

Het examen is volledig open boek. Er mag gebruik gemaakt worden van al het materiaal voorzien tijdens de lessen. Dit houdt in:

- Het boek 'Principes van databases' van prof. dr. Guy De Tré
- Alle slides, opgaves, (opgeloste) oefeningen, notities, examenvoorbereidingen,...

Het examen kan worden opgelost op lege witte bladeren. **Los elk deel van het examen (sectie 1-4) op op een apart blad.** Vermeld zeker op **elk** oplossingsblad je studentnummer, naam en richting.

Opgelet: omwille van de coronamaatregelen kunnen wij tijdens het examen geen vragen beantwoorden. Indien er vragen of onduidelijkheden zijn, schrijf deze dan nauwgezet op je oplossingsblad zodat wij hier tijdens het quoteren rekening mee kunnen houden.

1 Open vragen

Beantwoord onderstaande vragen.

1. Waarom mag een catalogus niet door een gebruiker worden aangepast?
2. Beschouw de 'meer-niveau'-relatie in Figuur 1
 - a) Beschrijf wat er zal gebeuren indien een gebruiker met toegangsniveau V de periode van 'Mona Lisa' wil aanpassen naar 1498.
 - b) Beschrijf wat er zal gebeuren indien een gebruiker met toegangsniveau P de periode van 'Mona Lisa' wil aanpassen naar 1498.
3. Beschouw vier concurrente transacties T1, T2, T3 en T4 die op timestamps 1 t.e.m. 10, de gegevens A, B en C lezen (R) en/of schrijven (W) zoals is weergegeven in Tabel 1. Zoals je ziet gebeuren er nog geen rollbacks of commits. Veronderstel dat een standaard locking-protocol wordt gebruikt zoals beschreven in de cursus.
 - a) Teken een 'wacht-voor'-graaf voor deze situatie.
 - b) Doet er zich een situatie van deadlock voor? Verklaar.
4. Waarom kunnen BASE-systemen geen volledige consistentie garanderen, terwijl ACID-systemen dat wel kunnen?

Tabel Schilderij primaire sleutel = {S_ID, B ₁ } vreemde sleutel = {Eigenaar, B ₅ }										
S_ID: char(3)	B ₁ : T _B	Naam: varchar	B ₂ : T _B	Periode: integer	B ₃ : T _B	Waarde: real	B ₄ : T _B	Eigenaar: varchar	B ₅ : T _B	B: T _B
S01	P	Vissershuis	P	1882	P	16.000.000	G	Boijmans	V	P
S03	P	Mona Lisa	P	1499	V	30.000.000	V	Louvre	P	P
S03	G	Mona Lisa	G	1499	G	75.000.000	G	Louvre	G	G

Figuur 1: 'meer-niveau' relatie

	1	2	3	4	5	6	7	8	9	10
T1	R(A)						R(C)	W(A)		
T2		R(A)							R(C)	
T3				R(C)	R(B)					W(B)
T4			R(C)			W(C)				

Tabel 1: Transacties

2 Conceptueel ontwerp

Een **immoweb**site heeft een nieuwe databank nodig ter ondersteuning van hun on-line diensten. Om alle informatie hieromtrent correct en gestructureerd te kunnen opslaan in een databank, is er uiteraard in eerste instantie een conceptueel ontwerp nodig. Enkele databankspecialisten ontwierpen zo een conceptueel ontwerp op basis van onderstaande **informatie die rechtstreeks uit de informatievergaderingsfase komt. Dit conceptueel ontwerp kan je vinden in Appendix 1.**

De immoweb site stelt twee soorten panden, namelijk huizen en appartementen, voor aan kandidaat-kopers. Deze appartementen en huizen moeten kunnen worden opgeslagen in de databank. Er mogen in heel de databank geen twee panden voorkomen met hetzelfde adres (en ieder pand heeft uiteraard een adres). Belangrijk is dat alle huizen en appartementen die op de website staan ook effectief worden aangeboden! Zowel huizen als appartementen die aangeboden worden hebben telkens een bijhorend gebruikersprofiel. Dit is de persoon of instantie die het huis te koop stelt op de immoweb site. Ieder gebruikersprofiel wordt uniek geïdentificeerd door een username. Onderliggend zijn er twee types gebruikers die huizen of appartementen te koop aanbieden: immobiliënkantoren en personen. Voor personen dient hun voor- en achternaam te worden opgeslagen en voor immobiliënkantoren dient de naam van het bedrijf te worden opgeslagen. Daarnaast dienen voor een immobiliënkantoor ook de overeenkomstige filialen te worden opgeslagen. Een dergelijk filiaal

