

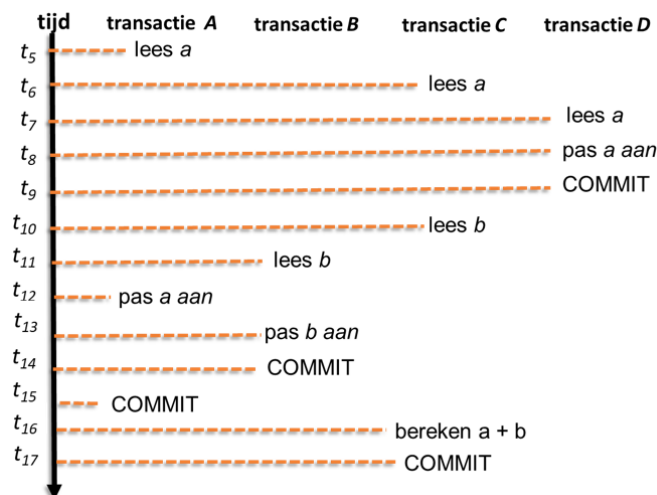
- Het examen is volledig open boek. Je mag al het materiaal gebruiken dat voorzien is tijdens de lessen (bv. het boek 'Principes van databases' van prof. dr. Guy De Tré, slides, opgaves, (opgeloste) oefeningen, . . .) of dat je zelf hebt gebruikt ter verwerking van de leerstof (bv. samenvattingen, notities, . . .).
- Het examen dient opgelost te worden op witte bladeren. Gebruik voor elk deel van het examen (sectie 1-4) een apart blad. Vermeld zeker op elk oplossingsblad je studentnummer, naam en richting.
- 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. (Het relationeel databankmodel) Beschouw een basisrelatie met schema `Wedstrijddeelname({RennerID:varchar, Wedstrijd:varchar, Jaar:integer, Uitslag:varchar})`. Geef de algebraïsche expressie waarmee je de twee tuples
 - `(RennerID:10012, Wedstrijd:Parijs-Roubaix, Jaar:2021, Uitslag:3)`,
en
 - `(RennerID:10032, Wedstrijd:Parijs-Roubaix, Jaar:2021, Uitslag:32)`kan toevoegen aan de basisrelatie. Als je gebruik maakt van extra basisrelaties dien je deze volledig en correct mee te geven.
2. (Herstel na falen) Geef bij elk van onderstaande uitspraken aan of ze *juist* of *fout* zijn en motiveer daarbij telkens kort waarom.
 - a) Wanneer men werkt met schaduwpagina's heeft men geen databankbuffers meer nodig.
 - b) Controlepunten zorgen ervoor dat het DBMS sneller kan herstellen na een soft-crash.

- c) Controlepunten zorgen ervoor dat het DBMS sneller kan herstellen na een hard-crash.
 - d) Bij een 'no steal' flushing strategie kan het DBMS volledig vastlopen zodat er niet meer verder gewerkt kan worden.
 - e) De 'write-ahead log'-regel zorgt ervoor dat de consistentie-eigenschap van transacties is voldaan.
3. (NoSQL) Leg kort uit waarom het bevragen van een NoSQL databank voor een gebruiker doorgaans moeilijker is dan het bevragen van een relationele databank.
4. (Delen van gegevens) Beschouw vier concurrent uitgevoerde transacties *A*, *B*, *C* en *D* die respectievelijk starten op tijdstippen t_1 , t_2 , t_3 en t_4 . Deze transacties zouden zonder concurrency-controle worden uitgevoerd zoals aangegeven in Figuur 1.
- a) Welke concurrency problemen doen zich voor bij de uitvoering gegeven in Figuur 1. Leg kort uit waarom.
 - b) Pas de 'locking'-methode toe op de uitvoering gegeven in Figuur 1.



Figuur 1: Uitvoering transacties

2 Conceptueel ontwerp

Een kleine vliegbase zou graag over een relationele databank beschikken waarin, op een gestructureerde wijze, data zitten opgeslagen met betrekking tot vliegtui-

gen, services en staanplaatsen. Ze vragen hiervoor aan jou om een volledig en correct conceptueel ontwerp op te stellen op basis van onderstaande informatie.

