

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 studentenummer, 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

Het relationeel databankmodel

Vraag 1

Waarom is het voor gebruikers niet toegestaan om de catalogus aan te passen?

Vraag 2

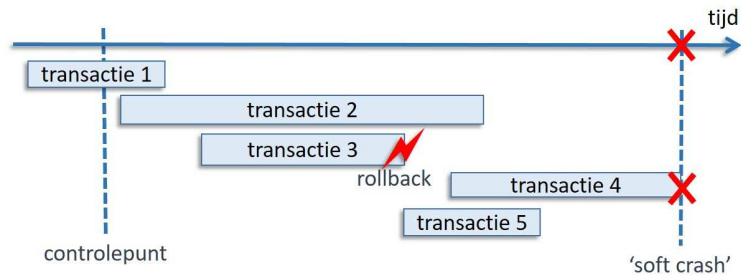
Waarom worden alternatieve sleutels gebruikt in de praktijk?

Objecttechnologie in databanken

Het naamgevingsmechanisme en bereikbaarheidsmechanisme spelen samen een belangrijke rol bij het vastleggen van objectpersistentie.

Vraag 3

Op welke manier maakt het ODMG model gebruik van het bereikbaarheidsmechanisme?



Figuur 1: Situatie van 'soft crash'.

Vraag 4

Op welke manier maakt SQL:2011 gebruik van het bereikbaarheidsmechanisme?

Beveiliging tegen falen

Beschouw het transactieverloop dat is weergegeven in Figuur 1.

Vraag 5

Waar of niet waar? Een hersteltechniek met uitgestelde aanpassing zal in bovenstaande situatie trager werken dan een techniek met onmiddellijke aanpassing. Verklaar.

Vraag 6

Waar of niet waar? De finale Redo-lijst bij onmiddellijke aanpassing bevat twee transacties. Verklaar.

Vraag 7

Waar of niet waar? Zonder controlepunten is het herstel eenvoudiger, maar neemt het veel meer tijd in beslag. Verklaar.

Vraag 8

Waar of niet waar? Voor een volledig herstel moet het deel van het logbestand tussen het controlepunt en het moment van falen 4 maal worden doorlopen. Verklaar.

2 Ontwerp

De overheid heeft je nodig in de strijd tegen het coronavirus! Stel een **conceptueel ontwerp** op voor de ontwikkeling van een databank die de contactopsporing van besmette patiënten in Vlaanderen moet ondersteunen.

Om te beginnen moet in de databank elke contactopspoorder kunnen worden opgeslagen. Een contactopspoorder heeft een uniek werknemersnummer. Verder moeten ook de voor- en achternaam, het geslacht ('m', 'v' of 'x'), het telefoonnummer, het e-mailadres en het bureaunummer waar de persoon werkt in het callcenter worden opgeslagen.

Contactopspoorers krijgen contactgegevens binnen van potentieel besmette personen (verder patiënten genoemd). Deze contactgegevens bestaan uit de voornaam, de achternaam, het geslacht ('m', 'v' of 'x'), het telefoonnummer en het e-mailadres van de persoon. De voornaam en achternaam van personen zijn uiteraard niet noodzakelijk uniek!

Om na te gaan met wie patiënten in de voorbije twee weken contact hebben gehad, belt een contactopspoorder hen op. Ieder telefoontje tussen contactopspoorder en patiënt moet worden geregistreerd in de databank, samen met het begin- en eindtijdstip van het telefoontje.

Tijdens een telefoontje wordt de patiënt ondervraagd over de contacten die hij/zij de voorbije 14 dagen heeft gehad met andere personen. Elk van deze contacten dient te worden geregistreerd in de databank, samen met het begin- en eindtijdstip van het contact (datum en uur) en de contactgegevens (voor- en achternaam, geslacht, e-mailadres, telefoonnummer) van de contactpersoon. Vanaf het moment dat contactpersonen gekend zijn door de contactopsporing, dienen deze personen zelf te worden opgebeld, en dus dienen deze personen op zichzelf te worden opgeslagen in de databank als patiënten. Verder dient ook een beoordeling van het risico van het contact tussen beide patiënten te worden opgeslagen ('laag risico', 'risicovol' of 'hoog risico'), samen met een beschrijving van het contact, en moet in de databank kunnen achterhaald worden tijdens welk telefoontje het contact werd geregistreerd. Aangezien contacten slechts worden opgeslagen tot 14 dagen terug in de tijd, mag de tijd tussen een geregistreerd contact en het telefoontje waarin het contact werd geregistreerd niet meer dan 14 dagen bedragen.

Op basis van de informatie uit de telefoongesprekken wordt tot slot bijgehouden welke personen in quarantaine moeten en welke personen moeten worden getest. Voor iedere persoon dienen alle geassocieerde testen te worden bijgehouden, samen met het tijdstip waarop de test is gebeurd en het resultaat van de test. Bovendien worden voor iedere persoon alle quarantaineperiodes opgeslagen, samen met het begintijdstip en eindtijdstip van de quarantaineperiode.