heeft een naam die uniek is binnen het immobiliënkantoor waartoe het behoort. Tot slot dient altijd eenduidig te kunnen worden afgeleid wie de eigenaar is van het pand dat te koop staat. In het geval dat een pand wordt aangeboden door een persoon is de eigenaar van het pand altijd deze persoon zelf. In het geval dat een pand wordt aangeboden door een immobiliënkantoor is het kantoor zelf niet de eigenaar, maar dient te worden bijgehouden welke persoon eigenaar is van het pand.

Beantwoord nu onderstaande vragen.

1. In het EER-diagram dat de ontwerpers opstelden zitten één of meer syntactische fouten. Dit zijn fouten die indruisen tegen de regels van hoe een correct EER-diagram er moet uitzien en zijn dus geen fouten tegen de semantiek van het probleem waarvoor een ontwerp wordt opgesteld. Pas het EER-diagram in Appendix 1 aan zodat er geen fouten tegen de EER-syntax (zoals gezien in de oefeningenlessen) meer aanwezig zijn. Het is dus niet de bedoeling om semantische fouten te verbeteren, maar zorg ervoor dat je bij het verbeteren van de syntactische fouten geen (extra) semantische fouten introduceert in het diagram. Hou dus rekening met de informatie uit de informatievergaringfase. Je mag je aanpassingen rechtstreeks op het EER-diagram in Appendix 1 maken. Zorg wel dat je Appendix 1 zeker mee indient met je antwoordenblad!
2. Een belangrijke vereiste die vermeld staat in de informatievergaring is op dit moment nog niet afgedwongen in het conceptueel ontwerp. Om welke vereiste gaat het? En op welke manier kan je deze vereiste zo elegant mogelijk afdwingen? Maak dit duidelijk door het deel van het conceptueel ontwerp dat relevant is voor deze vraag te hertekenen (je hoeft niet het hele conceptueel ontwerp te hertekenen, we gaan ervan uit dat het stuk dat je niet tekent, hetzelfde blijft in jouw nieuw conceptueel ontwerp). Verklaar je antwoord.
3. Werk verder op het conceptueel ontwerp dat je bekomen hebt na het oplossen van **vraag 2**. Kunnen er in dit ontwerp relatietypes worden weggelaten/samengevoegd zonder dat je informatie verliest? Denk bijvoorbeeld na of het met dit ontwerp mogelijk is dat een pand wordt aangeboden door persoon A, maar eigendom is van persoon B? Zo ja, hoe zou je dit kunnen oplossen door modificatie van het EER-diagram (dus niet door toevoegingen aan de functionele beschrijving!)? Zijn er ook mogelijke nadelen verbonden aan je wijzigingen? Bespreek je antwoord tekstueel, eventueel ondersteund met een tekening of schets van hoe je het zou aanpakken.
4. Werk verder op het conceptueel ontwerp dat je bekomen hebt na het oplossen van **vraag 2** (je hoeft dus geen rekening te houden met je antwoord op vraag 3). Breid het ontwerp uit zodat kandidaat-kopers (personen met een gebruikersprofiel) online een afspraak kunnen maken om het pand waarin ze geïnteresseerd zijn te bezichtigen. Panden zijn enkel bezikbaar tijdens bepaalde

openingstijden (bestaande uit een begintijdstip en een duur) die specifiek zijn voor het gegeven pand. Het tijdstip (dat ligt binnen een openingstijd van het pand) waarop de kandidaat-koper effectief zal langskomen, moet ook worden opgeslagen in de databank. Openingstijden van eenzelfde pand mogen uiteraard niet overlappen. Je hoeft opnieuw niet het hele conceptueel ontwerp te hertekenen op je antwoordblad, maar maak wel duidelijk waar jouw toevoeging(en) aansluiting zou(den) vinden op het ontwerp dat je hebt bekomen na oplossing van vraag 3. Maak ook duidelijk of er aanpassingen aan de functionele beschrijving moeten gebeuren.

