

Opdracht Gegevensbanken - deel 3: DB creatie en front-end

Tussentijds verslag

TEAMLEDEN

1. Pieter Van den Berghe
2. Niels Verhagen
3. Vincent Vliegen

SESSIE

1. Welke moeilijkheden ondervonden jullie tijdens deel 1 van de opdracht (d.w.z., de modellering van het domein in ER)?

Er waren meerdere mogelijkheden om het domein te modelleren in het entiteit-relatiemodel, afhankelijk van verschillende redeneringen en andere interpretaties van de opgave. De moeilijkheid hier was niet het vinden van een oplossing, maar het vinden van de meest optimale. Een voorbeeld hiervan was het opsplitsen van een ternaire relatie in verschillende binaire relaties.

2. Hoe verschilde jullie oplossing van de modeloplossing? Welke bijkomende veronderstellingen maakten jullie?

Onze oplossing verschilt op een aantal vlakken van de modeloplossing, dit valt deels te wijten aan een vorm van 'interpretatie' van de opgave.

In de ingestuurde oplossing wordt onder meer gebruik gemaakt van een quaternaire relatie 'transported over shipping route', deze relatie stelt op recursieve wijze de route tussen twee havens voor. In de modeloplossing gebruikt men een relatie 'route' om de route tussen twee verschillende havens voor te stellen.

Verder vinden we in de modeloplossing een afgeleide relatie 'trip' terug, dit staat niet vermeld in onze oplossing.

De door ons gemaakte aannames hadden hoofdzakelijk als doel de cardinaliteiten te verklaren, een voorbeeld hiervan is dat er minstens één container op een schip aanwezig moest zijn.

3. Welke moeilijkheden ondervonden jullie tijdens deel 2 van de opdracht (d.w.z., de omzetting van het ER-diagram naar een relationeel model)? Wat was de reden voor deze moeilijkheden?

Voor deel twee van de opdracht maakten we gebruik van het stappenplan dat in de les aan bod is gekomen. Omdat het ER-model relatief groot was t.o.v. de behandelde oefeningen in de oefenzitting was het aanvankelijk moeilijk om het overzicht te houden. Door het stappenplan nauwgezet te volgen slaagden we er toch in om het overzichtelijk te houden en de opdracht op papier uit te werken.

De volgende moeilijkheid was het relationeel model digitaal voorstellen. Om dit te doen gebruikten we excel, dit had als voordeel dat het erg overzichtelijk was. Het grote nadeel was het grafisch voorstellen van pijlen, dit was erg tijdrovend, met andere software was het hoogstwaarschijnlijk makkelijker geweest. Bij de omzetting van de excel file naar pdf verplaatsten de pijlen zich en kwamen ze op een foute plaats terecht, dit verliep dus ook alles behalve probleemloos en vroeg extra tijd.

4. Welke moeilijkheden ondervonden jullie tijdens deel 3 van de opdracht (d.w.z., het ontwikkelen van een web-based front-end voor de database in PHP/HTML)?

Dit derde deel van het werkje was voor ons veruit het moeilijkste. Niemand van onze groep had eerder ervaring gehad met PHP, hierdoor was het zeer moeilijk om aan de opdracht te beginnen en iets te laten werken. Omdat onze oefenzitting pas op donderdag doorging (en we toen pas vragen konden stellen) werd het tijdsgebrek een bijkomende moeilijkheid.

Om de eerste oefeningen te implementeren was het mogelijk om voorbeelden te bekijken en hierop verder te werken, deze aanpak werkte snel niet meer en een meer diepgaande kennis van PHP was vereist. Het uitdenken van de queries vormde in het algemeen zeker niet de moeilijkheid maar de vertaling naar PHP bleek het grote struikelblok. De voorkennis die we gekregen hadden in de hoorcolleges was niet voldoende, bijvoorbeeld een oefenzitting die focust op de implementatie in PHP had zeker veel geholpen.

5. Wat zijn de sterke punten van jullie design (het ER-model) en de implementatie?

Het design van het ER-model is relatief overzichtelijk gebracht en is beperkt in grootte. Dit vormt een voordeel om de structuur goed te begrijpen en hoewel we de modeloplossing gebruikt hebben voor de omzetting naar een relationeel model was het toch een goede leidraad.

Bij de implementatie hebben we geprobeerd om de gegeven structuur te behouden. We hebben verder gewerkt met de gegeven bestanden en ernaar geijverd om code in dezelfde 'stijl' te schrijven. Een voorbeeld hiervan is de code om shipments en orders te printen. De structuur van deze code is dezelfde als die van customers, onderling verschillen deze stukken code enkel op vlak van welke gegevens ze opvragen.

6. Hoe zou je het *design* en/of de *implementatie* kunnen verbeteren, vanuit het oogpunt van de database? Welke technieken uit de hoorcolleges zouden hierbij kunnen helpen?

Het design van de database kan verbeterd worden door de database te normaliseren. Hierdoor zullen errors bij verandering van de database ten gevolge van functionele afhankelijkheden vermeden worden. Ook kan de hoeveelheid redundante informatie geminimaliseerd worden en zou men attributen die vooral nulwaardes bevatten in het optimale geval vermijden.

Het globale idee om het design en/of de implementatie te verbeteren is efficiëntie. Dit zowel op vlak van onderhoud, data opvragen als geheugengebruik (dus geen redundante informatie opslaan).

7. De opdrachten uit het tweede deel introduceerden gradueel de verschillende concepten voor het bevragen van een database vanuit PHP. Bedenk en beschrijf een mogelijke (creatieve) uitbreiding van de web-interface voor de scheepsvaartdatabase, die de verschillende concepten gebruikt en gegevens uit de database combineert.

Een mogelijke uitbreiding van de web-interface zou een mogelijkheid zijn om alle trips en/of een bepaalde trip op een bepaalde route op te vragen aan de hand van een gegeven route_id en/of ship_id en hierna de informatie te updaten (bijvoorbeeld de aankomst/vertrekdatum een aantal dagen veranderen). Praktisch gezien kan dit handig zijn als de vertrek- of aankomstdatum veranderd moet worden door bijvoorbeeld een vertraging, ongeluk, storm,

Hiervoor moet men voor een gegeven route_id alle trips opvragen of voor een ship_id de trip_id opvragen. Hierna moet aan de hand van een form (zoals bij 'order shipments') de aangepaste gegevens aan de gebruiker vragen. Men kan hier bijvoorbeeld in html met checkboxes werken om bepaalde eigenschappen te kunnen aanpassen.