

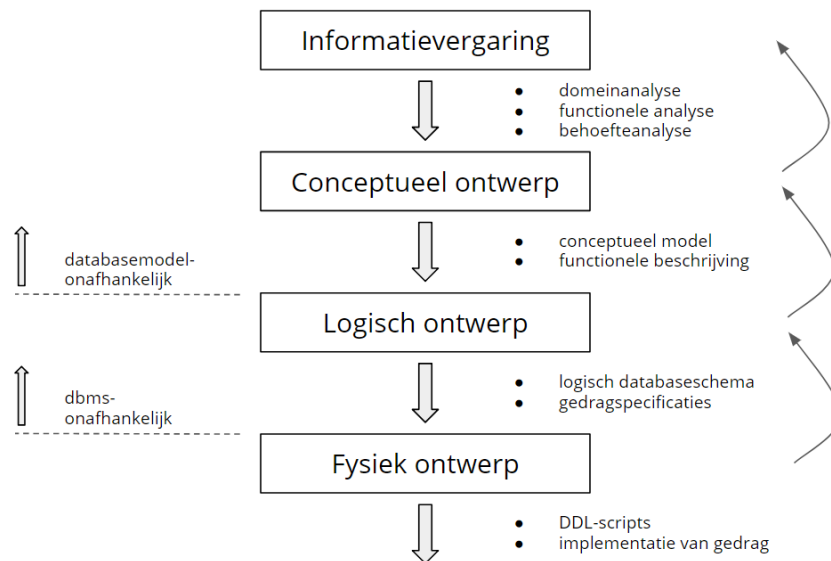
# Databanken: Ontwerpproject

---

## 1 Introductie

Gedurende de theorie- en oefeningenlessen maken jullie stap voor stap kennis met het ontwerpproces van een relationele databank. De theorielessen focussen hierbij vooral op het aanrijken van nieuwe (theoretische) concepten om het ontwerpproces zo correct en volledig mogelijk uit te voeren. Tijdens de oefeningenlessen passen we de leerstof toe op voorbeelden en bekijken we specifieke problemen en oplossingen die we vaak tegenkomen bij het ontwerpen van een databank. Ook werken we tijdens de oefeningenlessen een specifiek probleem (de projectopgave van vorig jaar) van begin tot eind, stap voor stap (klassikaal) uit.

De theorie- en oefeningenlessen zouden jullie een goede basis moeten geven om zelfstandig een volledige databank te ontwerpen. Om te bewijzen dat jullie hiertoe in staat zijn starten we, in parallel met de lessen, een databankontwerpproject. Gedurende dit project zijn jullie zélf verantwoordelijk voor het ontwerp, de implementatie en de bevraging van een databank ter ondersteuning van de Vlaamse toeristische sector. De motivatie achter dit overkoepelende project is vooral om jullie in de eerste plaats te laten kennismaken met het ontwerpproces. Daarnaast zullen jullie ook, zoals in het echte bedrijfsleven, creatieve oplossingen moeten bedenken voor problemen en moeten anticiperen op eventuele moeilijkheden en beperkingen van een databanksysteem. Voor we overgaan naar de praktische kant en de probleemstelling laten we jullie nog eens kort kennismaken met het hele ontwerpproces. Een voorstelling van dit proces vinden jullie in Figuur 1.



Figuur 1: Het databankontwerpproces

## 1.1 Het databankontwerpproces

Een eerste fase bestaat uit het **vergaren van alle noodzakelijke informatie**. Na deze fase weten we welke data opgeslagen moeten worden in de databank, welke betekenis deze data hebben, wat de functionaliteit is van een applicatie die de data gaat gebruiken, ... Deze fase resulteert in een domeinanalyse, een functionele analyse en een behoeftenanalyse. Dit is voor jullie ook het vertrekpunt van deze ontwerp-oefening en wordt toegelicht in Sectie 2.

Nadat alle informatie bekend is kunnen we deze informatie abstraheren. Dit resulteert in een abstracte voorstelling van de databank die het conceptueel model wordt genoemd. Het abstraheren van deze informatie in een conceptueel model en een functionele beschrijving heet het **conceptueel ontwerp**. Meer informatie hierover vinden jullie in Sectie 3.