3 Logisch ontwerp

Geef formeel (dus niet met voorbeelden!) een situatie aan waarvoor je geen CHECK-beperking kan gebruiken, maar genooddaakt bent om bij de omzetting naar het logisch ontwerp het gewenst gedrag bij toevoegen van data te onderzoeken (en hiervoor later een trigger te implementeren).

4 Workshops

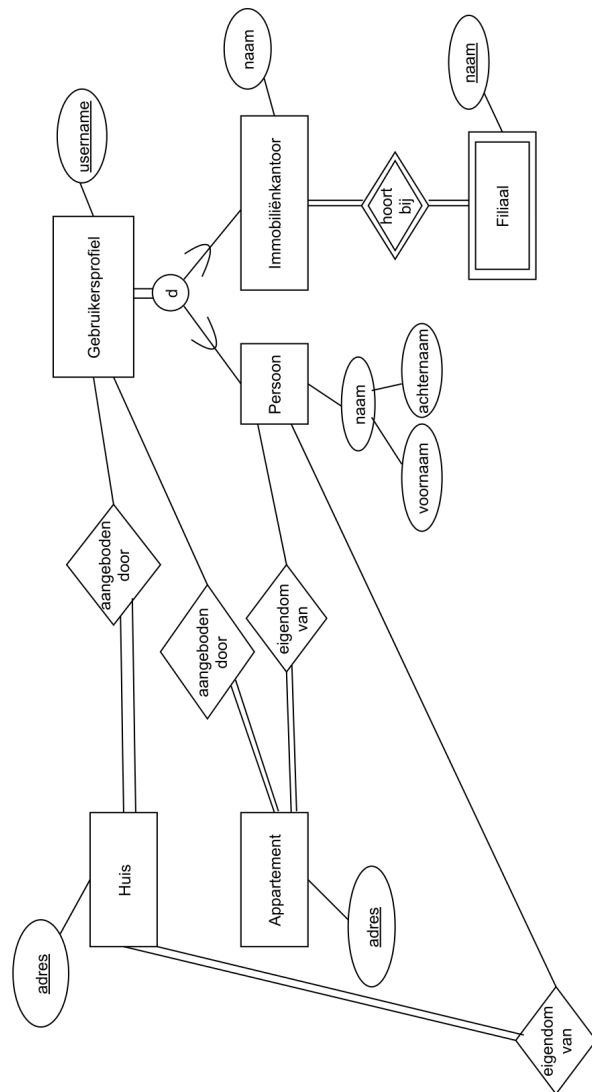
In Appendix 2 is de DDL-code gegeven van een databank voor de opslag van kampeerplekken in België. Bekijk deze DDL-code nauwgezet en los daarna onderstaande vragen op.

1. Waarom is het niet noodzakelijk om, bij het aanmaken van de tabel bivakzone, privezone en faciliteit, opnieuw een CHECK-beperking te definiëren, die stelt dat de waarden in de locatie kolommen hoogstens uit 20 karakters mogen bestaan, ook al is dit een vereiste?
2. Waar/niet waar. Het verwijderen van een (refererende) rij uit de tabel bivakzone door middel van een DELETE-statement zorgt ervoor dat de gerefereerde rij uit de tabel kampeerplek automatisch ook verwijderd wordt. Verklaar je antwoord.
3. In Appendix 3 is de DDL-code gegeven voor de aanmaak van een superrelatie met de naam super_kamperen. Daarnaast toont Tabel 2 ook een aantal voorbeeldrijen die in deze superrelatie zouden kunnen zitten. Tot slot worden er een aantal INSERT-statements gegeven. De bedoeling van deze INSERT-statements is om de data uit de supertabel op een correcte en volledige manier te importeren in de verschillende tabellen. Er zitten echter een aantal fouten in deze INSERT-statements, waardoor niet alle data correct kunnen worden geïmporteerd. Beschrijf per fout nauwgezet wat het probleem is en hoe je dit kan oplossen, zodanig dat de data wel correct en volledig aan de tabellen worden

toegevoegd. Bij het beoordelen van een INSERT-statement op zijn correctheid mag je er vanuit gaan dat alle voorgaande INSERT-statements (waar nodig) reeds werden gecorrigeerd en dus reeds data op een correcte manier aan de geassocieerde tabellen hebben toegevoegd.

