

Examen Databanken
Prof. dr. Guy De Tré
Theoriegedeelte

Academiejaar 2016-2017
1^o zittijd
dinsdag 10 januari 2017, 13u

Schriftelijk, gesloten boek

Vraag 1: Relationeel databankmodel

- a) Wat is een databaseschema? (1pt)
- b) Geef vier componenten die kunnen voorkomen in een relationeel databaseschema? (2pt)
- c) Hoe kan je bij het werken met een relationele database te weten komen wat het databaseschema is? (1pt)

Vraag 2: Toegangsmogelijkheden voor applicaties

- a) Wat is een cursor? (1pt)
- b) Leg uit hoe je met een cursor werkt. (2pt)

Vraag 3: Delen van gegevens

- a) Beschrijf het basisprotocol voor 'timestamping'. (2pt)
- b) Waarom beschermt dit protocol niet tegen 'dirty read'? (1pt)

Oefeningengedeelte

Er mag enkel gebruik gemaakt worden van:

- Het boek 'Principes van databases' van prof. Guy De Tré
- Eventueel de slides uit de theorielessen van prof. Guy de Tré
- De samenvatting van de PostgreSQL handleiding
- De samenvatting van de MongoDB handleiding

10/01/2016

1 SQL

De vragen over het gedeelte SQL hebben betrekking op de databank garage. Zie Figuur 1.1 voor het schema van deze databank.

1.1 Zoek de fout in de query

De volgende SQL vragen werden gesteld in de SQL Exerciser. Er worden ook antwoorden gegeven, maar daar kunnen fouten ingeslopen zijn. Zoek de fout(en) (0, 1 of meer) en motiveer waarom het fout is. Het volstaat hier dus niet om zelf een volledige correcte query te formuleren.

1.1.1 Solisten

Geef alle auto's terug die door één enkele techniek (mechanic) in één enkele garage ("Garage De Leeuw") zijn behandeld. We verwachten minstens de kolom 'carid' als resultaat.

```
select distinct carid from intervention f where garageid = (select  
garageid from garage where name = "Garage De Leeuw") and not exists  
(select * from intervention where mechanic != f.mechanic);
```

1.1.2 Vergelijkende studie: onderhoud van wagens uit 2011

Geef voor elk model uit 2011 het merk, de modelnaam en het gemiddeld aantal onderhoudsinterventies ("maintenance") per wagen. Rond het gemiddelde af op twee decimalen. Geef geen <NULL>-waarden in het resultaat weer, maar gebruik het cijfer 0. We verwachten minstens de kolommen: 'brand', 'modelname' en 'avgnumber'.

```
select brand, modelname, round(avg(amount),2) avgnumber from model inner
join (select modelid, carid, count(*) amount from intervention left join
car using (carid) where type = "maintenance") using (modelid) where year
= 2011 group by modelid, modelname;
```