In een derde fase kiezen we een databankmodel en zetten we het conceptueel model om in een logisch databankschema. Deze omzetting gebeurt door middel van een databankmodel-afhankelijk omzettingsalgoritme. Het omzetten van het conceptueel model naar een databankschema heet het **logisch ontwerp** en is verder uitgelegd in Sectie 4.

Tot slot kiezen we in de vierde en laatste fase een databankmanagementsysteem (dbms). In dit dbms kan het logisch ontwerp worden omgezet naar een **fysiek ontwerp** door middel van DDL-instructies, meestal in de vorm van DDL-scripts. Deze fase wordt toegelicht in Sectie 5.

Eens de databank ontworpen en geïmplementeerd is binnen een dbms kan ze effectief gebruikt worden om data te persisteren. Hiervoor is een dbms-specifieke databankmanipulatietaal voorzien. Een manipulatietaal ondersteunt het **invoeren, aanpassen, verwijderen** en **opzoeken** van data in een fysieke databank. Dit is voor jullie de laatste opdracht en is terug te vinden in Sectie 6.

Belangrijk om op te merken is dat dit ontwerpproces in geen geval sequentieel is en dat mogelijks fasen herzien moeten worden indien er problemen optreden tijdens latere fasen. Het is daarom ten allen tijde belangrijk om te communiceren met het management (de assistenten) en informatie te vragen indien er onduidelijkheden zijn.

## 1.2 Praktisch

Vooraleer jullie van start kunnen gaan willen we graag nog enkele praktische zaken aanhalen.

- Gedurende het jaar zullen er 6 **projectlessen** plaatsvinden waarin je aan je project kan werken. Deze lessen zullen niet volstaan om het hele project te maken. Best kunnen ze gebruikt worden om alle problemen, vragen of opmerkingen door te spelen aan de assistenten zodat zij jullie op de goede weg kunnen helpen.
- Dit project wordt gemaakt in **groepen** van 2 studenten. Dit betekent niet dat jullie niet met andere groepen mogen overleggen, maar onthoud wel dat het de bedoeling is dat jullie alles zelf kunnen voor het examen en dat wanneer er plagiaat wordt vastgesteld, er op gepaste wijze ingegrepen zal worden. Inschrijven kan via de groepenmodule van dit vak op Ufora **tot 11 oktober 2020, 23u30**. Ondervinden jullie problemen bij het vormen van groepen of bij het inschrijven, dan mogen jullie hiervoor steeds een e-mail sturen naar de assistenten.
- Het is mogelijk dat er tijdens het **examen** gepeild wordt naar jullie inzicht over dit project. Daarom wordt het ten eerste aangeraden om alle opdrachten zo goed mogelijk op te lossen.
- Dit project is **gequoteerd** en zal meetellen voor 30% van jullie niet-periodegebonden evaluatie. De andere 70% wordt bepaald door jullie oplossingen van de SQL-oefeningen. Om te slagen voor dit vak dienen jullie minstens 10/20 te halen voor de niet-periodegebonden evaluatie (dus de score op dit project samen met de score op de SQL-oefeningen).
- Het **quoteren** van dit project zal enkel en alleen gebeuren aan de hand van jullie fysieke databank op onze PostgreSQL server en de scripts die jullie gebruiken om de databank te vullen met data en te bevragen. Alles wat jullie als

extra materiaal indienen zal niet bekeken worden. Meer informatie in verband met het indienen en het quoteren vinden jullie in Sectie 7.

- Indien jullie een fout maken tegen de verwachte **vereisten** verliezen jullie alle punten voor dit project. Dit is onder andere het geval indien de bestandsnamen van de scripts niet correct zijn, de bestanden niet uitvoerbaar of onleesbaar zijn, er na de deadline wordt ingediend, . . . Controleer dus steeds dubbel of alles in orde is! Indien we, op welke manier dan ook, plagiaat of bedrog vaststellen, verliezen jullie alle punten voor dit project en volgen er verdere stappen.
- Indien je geslaagd bent voor de niet-periodengebonden evaluatie (10/20 of meer) kunnen jullie ervoor kiezen om de punten rechtstreeks **over te dragen** naar de tweede zittijd. Is dit niet het geval, dan dienen jullie opnieuw een project en SQL-oefeningen te maken in gewijzigde vorm.
- De **deadline** voor dit project is 14 december 2020 om 23u30 voor de studenten die het vak met code E018110 volgen en 18 december 2020 om 23u30 voor de studenten die het vak met code C003771 volgen.