Op de vliegbasis staan er meerdere vliegtuigen op een vaste staanplaats binnen een hangar. Elk vliegtuig heeft een uniek registratienummer en behoort tot één vliegtuigtype. Elk vliegtuigtype heeft een uniek modelnummer, een oppervlakte die nodig is om vliegtuigen van dit type te plaatsen (in m^2) en (optioneel) een capaciteit in zake het aantal passagiers.

De vliegbasis beschikt over meerdere hangars met één of meerdere staanplaatsen. Elke hangar heeft een uniek nummer op de vliegbasis. De staanplaatsen hebben per hangar een uniek nummer en een bijhorende oppervlakte (in m^2).

Voor elk vliegtuig moet het huidige onderhoudscontract worden opgeslagen. Dit onderhoudscontract heeft een uniek contractnummer en een bijhorende contactpersoon (eigenaar). De contactgegevens van deze persoon bestaan uit een voornaam, een achternaam, een telefoonnummer, een adres en een uniek e-mailadres.

De verschillende services (onderhoudswerken) worden bijgehouden. Per service wordt het vliegtuig, de technicus die de service heeft uitgevoerd, de datum, het begin- en het eindtijdstip geregistreerd. Een technicus heeft dezelfde gegevens als een contactpersoon. Daarnaast kan een technicus per tijdstip slechts aan 1 vliegtuig werken, en kan hij/zij enkel werken aan vliegtuigen die een type hebben waarvoor hij/zij is opgeleid. Tot slot moet het totaal aantal gepresteerde uren per maand per technicus eenvoudig opgevraagd kunnen worden uit de databank.

De databank dient uiteraard enkel data te bevatten die voldoen aan de opgegeven beperkingen alsook aan beperkingen die logisch af te leiden vallen uit bovenstaande informatie.

3 Logisch en fysiek ontwerp

Geef, voor elke voorwaarde die hieronder gegeven is, aan of je deze voorwaarde kan afdwingen met een eenvoudige CHECK-beperking, zonder gebruik te maken van een (sub)query of een (zelfgedefinieerde) functie in de code van de CHECK-beperking. Leg ook steeds uit waarom je dit (niet) kan. In het geval je een voorwaarde *niet* kan afdwingen met een eenvoudige CHECK-beperking, geef een type beperking waarmee je dit idealiter wel kan doen.

1. De waarden onder attribuut a in basisrelatie R moeten uniek zijn.
2. Er mag geen NULL-waarde voorkomen onder attribuut a in basisrelatie R .
3. De waarden onder attribuut a_1 in basisrelatie R_1 moeten voorkomen als waarden onder attribuut a_2 in basisrelatie R_2 .

4. De waarden onder attribuut *a* in basisrelatie *R* moeten beginnen met de letter 'b'.
5. De waarden onder attribuut *a* in basisrelatie *R* moeten strikt positief zijn als de waarden onder attribuut *b* in basisrelatie *R* strikt negatief zijn.

4 Workshops

In Appendix 1 is de DDL-code gegeven voor de aanmaak van een databank waarin data in verband met trainingen van leden bij personal trainers in een fitnessclub worden opgeslagen. Bekijk deze DDL-code nauwgezet en los daarna onderstaande vragen op.

1. Waar/niet waar. Een lid kan trainingen hebben bij verschillende trainers op eenzelfde begintijdstip. Verklaar je antwoord.
2. Waarom heeft het geen zin om SET NULL of SET DEFAULT te kiezen als verwijderstrategie voor de beperking met naam training_trainer_fk? Wat gebeurt er momenteel als je data verwijdert uit de tabel trainer die vanuit de tabel training worden gerefereerd?
3. In Appendix 2 is de inhoud van een .csv bestand gegeven dat data bevat die geïmporteerd moeten worden in de tabellen lid, trainer en training. Geef de id's van de rijen die tijdens het importeren *potentieel* problemen kunnen opleveren rekening houdend met de DDL-code gegeven in Appendix 1. Je mag veronderstellen dat de data eerst worden toegevoegd aan de tabel lid, dan aan trainer en dan aan training en dat de data correct geformateerd zijn. Verklaar duidelijk voor elke aangeduide rij waarom deze rij problemen zou kunnen opleveren.
4. Stel dat de databank reeds gevuld is met data, zoals weergegeven in Appendix 3. Teken de *volledige* tabel die zal worden teruggegeven als resultaat van onderstaande query.

```
SELECT DISTINCT x.nummer, x.aant AS aant
  FROM (SELECT t.nummer, COUNT(tr.nummer) AS aant
        FROM trainer t
             LEFT JOIN training tr ON t.nummer = tr.nummer
        GROUP BY t.nummer) x
      JOIN (SELECT t.nummer, COUNT(tr.nummer) AS aant
            FROM trainer t
                 LEFT JOIN training tr ON t.nummer = tr.nummer
            GROUP BY t.nummer) y
        ON y.aant > x.aant
ORDER BY x.aant DESC;
```