4. Jef en Marie hebben een probleem en ze vragen jouw hulp. Bij het verspreiden van de data vanuit de superrelatie `super_kamperen` naar de tabel `bivakzone`, krijgen ze een foutmelding. De foutmelding stelt dat het toevoegen van een rij aan de tabel `bivakzone` de vreemde sleutelbeperking gedefinieerd in de tabel `bivakzone` schendt. Doe een voorstel van iets dat Jef en Marie kunnen proberen om hun probleem op te lossen. Let op, deze vraag is niet gebaseerd op de INSERT-statements gegeven in Appendix 3, maar is gebaseerd op een mogelijk importeurscript dat Jef en Marie hebben geschreven.
5. Op dit moment is het mogelijk dat een bepaalde locatie zowel voorkomt in de tabel `bivakzone` alsook in de tabel `privezone`. Hoe zou je het fysieke ontwerp aanpassen zodat een bepaalde locatie enkel kan worden toegevoegd in de tabel `bivakzone` of in de tabel `privezone` (maar niet in beide tabellen)? Beschrijf nauwgezet in woorden (dus niet met code!).

Appendix 1: Conceptueel ontwerp immowebste



Functionele beschrijving

- personen kunnen enkel een pand aanbieden als ze er eigenaar van zijn.

Appendix 2: DDL-code kampeerdatabank

```
CREATE TABLE kampeerplek
(
    locatie    VARCHAR PRIMARY KEY CHECK ( length(locatie) <= 20 ),
    naam       VARCHAR,
    aantal_pers INTEGER NOT NULL CHECK ( aantal_pers > 0 )
);

CREATE TABLE bivakzone
(
    code        VARCHAR PRIMARY KEY,
    locatie     VARCHAR UNIQUE NOT NULL,
    CONSTRAINT bivakzone_fk FOREIGN KEY
        (locatie) REFERENCES kampeerplek (locatie)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE privezone
(
    locatie     VARCHAR PRIMARY KEY,
    host        VARCHAR NOT NULL,
    CONSTRAINT privezone_fk FOREIGN KEY
        (locatie) REFERENCES kampeerplek (locatie)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE faciliteit
(
    locatie     VARCHAR NOT NULL,
    faciliteitnaam VARCHAR NOT NULL,
    CONSTRAINT faciliteit_pk PRIMARY KEY (locatie, faciliteitnaam),
    CONSTRAINT faciliteit_fk FOREIGN KEY
        (locatie) REFERENCES kampeerplek (locatie)
);
```

Appendix 3: DDL-code importeren

```
CREATE TABLE super_kamperen
(
    locatie      VARCHAR PRIMARY KEY,
    naam         VARCHAR,
    aantal_pers INTEGER,
    code         VARCHAR,
    host         VARCHAR,
    wifi         BOOLEAN,
    water        BOOLEAN,
    toilet       BOOLEAN
);
```

locatie	naam	aantal_pers	code	host	wifi	water	toilet
A	Tuin	4	NULL	Jan	true	true	true
B	Bos	2	1	NULL	false	false	true
C	NULL	2	NULL	Piet	false	true	true

Tabel 2: super_kamperen

```
INSERT INTO kampeerplek (locatie, naam, aantal_pers)
    SELECT DISTINCT locatie, naam, aantal_pers FROM super_kamperen;

INSERT INTO bivakzone (code, locatie)
    SELECT DISTINCT locatie, code FROM super_kamperen;

INSERT INTO privezone (locatie, host)
    SELECT locatie, host FROM super_kamperen WHERE host IS NOT NULL;

INSERT INTO faciliteit (locatie, faciliteitnaam)
    SELECT locatie, 'wifi' FROM super_kamperen WHERE wifi = true
    INTERSECT
    SELECT locatie, 'water' FROM super_kamperen WHERE water = true
    INTERSECT
    SELECT locatie, 'toilet' FROM super_kamperen WHERE toilet = true;
```