Indien jullie vragen hebben kunnen jullie ons steeds contacteren door

- te mailen naar [toon.boeckling@ugent.be](mailto:toon.boeckling@ugent.be), [yoram.timmerman@ugent.be](mailto:yoram.timmerman@ugent.be), [joachim.peeters@ugent.be](mailto:joachim.peeters@ugent.be) en [milan.peelman@ugent.be](mailto:milan.peelman@ugent.be),
- na afspraak langs te komen in ons bureau (Sint-Pietersnieuwstraat 25, Technicum, verdieping -T, <https://soleway.ugent.be/routes/4453>). Zorg wel steeds dat je ons contacteert vooraleer je langskomt zodat er zeker iemand aanwezig is,
- tijdens de oefeningen- en projectlessen vragen te stellen.

Veel succes!

## 2 Informatievergaring

Vooraleer we van start kunnen gaan met het ontwerpen van een databank moeten we alle informatie die belangrijk is voor dit ontwerp verzamelen en onderzoeken. Hiervoor is het belangrijk om de algemene probleemstelling te kennen en te weten welke data de opdrachtgever wil kunnen opslaan in de databank. Ook is het belangrijk om te weten wat de betekenis van de verschillende data is. Dit resulteert in een **domeinanalyse**. Daarnaast moeten we ook de informatiestromen (herkomst van informatie, aanmaak van nieuwe informatie, . . .) analyseren. Deze **functionele analyse** kan resulteren in data en functionaliteiten die oorspronkelijk niet bekend

waren. Tot slot voeren we een **behoefteanalyse** uit waarbij kennis wordt vergaard over de gewenste functionaliteiten, de applicaties die de data gaan gebruiken en de manier waarop deze applicaties de data gaan gebruiken.

Aangezien jullie niet rechtstreeks met Toerisme Vlaanderen contact kunnen opnemen hebben wij reeds voor jullie deze fase uitgewerkt. Alle wensen van de instantie zijn in één beschrijving hieronder samengevat. Indien er onduidelijkheden of problemen zijn over bepaalde zaken kunnen jullie ons steeds om hulp vragen. Onthoud wel dat dit een oefening is op het zelf genereren van creatieve oplossingen en ideeën en dat alle informatie die jullie nodig hebben eigenlijk gegeven is in de beschrijving. Indien jullie het noodzakelijk achten mogen jullie steeds extra's toevoegen aan deze beschrijving.

## 2.1 Beschrijving

Toerisme Vlaanderen heeft jouw hulp nodig om een databank te ontwikkelen voor de opslag van gegevens met betrekking tot toeristische activiteiten en hotels in Vlaanderen.

Eerst en vooral is het de bedoeling dat in de te ontwerpen databank toeristische activiteiten kunnen worden opgeslagen. Elke toeristische activiteit heeft een naam en behoort tot een specifiek type (bv. 'Monument', 'Excursie', 'Natuur',...). Verder dient voor elke toeristische activiteit een beschrijving (optioneel) te worden opgeslagen, samen met het adres (straat en huisnummer (optioneel), postcode - bestaande uit 4 cijfers- en gemeente), de toeristische regio die verantwoordelijk is voor de activiteit, een telefoonnummer (optioneel), een website (optioneel) en de prijs per persoon. Een toeristische activiteit kan uniek worden geïdentificeerd door de combinatie van zijn naam en postcode. Belangrijk is dat voor iedere activiteit toegankelijkheidsinformatie bekend is. Deze informatie geeft weer of de activiteit toegankelijk is voor mensen met een specifiek type beperking. Een voorbeeld is dat de activiteit 'Kajakken in Gent' toegankelijk is voor doven en mensen met autisme. Voor elke activiteit dient in de databank expliciet te worden bijgehouden voor welke beperkingen de activiteit toegankelijk is. Aangezien je dit kan afleiden uit andere informatie opgeslagen in de databank, dien je niet op te slaan voor welke beperkingen een bepaalde activiteit niet toegankelijk is.