1.2 Stel de SQL query op

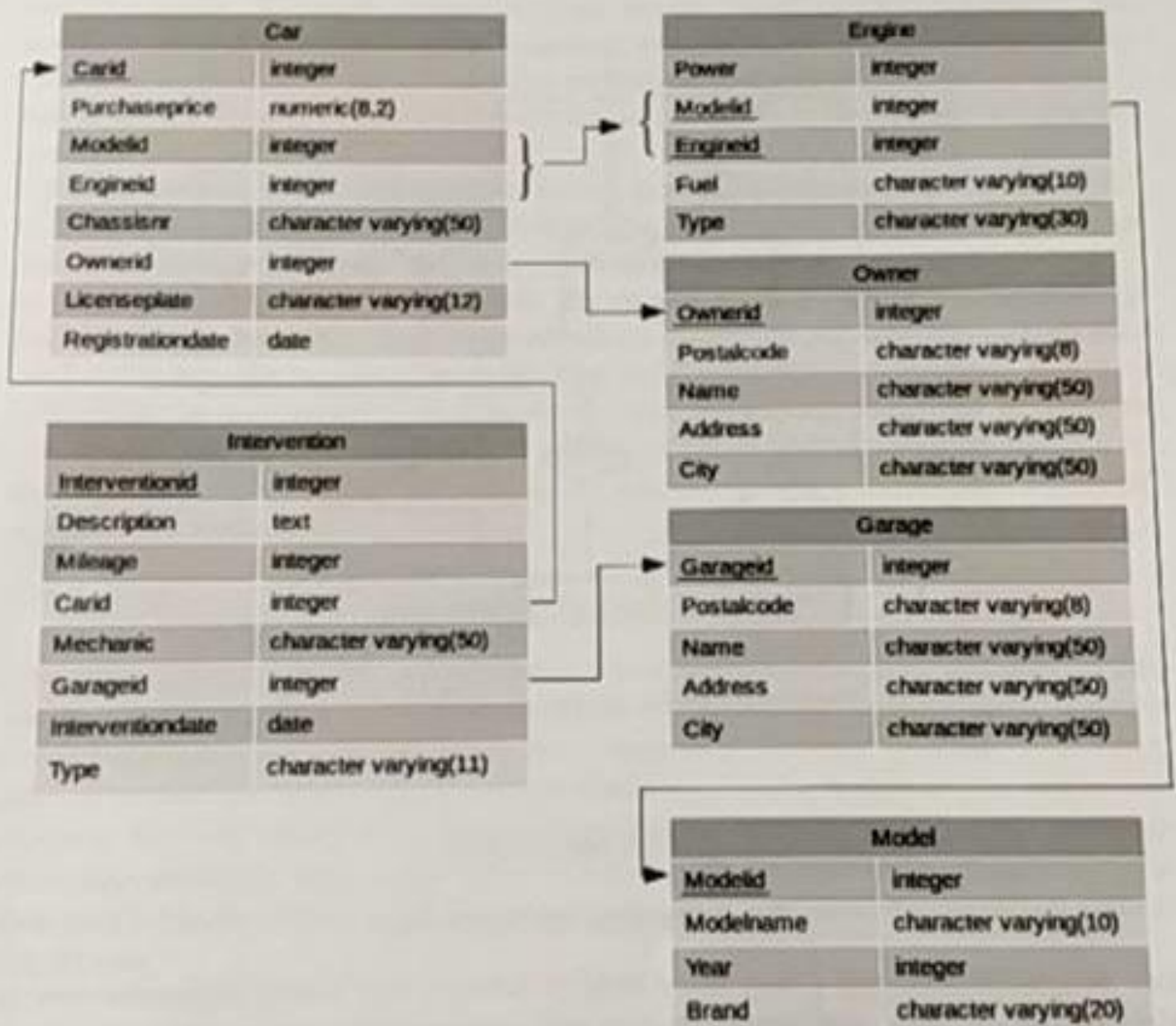
In dit gedeelte dien je een volledig correcte SQL query neer te schrijven die voldoet aan de opdracht (zoals in de SQL Exerciser).

1.2.1 Zeer onsuccesvol model

Geef alle modellen die niet werden verkocht. We verwachten minstens de kolommen 'modelname', 'brand' en 'year'.

1.2.2 Interventies met ervaring

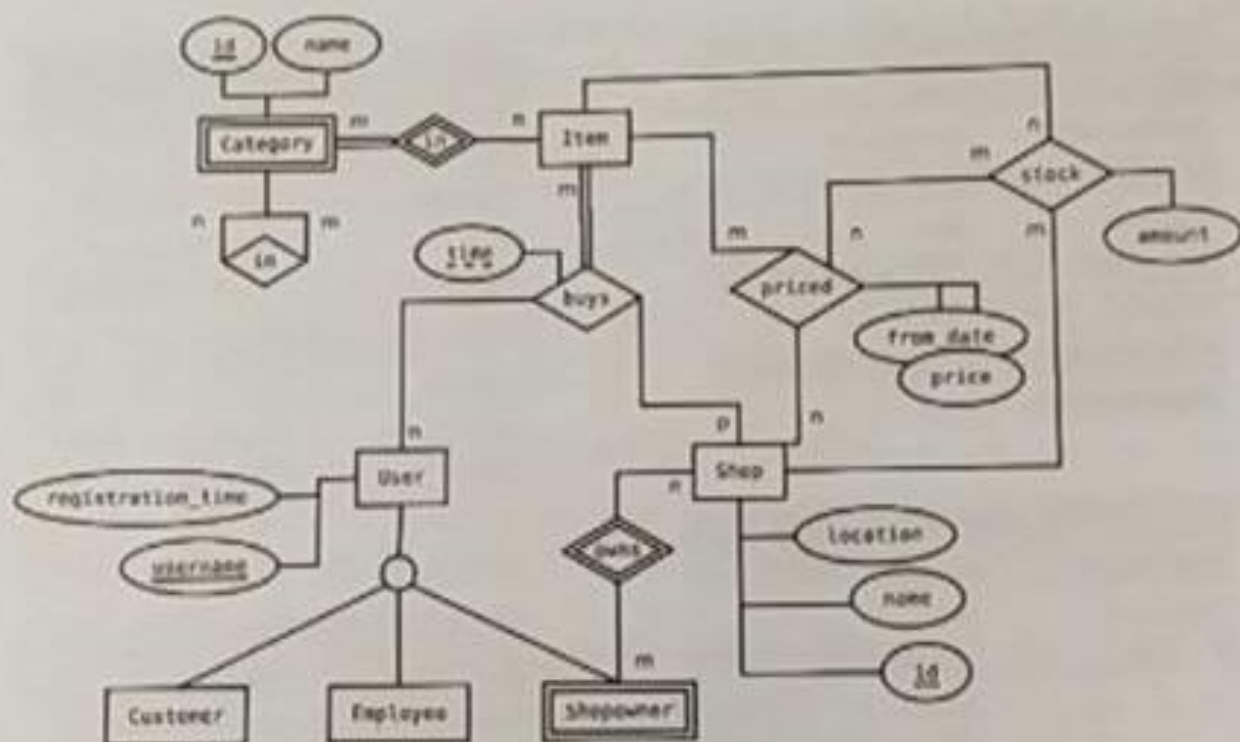
Geef alle interventies terug die in 2016 zijn gebeurd. Ze moeten afgehandeld zijn door een technicus (mechanic) die op het moment van de interventie (intervention) al wat ervaring heeft gehad met dat type interventie. Dit wordt gedefinieerd door de regel dat hij in de maand voor de interventie meer dan vijf interventies van hetzelfde type heeft uitgevoerd. Bijvoorbeeld: interventie '10' van het type onderhoud (maintenance) hoort in het resultaat thuis omdat Frank interventie '10' heeft afgehandeld op 10 december en omdat hij tussen 10 november en 10 december zes onderhoudsinterventies heeft gedaan.