Appendix 1: DDL-code fitness databank

```
CREATE TABLE lid (  
    naam VARCHAR PRIMARY KEY  
);  
  
CREATE TABLE trainer (  
    nummer          INTEGER PRIMARY KEY CHECK ( nummer > 0 ),  
    begin_contract DATE      NOT NULL,  
    duur_contract   INTERVAL NOT NULL CHECK (duur_contract > '1 month'::INTERVAL)  
);  
  
CREATE TABLE training (  
    nummer          INTEGER,  
    begintijd       TIMESTAMP,  
    duur            INTERVAL NOT NULL CHECK (duur > '0 minutes'::INTERVAL),  
    naam            VARCHAR NOT NULL,  
    CONSTRAINT training_pk PRIMARY KEY (nummer, begintijd),  
    CONSTRAINT training_lid_fk FOREIGN KEY (naam) REFERENCES lid (naam),  
    CONSTRAINT training_trainer_fk FOREIGN KEY (nummer) REFERENCES trainer (nummer)  
);  
  
CREATE FUNCTION function1() RETURNS trigger AS  
$BODY$  
BEGIN  
    IF EXISTS(  
        SELECT 1 FROM training  
            WHERE nummer = NEW.nummer  
                AND NEW.begintijd < begintijd + duur  
                AND NEW.begintijd + NEW.duur > begintijd) THEN  
        RAISE EXCEPTION 'foutmelding';  
    END IF;  
    RETURN NEW;  
END;  
$BODY$  
LANGUAGE plpgsql;  
  
CREATE TRIGGER trigger1  
    BEFORE INSERT ON training  
    FOR EACH ROW EXECUTE PROCEDURE function1();
```

Appendix 2: .csv data fitness databank

id	lid	trainer	begin_contract	duur_contract	begintijd_training	duur_training
1	Ann	1	14/01/2022	12 maanden	17/01/2022 09:00:00	30 minuten
2	Ann	2	14/01/2022	6 maanden	17/01/2022 09:00:00	30 minuten
3	Ann	1	14/01/2022	12 maanden	17/01/2022 09:20:00	30 minuten
4	Ann	2	14/01/2022	6 maanden	19/01/2022 09:00:00	30 minuten
5	Piet	2	14/01/2022	6 maanden	19/01/2022 09:00:00	30 minuten
6	Ann	3	14/01/2022	1 maand	20/01/2022 09:30:00	30 minuten
7	Marie	2	14/01/2022	6 maanden	25/01/2022 09:00:00	NULL

Tabel 1: .csv data fitness databank.

Appendix 3: Voorbeelddata SQL-query fitness databank

<u>naam</u>	nummer	begin_contract	duur_contract
Ann	1	14/01/2022	12 maanden
Piet	2	14/01/2022	12 maanden
	3	14/01/2022	12 maanden

Tabel 2: Voorbeelddata tabel lid.

Tabel 3: Voorbeelddata tabel trainer.

nummer	begintijd	duur	naam
1	17/01/2022 09:00:00	30 minuten	Ann
1	18/01/2022 09:00:00	30 minuten	Ann
1	20/01/2022 09:00:00	30 minuten	Ann
2	19/01/2022 09:00:00	30 minuten	Piet
2	25/01/2022 09:00:00	30 minuten	Piet

Tabel 4: Voorbeelddata tabel training.