Personen kunnen zich inschrijven voor toeristische activiteiten. Ze doen dit tijdens een welbepaalde openingsperiode van de activiteit. Het is daarom belangrijk dat voor elke activiteit wordt bijgehouden wat de openingsperiodes zijn. Deze periodes worden voorgesteld door een begintijdstip (datum en tijd) en de duur van de opening. Openingsperiodes van eenzelfde toeristische activiteit mogen uiteraard niet overlappen. Het tijdstip waarop de reservatie wordt gemaakt en het aantal personen waarvoor gereserveerd wordt dienen ook opgeslagen te worden, alsook een aanduiding of de reservatie is bevestigd. Het tijdstip dient uiteraard te liggen voor het aanvangstijdstip van de activiteit zelf. Van de personen die de reservatie ma-

ken dienen voornaam, familienaam en een uniek e-mailadres te worden opgeslagen. De persoon die de reservatie maakt kan bij het reserveren zelf een aantal beperkingen opgeven van één of meerdere personen uit de ingeschreven groep. Op die manier kan de organisator van de activiteit de noodzakelijke maatregelen nemen om de groep goed te kunnen begeleiden. Voor elke reservatie moet, op basis van de toegankelijkheden van de gereserveerde activiteit en de opgegeven beperkingen bij reservatie, eenvoudig kunnen worden nagegaan of de reservatie voldoet aan de opgegeven beperkingen (indien de gereserveerde activiteit aangeeft dat ze toegankelijk is voor elk van de bij reservering opgegeven beperkingen) of er mogelijks problemen kunnen opduiken (indien een beperking wordt opgegeven bij reservatie die eigenlijk niet geschikt is voor de geboekte activiteit).

Naast toeristische activiteiten wil Toerisme Vlaanderen de databank ook gebruiken om hotels te kunnen opslaan. Van elk hotel dient een uniek hotelid, een naam, het aantal sterren (optioneel), een beschrijving (optioneel), het adres (straat, huisnummer, postcode -bestaande uit 4 cijfers- en gemeente), een e-mail (optioneel), de toeristische regio waartoe het hotel behoort en de minimumprijs per nacht (optioneel) te worden opgeslagen. Net zoals toeristische activiteiten kunnen personen ook hotels boeken. Hierbij moet het boekingstijdstip, het aantal personen waarvoor de boeking gemaakt wordt, een aanduiding van de boekingsbevestiging en de begin- en einddatum (waarbij de begindatum steeds strikt voor de einddatum ligt) van de boeking worden opgeslagen. Een persoon kan gedurende eenzelfde periode slechts 1 hotel boeken. Ook mag je ervan uit gaan dat een persoon slechts 1 boeking voor een hotel/reservatie voor een activiteit per tijdstip kan indienen. Tot slot is het belangrijk om te vermelden dat sommige hotels korting aanbieden op bepaalde toeristische activiteiten. Dit percentage korting dient te worden opgeslagen. Een hotel kan enkel korting geven voor een toeristische activiteit indien het hotel en de activiteit zich in dezelfde toeristische regio bevinden.

## 2.2 Bijkomende info

Om de informatievergaring af te sluiten, willen we nog een aantal zaken duidelijk maken in verband met de operaties die uitgevoerd kunnen worden op de data. Dit is een hulp bij het implementeren van de beperkingen later in dit project. Ten eerste moet het ten allen tijde mogelijk zijn data toe te voegen. Daarnaast is het ook mogelijk om alle data te verwijderen met de restrictie dat dit enkel en alleen kan als ze niet meer gebruikt worden elders in de databank. Een voorbeeld is dat een hotel enkel verwijderd kan worden indien er geen enkele boeking van dit hotel meer in de databank is opgeslagen. Tot slot is het in geen geval mogelijk om data die reeds in de databank zit aan te passen. Met aanpassingen moet je dus geen rekening houden.