Figuur 1.1: Schema Garage

2 EER

Interpreteer het EER diagram in Figuur 2.1 en haal er de fouten uit. Als er iets open staat voor interpretatie mag je dat vermelden. Als er iets is dat je anders zou doen maar dat niet fout is, moet je dat niet vermelden.



Figuur 2.1: EER Shop - zoek de fout(en)

3 Normalisatie

3.1 Tekstuele normalisatie

Een Belgische bank zou graag een databank hebben met daarin gegevens over de klanten van de bank, hun accounts, hun rekeningnummers en de bedragen die zij ontvangen en betalen.

Hieronder staat een beschrijving van de gegevens die in die databank moeten komen. Zet deze beschrijving om naar een **functioneel afhankelijkheidsdiagram** (zoals in de oefeningenlessen). In de tekst staan een aantal woorden in het vet. Deze woorden zijn de namen van de attributen waarvan u gebruik moet maken. In principe hoeft u **alleen deze attributen** op te nemen in uw antwoord, maar indien u dit noodzakelijk acht, mag u een zeer beperkt aantal attributen toevoegen.

Verantwoording van uw handelingen is niet nodig, tenzij waar/wanneer u dit zelf noodzakelijk acht.

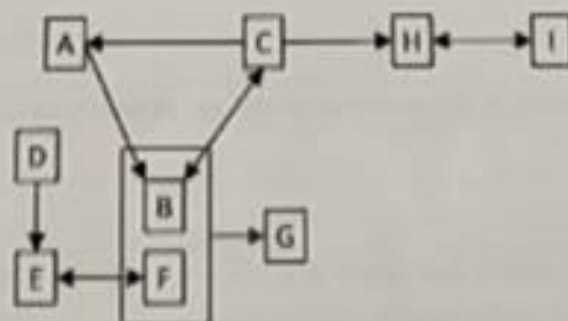
De bank in kwestie heeft alleen maar klanten met Belgische nationaliteit en biedt deze klanten alleen maar Belgische bankrekeningnummers aan. Elke klant van de bank krijgt van de bank precies één account, die overeenkomt met precies één **accountnummer**. Dit accountnummer moet worden bewaard in de databank. Accounts kunnen niet worden gedeeld met of overgedragen aan andere personen. Voor elke account moet ook de volledige **naam** van de corresponderende klant worden bewaard. Uiteraard kunnen er meerdere verschillende klanten zijn die precies dezelfde naam hebben.

Aan elke account wordt minstens één rekening verbonden (meer dan één kan ook). Uiteraard kan éénzelfde rekening slechts aan één account verbonden zijn. Elke rekening wordt gekenmerkt door een uniek **IBAN-nummer**.

