de uitwerking: zorg dat je eerst goed kijkt voor welke tabel je de foreign key moet vastleggen / lees het schema goed / kijk naar parent & child! En let op de naamgeving van constraints, en op je syntax -> ook als het gaat om komma’s en haakjes…

Vraag 4c: Gegeven de volgende beperkingsregel:

“De kolom rating mag slechts de waarden 1, 2, 3, 4 of 5 bevatten.”

Geef het SQL statement dat deze constraint achteraf (wanneer de tabel al bestaat) implementeert. Zorg dat de constraint een betekenisvolle naam krijgt.

Uitwerking:

ALTER TABLE CustomerFeedback

ADD CONSTRAINT CK\_Rating

CHECK (rating IN (1,2,3,4,5))

of

ALTER TABLE CustomerFeedback

ADD CONSTRAINT CK2\_Rating

CHECK (rating BETWEEN 1 AND 5)

of

ALTER TABLE CustomerFeedback

ADD CONSTRAINT CK2\_Rating

CHECK (rating >= 1 AND rating <= 5)

Vraag 4d:

Gegeven de volgende beperkingsregel:

“Een customer mag maximaal 1 keer per dag feedback geven.”

Geef het SQL statement dat deze constraint achteraf (wanneer de tabel al bestaat) implementeert. Zorg dat de constraint een betekenisvolle naam krijgen.

Uitwerking:

ALTER TABLE CustomerFeedback

ADD CONSTRAINT AK\_Feedback\_date UNIQUE (customer\_mail\_address, feedback\_date)

Wat ook kan: de primary key van de table droppen, en daarna een nieuwe primary key leggen op de combinatie customer\_mail\_address en feedback\_date.