3 Gebruik

In Appendix 1 is een PostgreSQL DDL-script gegeven voor de opbouw van een databank die gebruikt wordt in een online vragentool. Hieronder volgen er een aantal vragen in verband met het gebruik van deze databank.

Vraag 1

Gegeven de inhoud van de tabel gebruiker

nickname	email
pieters	piet.pieters@ugent.be
jean	jan.janssens@ugent.be

en van de tabel vraag

id	inhoud	opgelost	gepost_door
1	Wat is een vreemde sleutel?	false	pieters
2	Ik snap er niks van	false	jean

Hieronder is een (genummerde) sequentie van SQL-commando's gegeven die inwerken op deze databank. Hoe zal de resultatentabel eruit zien na uitvoering van het laatste commando? Teken deze tabel **volledig**. Je mag veronderstellen dat de tabel stem initieel leeg is en de syntax van de commando's correct is. Ook worden alle statements één voor één uitgevoerd (en dus niet in één keer). Statements die falen mag je dus negeren. We houden geen rekening met de correcte formatering van waarden in de getekende tabel.

1. INSERT INTO vraag(id, inhoud, opgelost, gepost_door)
VALUES (3, 'Waarvoor staat SQL?', false, 'pieters');
2. INSERT INTO stem VALUES ('pieters', 1, 1);
3. UPDATE vraag SET opgelost = true WHERE inhoud = 'Ik snap er niks van';
4. INSERT INTO stem VALUES ('jean', 2, -1);
5. INSERT INTO stem VALUES ('jean', 1, 1);
6. INSERT INTO stem VALUES ('pieters', 1, 1);
7. INSERT INTO stem VALUES ('pieters', 3, 1);
8. INSERT INTO vraag(id, inhoud, opgelost, gepost_door)
VALUES (4, 'Ben ik geslaagd?', false, 'jean');
9. SELECT * FROM view1;

Vraag 2

Aangezien gebruikers vaak op vragen stemmen die ze zelf hebben gepost, willen de databankbeheerders een trigger implementeren die ervoor zorgt dat dit niet meer mogelijk is.

1. Op welke tabel moet deze trigger gedefinieerd worden?
2. Leg stap voor stap, in eigen woorden (zonder code!), uit welke logica je moet implementeren in de functie die gekoppeld wordt aan deze trigger.

4 Analyse

In Appendix 2 is een logisch databankschema te vinden van (een deel van) de resto databank. Hieronder is een SELECT-query gegeven die inwerkt op deze databank. Analyseer de SELECT-query aandachtig en beantwoord vervolgens onderstaande vragen.

```
1.  SELECT DISTINCT r1.restonaam, r1.beschrijving FROM resto r1
2.      INNER JOIN aanbod a1 USING (restonaam)
3.      INNER JOIN productsoort ps1 USING (productnaam, soortnaam)
4.      INNER JOIN product p USING (productnaam)
5.  WHERE vegetarisch = 'vegetarisch'
6.  GROUP BY r1.restonaam, r1.beschrijving
7.  HAVING AVG(minimumprijs) <= ALL (SELECT AVG(minimumprijs)
8.                                  FROM product p2
9.                                  INNER JOIN productsoort ps2
10.                                 USING (productnaam)
11.                                 INNER JOIN aanbod a2
12.                                 USING (productnaam, soortnaam)
13.                                 INNER JOIN resto r2
14.                                 USING (restonaam)
15.                                 WHERE p2.vegetarisch = 'vegetarisch'
16.                                 GROUP BY r2.restonaam)
17.  ORDER BY r1.restonaam
```

Vraag 1

De tabel die het resultaat is na het uitvoeren van bovenstaande query is een antwoord op een specifieke analysevraag. Geef een gedetailleerde en een zo volledig mogelijke beschrijving van deze vraag.