De bank wil ook dat alle verrichtingen die op een rekening gebeuren, worden opgeslagen in de databank. Daarbij moet ook de **datum** en het **tijdstip** waarop elke verrichting gebeurt, worden benoemd. Een verrichting kan een **inkomst** zijn: een bedrag dat op een rekening gestort wordt. Een verrichting kan ook een **uitgave** zijn: een bedrag dat van een rekening afgaat. Er wordt hierbij verondersteld dat op éénzelfde dag, op éénzelfde tijdstip slechts één verrichting kan worden uitgevoerd op éénzelfde rekening.

3.2 Abstracte normalisatie

Hieronder staat een abstract functioneel afhankelijkheidsdiagram, dat precies één relatie visualiseert (Fig. 3.1). **Normaliseer** de relatie die erdoor wordt weergegeven **stapsgewijs** tot **BCNF** (zoals in de oefeningenlessen). Doorloop alle fases, gebruik voldoende tussenstappen en **verklaar** waar nodig waarom u ze wel of niet uitvoert. Wanneer **inbreuken** op een normaalvorm bestaan, geef dan elk van deze inbreuken duidelijk weer. **Motiveer** de conclusies die u bereikt. Geef voor elke (bekomen) relatie duidelijk **alle kandidaatsleutels** aan. Geef duidelijk het eindresultaat aan.



Figuur 3.1: Abstract functioneel afhankelijkheidsdiagram

4 Indexering

Welke indexen zou je aanmaken opdat onderstaande SQL query efficiënter zou kunnen uitgevoerd worden? Je mag er van uitgaan dat de gebruikte relaties heel zelden worden aangepast. Geef voor elke index aan waarom je deze aanmaakt, op welke attributen hij gebouwd is en van welke soort hij is.

```
select * from person p1 inner join person p2 on p1.parentid = p2.  
personid where p1.length < 140 and p2.length > 140;
```

5 Query optimalisatie

Beschouw het query plan in figuur 5.1. Daarbij hoort de volgende query die we uit jullie antwoorden in de SQL Exerciser plukten:

```
select mo.modelid, mo.brand, mo.modelname, mo.year, eo.type as  
enginetype from model mo inner join engine eo using (modelid) where  
eo.power >= all(select ei.power from model mi inner join engine ei  
using (modelid) where (mi.modelid = mo.modelid));
```

QUERY PLAN
Hash Join (cost=2.10..1100.63 rows=135 width=31) (actual time=0.113..5.275 rows=49 loops=1)
Hash Cond: (eo.modelid = mo.modelid)
Join Filter: (SubPlan 1)
Rows Removed by Join Filter: 221
-> Seq Scan on engine eo (cost=0.00..5.70 rows=270 width=20) (actual time=0.006..0.027 rows=270 loops=1)
-> Hash (cost=1.49..1.49 rows=49 width=19) (actual time=0.020..0.020 rows=49 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 11kB
-> Seq Scan on model mo (cost=0.00..1.49 rows=49 width=19) (actual time=0.003..0.007 rows=49 loops=1)
SubPlan 1
-> Nested Loop (cost=0.00..8.05 rows=6 width=4) (actual time=0.015..0.018 rows=3 loops=270)
-> Seq Scan on model mi (cost=0.00..1.61 rows=1 width=4) (actual time=0.003..0.004 rows=1 loops=270)
Filter: (modelid = mo.modelid)
Rows Removed by Filter: 29
-> Seq Scan on engine ei (cost=0.00..6.38 rows=6 width=8) (actual time=0.011..0.014 rows=3 loops=270)
Filter: (modelid = mo.modelid)
Rows Removed by Filter: 154
Planning time: 0.239 ms
Execution time: 5.326 ms

Figuur 5.1: Queryplan

Geef aan de hand van het query plan de 2 traagste elementen aan en tracht een verklaring te vinden aan de hand van de query. Beschrijf kort in je eigen woorden hoe je de uitvoeringstijd van deze query zou kunnen verkorten.

6 NoSQL



Figuur 6.1: EER schema.

Figuur 6.1 toont een EER schema dat informatie over vakken, studenten en leraren modelleert. Zet dit schema eerst om naar een relationeel databankmodel. Omschrijf daarna de instructies die je moet gebruiken om de informatie in tabel 6.1 toe te voegen aan:

1. een PostgreSQL databank (ga ervan uit dat de tabellen bestaan maar geen data bevatten).
2. een Mongo databank.

Je hoeft hierbij de statements niet in de correcte syntax te geven, de logica volstaat.

Naam student	Robin Van den Bossche
Vak	Informatiebeheer (Guy De Tré)
Vak	Databanken (Guy De Tré)

Tabel 6.1: Toe te voegen data.