### 3 Conceptueel ontwerp

In dit deel focussen we ons op het abstraheren en modelleren van de informatie beschreven in 2.1. Voor deze stap is het belangrijk dat jullie alle informatie reeds uitgebreid hebben geanalyseerd en de probleemstelling goed hebben begrepen. Nu moeten jullie deze informatie omzetten naar een abstracte voorstelling van de databank. Hiervoor gebruiken we het conceptueel **Enhanced Entity-Relationship-diagram** (EER-diagram). Een nadeel van dit diagram is dat het geen ondersteuning biedt voor het weergeven van de functionaliteit en het gedrag van de abstracte concepten en dat bepaalde beperkingen niet afgedwongen kunnen worden. Daarom is het belangrijk om alles wat niet voorgesteld kan worden door het diagram neer te schrijven in een **functionele beschrijving**.

Aangezien jullie het conceptueel ontwerp reeds uitgebreid hebben bestudeerd in vorige lessen behandelen we dit hier niet meer in detail. Voor meer informatie kunnen jullie steeds de slides van de oefeningenlessen 'Conceptueel ontwerp' of Hoofdstuk 3 in het boek 'Principes van databases' raadplegen.

#### 3.1 Opdrachten

1. Zet de beschrijving gegeven in 2.1 om naar een EER-diagram. Let op de juiste notaties en zorg ervoor dat je diagram zo volledig mogelijk is.
2. Alles wat niet weergegeven kan worden in het EER-diagram moet gespecificeerd worden in een aparte functionele beschrijving.

Let bij het conceptueel ontwerp goed op volgende zaken.

- Neem de opgave verschillende keren goed door en probeer alle informatie die je leest te modelleren. Hou steeds de beschrijving van het probleem bij de hand zodat je niets vergeet in de omzetting. Wees dus volledig!
- Vraag je steeds af of je conceptueel model strookt met je eigen interpretatie en met de beschrijving. De bedoeling is dat je model aantoont dat je weet waarover het probleem gaat en dat je de beschrijving juist interpreteert.
- Zorg voor het consistent zijn van notaties in het diagram en de functionele beschrijving.
- Hoe leesbaarder en eenvoudiger het conceptueel ontwerp is, hoe makkelijker de volgende fasen zullen zijn. Besteed genoeg tijd aan dit ontwerp en controleer dubbel of alles klopt met de beschrijving. Indien je vragen, opmerkingen of problemen hebt, mag je steeds de assistenten contacteren.

## 4 Logisch ontwerp

Wanneer het conceptueel model en de functionele beschrijving ontworpen zijn kan er een databankmodel gekozen worden. Daarbij moet er steeds een afweging worden gemaakt tussen de voor- en nadelen van de databankmodellen in de context van het gestelde probleem. Voor de eenvoudigheid en om uiteindelijke inconsistenties in de dataopslag te vermijden kiezen we hier voor het relationele databankmodel. Het logisch databankontwerp bestaat uit de omzetting van het EER-diagram naar een **relationeel databankschema** door middel van een omzettingsalgoritme. Ook wordt er geredeneerd over eenvoudige relatie-beperkingen (uniciteit, optionaliteit, domeinrestricties, . . .) en functionele beperkingen. Voor meer informatie verwijzen we naar de oefeningenlessen 'Logisch ontwerp', en naar Hoofdstuk 4 en 5 in het boek 'Principes van databases'.

### 4.1 Opdrachten

1. Zet het EER-diagram dat jullie hebben ontworpen in 3.1 om naar een relationeel databankschema.
2. Benadruk expliciet in je relationeel databankschema de verschillende eenvoudige beperkingen (sleutels, uniciteit, optionaliteit, domeinrestricties, . . .).
3. Redeneer over de (extra) functionele beperkingen van je relationeel databankschema op basis van de vereisten beschreven in 2.2.

Let bij de omzetting goed op volgende zaken.