Vraag 2

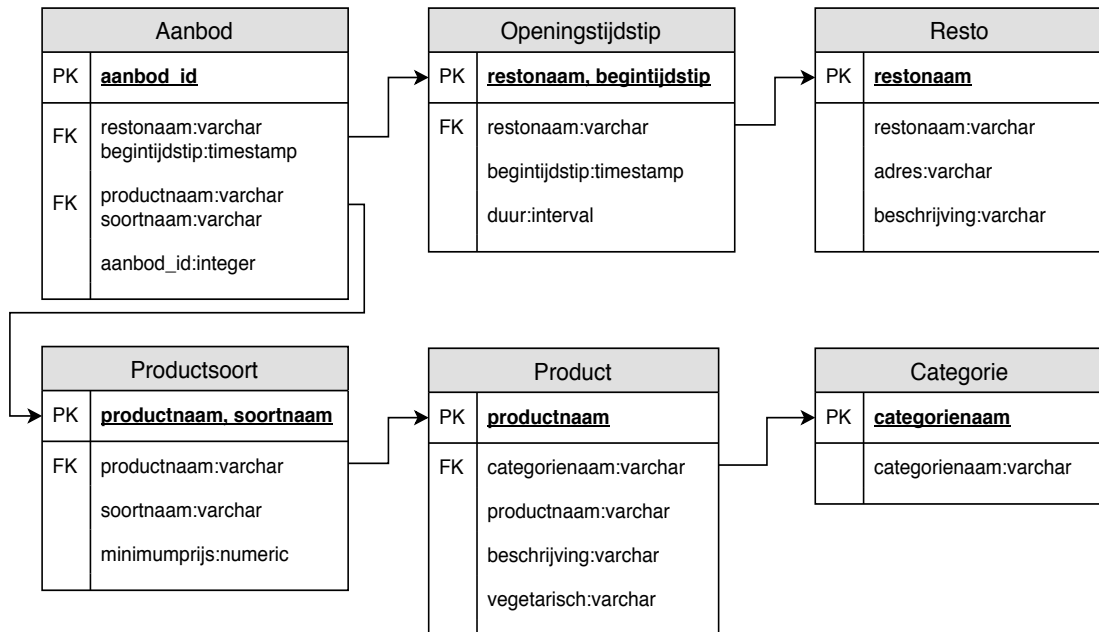
Een aantal zaken in de gegeven query zijn overbodig of kunnen geoptimaliseerd worden. Geef 2 mogelijke optimalisaties die ervoor zouden kunnen zorgen dat de uiteindelijke uitvoeringskost zal dalen zonder dat de resultatentabel verschilt. Geef een gedetailleerde en een zo volledig mogelijke beschrijving van elke voorgestelde optimalisatie (zodat we eenduidig kunnen bepalen of dit correct is). Je mag ervan uit gaan dat de code correct uitvoert. Ga dus niet op zoek naar mogelijke syntaxfouten.

Appendix 1: DDL-script vraagdatabank

```
CREATE TABLE gebruiker (  
    nickname CHARACTER VARYING,  
    email CHARACTER VARYING UNIQUE NOT NULL,  
    CONSTRAINT gebruiker_pkey PRIMARY KEY (nickname)  
);  
  
CREATE TABLE vraag (  
    id SERIAL,  
    inhoud CHARACTER VARYING NOT NULL,  
    opgelost BOOLEAN NOT NULL,  
    gepost_door CHARACTER VARYING NOT NULL,  
    CONSTRAINT vraag_pkey PRIMARY KEY (id),  
    CONSTRAINT vraag_gebruiker_fkey  
        FOREIGN KEY (gepost_door) REFERENCES gebruiker(nickname)  
);  
  
CREATE TABLE stem (  
    nickname CHARACTER VARYING,  
    vraag_id INTEGER,  
    score INTEGER NOT NULL,  
    CONSTRAINT stem_pkey PRIMARY KEY (nickname, vraag_id),  
    CONSTRAINT stem_gebruiker_fkey  
        FOREIGN KEY (nickname) REFERENCES gebruiker(nickname),  
    CONSTRAINT stem_vraag_fkey  
        FOREIGN KEY (vraag_id) REFERENCES vraag(id),  
    CONSTRAINT stem_score_check  
        CHECK (score IN (-1, 1))  
);  
  
CREATE VIEW view1 AS  
    SELECT v.id, v.inhoud,  
        CASE WHEN SUM(s.score) IS NULL THEN 0 ELSE SUM(s.score) END AS score  
    FROM vraag v  
        LEFT JOIN stem s ON v.id = s.vraag_id  
    WHERE NOT opgelost  
    GROUP BY v.id  
    ORDER BY score DESC;
```

Appendix 2: Logisch databankontwerp Resto

In onderstaande figuur vind je een schematische weergave van een mogelijk logisch ontwerp voor (een deel van) de resto databank. Hierbij wordt iedere basisrelatie weergegeven door een rechthoek, die bovendien een oplijsting van alle attributen met bijhorend datatype bevat. Daarnaast worden de attributen die behoren tot de primaire sleutel bovenaan weergegeven, en worden vreemde sleutels voorgesteld door een pijl tussen de betreffende attribuutverzamelingen. Alle extra beperkingen die niet kunnen weergegeven worden in dit schema worden onderaan opgelijst.



Extra beperkingen

1. Resto:
 - UNIQUE en NOT NULL: adres
 - Optioneel: beschrijving
2. Product:
 - CHECK: vegetarisch \in {'geen van beide', 'vegetarisch', 'veganistisch'}
 - Optioneel: beschrijving
3. Productsoort:
 - CHECK: minimumprijs ≥ 0
4. Openingstijdstip:

- CHECK: duur > 0
- Insert: check dat openingsperiode niet overlapt met andere openingsperiode van dezelfde resto.

5. Aanbod:

- Insert: check dat, indien de openingsperiode verder loopt dan 11u30, er geen producten uit categorie 'Ontbijt' aangeboden worden.