

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

Tussentijds verslag

TEAMLEDEN

1. Lies Bollens
2. Jakob Festraets
3. Pieterjan Beerden

SESSIE

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

De opdracht leek op zich niet zo heel moeilijk, maar nadat we met drie individueel de opdracht hadden volbracht, bleken er toch veel verschillen te zijn tussen onze versies. Het was mogelijk om de opdracht op verschillende manieren te interpreteren. Hierdoor was het nodig om te overleggen en uit onze verschillende versies de beste te selecteren. Dit heeft ons meer inzicht bijgebracht.

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

- Wij hadden geen klasse 'country' aangemaakt, omdat we het niet nodig vonden om voor deze opdracht country als een aparte entiteit te beschouwen. Country werd bij ons als een attribuut geïmplementeerd.
- De te betalen taks van een schip aan de haven hebben we beschouwd als een attribuut behorende bij de relatie tussen een haven en een schip. In de modeloplossing werd deze taks als een attribuut bij de haven beschouwd en was er ook geen directe relatie tussen een haven en een schip.
- Een 'shipment' werd bij ons beschouwd als een zwakke entiteit, met als identificerende relatie de 'customer' die het pakket verstuurd heeft. In de modeloplossing was 'shipment' een sterke entiteit.
- Volgens ons werd een 'shipment' door de 'shipbroker' op een 'container' geplaatst en was dit dus een ternaire relatie. De 'container' stond verder in relatie met de 'shipping line' die deze 'container' bevatte. Dit is dus een identificerende relatie. Een 'container' stond ook nog in een ternaire relatie met een 'trip' en een 'ship'. Deze 'trip' stond verder nog in relatie met de twee havens waartussen de route liep. In de modeloplossing heeft een 'shipment' een ternaire relatie met een 'container' en een 'trip'. Een 'container' staat voor de rest alleen in relatie met de 'shipping line' die de container bezit. Een 'trip' staat verder nog in een ternaire relatie met een 'route' en een 'ship'. Een trip wordt als een zwakke entiteit beschouwd met deze ternaire relatie als identificerende relatie, wat bij ons niet het geval was.
- Een eerste veronderstelling die we gemaakt hebben is dat een route wordt beschouwd als een sterke entiteit. Elke route heeft dus een eigen ID. Dit zorgt er voor dat de route niet afhangt van zowel de vertrek- als de aankomsthaven, wat volgens ons het geheel minder ingewikkeld maakt omdat er minder relaties moeten gecreëerd worden nu.
- Een tweede veronderstelling is dat elk pakket dat een klant wil versturen een zwakke entiteit is, behorende bij de klant die het pakket wil versturen. Elk pakket heeft een ID als sleutel. Samen met de ID van de klant zorgt dit ervoor dat elk pakket uniek bepaald is.

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?

Bij dit deel hebben we niet echt veel moeilijkheden ondervonden. In het begin wisten we niet dadelijk wat de verschillende stappen waren, maar aangezien deze in het boek zeer uitgebreid staan uitgelegd, konden we deze volgen en bevatte de opdracht niet zo veel moeilijkheden meer.

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)?

De grootste moeilijkheid was het leren van de syntax van PHP. Dit werkje was voor ons allemaal de eerste keer dat we in aanraking kwamen met PHP en het duurde even vooraleer we gewoon waren aan de syntax. Dit zorgde ervoor dat we, vooral in het begin, niet veel vooruitgang boekten en veel zaken nog moesten opzoeken. Een andere moeilijkheid was het doorgronden van de gegeven code. Omdat deze code al redelijk uitgebreid was, met veel hulpbestanden, moesten we bij elke vraag eerst een tijd bezig zijn met uitzoeken wat de code nu juist betekende. Anderzijds was het ook wel goed dat we al wat code kregen, omdat we zo rapper gevoel kregen met de PHP syntax. Verder bleek het ook vaak ingewikkeld om SEQUEL queries te maken die moesten samengesteld worden met attributen uit verschillende entiteiten.

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

Zoals al eerder vermeld wordt de route beschouwd als een sterke entiteit. Elke route heeft dus een eigen ID. Dit zorgt er voor dat de route niet afhangt van zowel de vertrek- als de aankomsthaven, wat volgens ons het geheel minder ingewikkeld maakt omdat er nu minder relaties moeten gecreëerd worden.

Qua implementatie hebben we ons gehouden aan alle verschillende bestanden die al aangemaakt waren. Door onze code op te splitsen in veel verschillende bestanden, zorgen we er voor dat het geheel overzichtelijk blijft. De verschillende entiteiten (bv. Een 'shipment', een 'order', ...) hebben we op een vrij gelijkaardige manier geïmplementeerd. Zo zijn alle SEQUEL queries in de specifieke mappers geplaatst en worden deze functies in de globaal gebruikte mapper gebruikt. Op deze manier is het makkelijk om voor een extra entiteit ook de benodigde code te schrijven en is het niet nodig om helemaal opnieuw te beginnen.

Een andere sterkte van onze implementatie is dat de meeste informatie in objecten wordt opgeslagen voordat het op de website getoond wordt. Hierdoor moeten we de info maar eenmalig opvragen en kunnen we ze daarna makkelijk opvragen en bewerkingen mee doen. Bovendien maakt dit de code zeer leesbaar en aanpasbaar.

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 totaal van design en implementatie van de database zou verbeterd kunnen worden met normalisatie, een techniek die in de hoorcolleges werd aangesneden. Dit zorgt voor het besparen van opslagruimte en het vermijden van dubbele opslag. Een database implementatie is optimaal wanneer ze zich in de 3^{de} normaalvorm bevindt, dit wil zeggen dat alle niet-sleutelattributen volledig functioneel afhankelijk moeten zijn van de primaire sleutel en alle attributen die niet tot een sleutel behoren niet mogen afhangen van een niet-sleutelattribuut.

Nog een verbetering aan het design van en het gebruik van de website is mogelijk een algemene zoekfunctie op pagina's waar lange lijsten van informatie op worden weergegeven. Eindeloos scrollen om een customer of een order te vinden zou dan uitgesloten zijn.

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 scheepsvaardatabase, die de verschillende concepten gebruikt en gegevens uit de database combineert.

Indien het systeem commercieel gebruikt zou worden, zou het nuttig zijn een inlogstelsel te maken. Op deze manier kan een klant enkel bekijken welke pakketjes hij zelf heeft verstuurd en geen pakketjes voor andere personen bestellen. Aangezien elk pakket een unieke ID heeft, kan de klant door middel van deze ID in een pagina in te geven, bekijken waar zijn pakket zich op dit moment bevindt (nog in de haven, op de boot, al onderweg vanuit de haven,...) en bekijken wanneer het aankomt. Wanneer een klant pakketten wil versturen naar verschillende klanten, kan hij de ID van elk pakket aan een van zijn klanten geven, die vervolgens ook kunnen kijken waar het pakket zich bevindt, maar niets kunnen aanpassen aan de database.

Met behulp van de informatie over de klanten kan je ook gemakkelijk gerichte reclame maken. Zo kan je gemakkelijk een groep van bepaalde klanten kiezen die een aantal pakketten op een bepaalde datum in een bepaalde plaats hebben verstuurd en deze informatie gebruiken om interessante reclames aan de klanten aan te bieden.