- Doe de omzetting van je conceptueel ontwerp naar een relationeel databankschema zoals gezien in de oefeningenlessen. Wees zeker dat je niets vergeet om te zetten vanuit het EER-diagram. Hou hierbij de beschrijving gegeven in 2.1 en 2.2 goed in de gaten.
- Schrijf nauwgezet uit hoe en waarom je bepaalde beslissingen neemt voor de omzetting. Zo is het makkelijker om later aanpassingen te maken en te weten wat je hebt gedaan.
- Zorg ervoor dat de benamingen en verwoording van je schema in lijn liggen met je EER-diagram en functionele beschrijving uit de vorige fase.
- Probeer zoveel mogelijk informatie in je schema kwijt. Vermeld alle zaken die je denkt niet te kunnen omzetten.
- Lees na de omzetting je relationeel databankschema na en redeneer over de data die later in dit project in de relaties zal worden opgeslagen.
- Voor vragen, opmerkingen of problemen mag je steeds de assistenten contacteren.



## 5 Fysiek ontwerp

De logische ontwerpsfase resulteert in een relationeel databankschema en in een verzameling beperkingen. Eens dit alles voorhanden is kunnen we beginnen met het fysiek databankontwerp. Daarvoor moet eerst een relationeel databankbeheersysteem (dbms) gekozen worden en nadien moeten alle elementen uit de vorige fase geïmplementeerd worden. Voor dit probleem kiezen we voor het open-source dbms PostgreSQL.

De implementatie van een relationele databank gebeurt door middel van de gestandaardiseerde Structured Query Language (SQL). Voor het fysiek databankontwerp worden instructies gebruikt die gecategoriseerd worden onder de datadefinitietaal (DDL) van SQL. Meestal worden SQL- (DDL-) scripts gebruikt voor het gemakkelijk opslaan en uitvoeren van deze instructies. Na deze fase heb je een **werkende databank** bestaande uit tabellen aangevuld met beperkingen zoals primaire sleutels, vreemde sleutels, CHECK-beperkingen, triggers,... en ben je klaar om ze te vullen met data.

Vooraleer jullie beginnen met dit fysiek ontwerp is het aangeraden om de workshops 'Introductie PostgreSQL', 'Fysiek ontwerp' en (later) 'Triggers, functies & views' te maken. Hierin maken jullie kennis met het beheren, opzetten en implementeren van een fysieke relationele databank in PostgreSQL. Voor meer informatie kunnen jullie ook steeds Hoofdstuk 6 van het boek 'Principes van databases' raadplegen.

### 5.1 Opdrachten

1. Connecteer met jullie persoonlijke databank op onze server via de verkregen connectieparameters.
2. Implementeer jullie relationeel databankschema.
3. Implementeer alle vereisten en beperkingen zoals beschreven in 2.1 en 2.2.

Let bij het fysiek ontwerp goed op volgende zaken.

- Zorg ervoor dat je een zo volledig mogelijke omzetting doet. Dit maakt het makkelijker om in de volgende fase je data correct in te voeren. Herbekijk tijdens het implementeren steeds de probleemstelling.
- Zet al je DDL-statements om de databank te implementeren samen in 1 groot SQL-script. Dit script kan in de volgende fase ook aangevuld worden met statements om de data in de databank te importeren. Door in het begin van het script alle tabellen, beperkingen, triggers,... te verwijderen kan je dit script steeds opnieuw uitvoeren wanneer je iets hebt aangepast. Dit zorgt er ook voor dat je steeds van een lege (en dus propere) toestand opnieuw kan starten zonder dat er wijzigingen verloren gaan.

- De implementatie van jullie relationele databank wordt gedaan tijdens (en na) de projectles 'Fysiek ontwerp'. De implementatie van de triggers, functies en views is voor tijdens en na de projectles 'Triggers, functies & views'.
- Indien je een foutmelding krijgt bij het uitvoeren van een statement, probeer dan te achterhalen wat dit betekent (door bv. de foutmelding op te zoeken) en denk goed na waarom dit zou kunnen gebeuren.
- Voor vragen, opmerkingen of problemen mag je steeds de assistenten contacteren.

## 5.2 Indienen

Voor deze fase dienen jullie geen extra materiaal in te dienen. We zullen deze fase enkel beoordelen op basis van jullie fysieke databank die zich bevindt op onze server. Zorg er dus voor dat deze databank volledig afgewerkt is tegen de deadline en dus alle nodige

- tabellen,
- functies en triggers,
- aanvullende beperkingen (zoals sleutels, uniciteit, domeinrestricties,...) en
- views

bevat.

## 6 Datamanipulatie

Het fysiek ontwerp resulteert in een werkende databank die klaar is voor gebruik. In deze databank zorgen alle gedefinieerde elementen ervoor dat de gewenste functionaliteit effectief aanwezig is. Wanneer alles geïmplementeerd en getest is, kunnen we overgaan naar de laatste fase van het ontwerpproces: de datamanipulatie. In deze fase is het de bedoeling om de ontworpen databank te gaan gebruiken. Dit gebeurt door de databank te vullen met data (zie 6.1). Daarna is het mogelijk om deze data te gaan analyseren (zie 6.2).

Het toevoegen, aanpassen en verwijderen van data in een databank door middel van SQL-queries is aan bod gekomen in de workshop 'Data importeren' en 'Datamanipulatie'. In de oefeningenlessen SQL hebben jullie daarnaast gefocust op het bevragen en analyseren van data door middel van (complexe) SQL-queries. Voor meer informatie kunnen jullie ook steeds Hoofdstuk 6 van het boek 'Principes van databases' raadplegen.

## 6.1 Importeren

Een eerste stap in deze fase bestaat eruit om data te importeren in jouw aange-maakte databank. Het management heeft voor jullie drie .csv bestanden voorzien waarin alle data die ze in de databank willen hebben reeds zijn opgeslagen. Deze bestanden hebben volgende naam en inhoud.

- `activiteitreservaties.csv`: Dit bestand bevat alle data met betrekking tot toeristische activiteiten en het reserveren van toeristische activiteiten door personen.
- `hotelboekingen.csv`: Dit bestand bevat alle data met betrekking tot hotels en het boeken van hotels door personen.
- `kortingen.csv`: Dit bestand bevat alle data met betrekking tot hotels die kortingen aanbieden op toeristische activiteiten.

Let op: alle .csv-bestanden beginnen met een header op de eerste rij die niet geïm-porteerd moet worden als data in de databank. Gebruik daarnaast ook de bestanden enkel voor de beschreven doeleinden, m.a.w. gebruik `kortingen.csv` niet voor het importeren van bv. hotels.

### 6.1.1 Opdrachten

Om jullie op weg te helpen hebben wij reeds een .sql-script voorzien voor de aan-maak van 3 super-relaties (`super_activiteitreservaties`, `super_hotelboekingen` en `super_kortingen`) waarin je de data uit de .csv-bestanden makkelijk kan impor-teren. Jullie taak is nu om zo goed mogelijk de eigenlijke data niet-redundant te gaan verspreiden over jullie tabellen ontworpen in Sectie 5. Met 'niet-redundant' wordt bedoeld dat er in geen enkele tabel rijen dubbel mogen voorkomen. Daar-naast is het ook belangrijk dat het proberen importeren van foutieve data (bv. een boeking die ingediend is na de periode van de boeking) opgevangen en afgehan-deld wordt door het databanksysteem. Het is dus belangrijk dat alle voorwaarden en beperkingen die in de informatievergaringsstap werden geïdentificeerd, effectief worden geïmplementeerd in je databank. Test dit grondig, want ook hierop zullen wij een groot deel van de punten geven!

Let bij het importeren van data goed op volgende zaken.

- Voor het toevoegen van data is er een .sql-script gegeven voor de aanmaak van 3 super-relaties, zoals aangehaald hierboven. Het is noodzakelijk voor het cor-rect quoteren van het project dat alle data uit `activiteitreservaties.csv`, `hotelboekingen.csv` en `kortingen.csv` in respectievelijk de `super_activiteitreservaties`, `super_hotelboekingen` en `super_kortingen`

relaties worden geïmporteerd. Vanuit deze super-relaties kunnen ze dan verspreid worden naar jouw eigen relaties.

- Alle data in de .csv bestanden zijn geformateerd als tekstuele data. Het is dus aan jou om de juiste conversie te doen naar de datatypes van de attributen in jouw relaties.
- Indien je een foutmelding krijgt bij het uitvoeren van een statement, probeer dan te achterhalen wat dit betekent (door bv. de foutmelding op te zoeken) en denk goed na waarom dit zou kunnen gebeuren.
- Voor vragen, opmerkingen of problemen mag je steeds de assistenten contacteren.

### 6.1.2 Indienen

Zorg er op het einde van de datamanipulatie-fase voor dat jullie fysieke databank op onze server correct en volledig gevuld is met de gegeven data. Ook is het noodzakelijk dat jullie 1 .sql-script indienen met de naam `db_importeren_groepX.sql` waarbij je 'X' vervangt door je groepsnummer. Dit bestand bevat alle INSERT-statements voor de verspreiding<sup>1</sup> van data vanuit de 3 gegeven super-relaties naar jullie tabellen. Let op, het is niet mogelijk om dit script automatisch te genereren via bv. `pg_dump`. Je dient dus de INSERT-statements die je doorheen het project hebt gebruikt, manueel in hetzelfde bestand te plaatsen! Het is opnieuw **zeer** belangrijk dat dit script correct, volledig en zonder onderbrekingen uitvoert! Als dit niet het geval is, kunnen wij jullie project helaas niet verbeteren en quoteren. Test dus grondig dat je script effectief doet wat het zou moeten doen!

## 6.2 Analyse

Nu alle data in de databank zitten, is het mogelijk om te testen of dit correct is gebeurd door middel van analyses.

### 6.2.1 Opdrachten

Aangezien jullie ondertussen experts zijn in het schrijven van SQL-queries is het jullie taak om een aantal tellingen te doen van data in de databank. De vragen waarop jullie als antwoord een SQL-query moeten schrijven staan hieronder opgelijst.

1. Tel het totale aantal unieke boekingen die werden geregistreerd (zowel op hotels als toeristische activiteiten)

---

<sup>1</sup>Je mag er dus vanuit gaan dat de data reeds aanwezig is in de super-relaties, dus het importeren van de .csv-bestanden naar de super-relaties moet niet worden opgenomen in het script.

2. Tel het aantal unieke personen die een boeking plaatste voor meer dan 1 persoon
3. Tel het aantal unieke hotels
4. Tel het aantal unieke activiteiten die faciliteiten voorzien voor mensen met een beperking
5. Tel het aantal unieke hotelboekingen die geboekt werden voor een periode/tijdspanne van 2 dagen of meer
6. Tel het aantal unieke openingsperiodes

### 6.2.2 Indienen

Na het analyseren van de data dienen jullie opnieuw 1 .sql-script in met de naam `db_queries_groepX.sql` waarbij je 'X' vervangt door je groepsnummer. Dit bestand bevat alle SQL-queries die op bovenstaande vragen een antwoord geven. Je dient dit bestand opnieuw (zoals bij het importeren van de data) zelf aan te maken, door de query's er manueel in te plakken. Elke regel in jullie oplossingsbestand moet 1 query bevatten die verplicht eindigt met een ';'. Let op, net zoals bij de vorige scripts is het opnieuw **zeer** belangrijk dat dit script correct en volledig uitvoert! Als dit niet het geval is, kunnen wij jullie project helaas niet verbeteren en quoteren.

## 7 Indienen

Om het correct quoteren van dit project mogelijk te maken is het noodzakelijk om

1. Een volledig geïmplementeerde en gevulde databank te hebben op onze server, en
2. 2 .sql-scripts in te dienen, zijnde
  - `db_importeren_groepX.sql` met alle statements voor de verspreiding van data vanuit de 3 gegeven super-relaties naar jullie tabellen, en
  - `db_queries_groepX.sql` met alle SQL-queries die antwoord geven op bovenstaande analyse-vragen.

Hierin moeten jullie vanzelfsprekend 'X' vervangen door jullie groepsnummer. Deze .sql-bestanden dienen jullie te bundelen in een .zip-bestand met de naam `db_project_groepX.zip` waarbij je 'X' opnieuw vervangt door je groepsnummer. Dit .zip-bestand sturen dienen jullie **voor** de deadline in via de 'Opdrachten'-module op Ufora.

Zoals reeds vermeld zal dit project voor 30% van jullie niet-periodegebonden evaluatie meetellen. Alle punten staan op volledigheid en kwaliteit van de oplossing. Dit zal automatisch beoordeeld worden door uitvoering van een verbeterscript dat zal testen of voldaan is aan alle vereisten en beperkingen en of alle data aanwezig zijn in jullie databank.