

# 2016 januari examen

## De Mazière

### THEORIE

1) twee tabellen + query geef output en kan het beter

2) Mondeling: Wat is een transactie en wat is het nu, wat zijn de gevaren en oplossingen hiertegen + wat is commit + rollback.

3) Hoe kan je een RDB toch forceren om over te erven + welke van de 4 DML ... kan je gebruiken om in 1 commando aan datamanipulatie te doen.

### OEFENINGEN

## Bertels

1. [10 min]-----

Geef een alternatieve equivalente query zonder het gebruik van CUBE

```
SELECT  row_number() over () as volgnr  
        geslacht, plaats, count(*)  
FROM    spelers  
GROUP BY geslacht, plaats WITH CUBE  
ORDER BY geslacht, plaats
```

Aandachtspunt: let op met row\_number(), denk verder.

Equivalente queries halen dezelfde gegevens op een een gegevensbank.

2. [15 min]-----

Hoe garandeer je integriteit van een 1-op-1- of -0-relatie tussen twee tabellen? Illustreer met een concreet

voorbeeld in schemavorm en geef de bijhorende sql-code. De twee tabellen moeten ook voldoen aan de basisregels bij het ontwerpen van een tabel. Leg nu uit waarom jouw code de 1-op-1- of -0-relatie afdwingt.

3. [20 min]-----

We beschouwen de volgende context:

De kinderen liggen op grote slaapzalen. Elke zaal heeft een nummer, een capaciteit en een bezetting (aantal bedden). Bij elke zaal horen een aantal verzorgsters. Elke patiënt moet een aantal geneesmiddelen innemen per dag

ZAAL

/ \  
/ \

VERZORGSTER PATIËNT

|

GENEESMIDDEL

zaal :        zaalnr [PK], cap, bez

verzorgster :   persnr [PK], vnaam, vadres

patiënt :        pnr [PK], pnaam, padres

geneesmiddel :   gnaam [PK], hoev

Verkorte notatie:

GU   ZAAL

      PATIENT

      GENEESMIDDEL (GNAAM = VALIUM)

GU   ZAAL (ZAALNR = key1zaalkey)

PERFORM UNTIL STATUS1NOTOK

      GNP   VERZORGSTER

      PERFORM UNTIL STATUS2NOTOK

```

        DISPLAY VERZORGSTER
        GNP VERZORGSTER
    END-PERFORM
GN  ZAAL (ZAALNR <> key1zaalkey)
    PATIENT
    GENEESMIDDEL (GNAAM = VALIUM)
    IF DBSTATUS1OK
        GN  ZAAL (ZAALNR = key1zaalkey)
    END-IF
END-PERFORM

```

Wat betekenen GN en GNP? Wat is het verschil tussen beide instructies?

Waarvoor staat PCB en wat is de tegenhanger in een RDBMS?

Wat doet dit programma?

4. [10 min]-----

Leg volledig uit: ACID

Stelling: "Dit is een zwakte bij de meeste RDBMS'en, NoSQL-databanken zijn hier beter in."

Verdedig kort of verwerp deze stelling.

OEFENINGEN RDBMS. [60 min]-----

Enkel SQL (ISO), geen dialecten.

U gebruikt hiervoor de databank Serverroom. Schrijf de query's net uit op papier.

5.

Geef een lijst met servers (serial, jaar, os) waarvoor er geen interventies gebeurd zijn.

6.

Geef de naam van het personeelslid dat het hoogste aantal interventie-eenheden heeft gepresteerd aan gameservers. Gebruik hiervoor geen LIMIT of TOP.

7.

Geef de naam van het OS waarvoor gemiddeld gezien de meeste interventie-eenheden zijn gepesteed.  
Gemiddeld gezien wil zeggen: het totaal aantal interventies bekijken in verhouding tot het totaal aantal servers van dit OS. Gebruik geen LIMIT of TOP.

8.

De klant met het minst aantal interventies in december 2010 krijgt een vast kortingspercentage van 15%. Maak gebruik van een transactie.

9.

Geef voor elke klant een cumulatief overzicht van zijn betaalde facturen gesorteerd op datum (met oplopend rijnummer volgens het klantnummer en de datum van de factuur). Elk woord telt. Een voorbeelduitvoer ter aanvulling:

Rijnr.	ID klant	Naam	Factuurnr.	Cumulatief bedrag
1	1	JD	23	100
2	1	JD	24	150
...				
56	7	MN	21	23
...				

Hierbij heeft klant 1 een eerste factuur van 100 euro en een tweede van 50 euro betaald. Eerste en tweede duiden hier op de datum van de factuur voor die klant. Rij 56 is de eerste boete van klant 7.

-----

Een appel per dag houdt de dokters weg, vooral als je goed mikt.  
- W.C.

---

## # SERVERROOM

### ## Korte beschrijving:

Een serverroom heeft verschillende servers voor verschillende klanten. Een server heeft maar één besturingssysteem en is niet gevirtualiseerd. Sommige servers zijn gameservers en hebben een hogere prioriteit wat betreft nodige interventies.

Voor iedere interventie wordt het aantal gewerkte tijdseenheden genoteerd. Maandelijks wordt er een factuur gemaakt per klant die interventies gehad heeft. Een standaardbedrag wordt dan vermenigvuldigd met het aantal te factureren eenheden. Als een klant een korting geniet, wordt dit percentage in mindering gebracht op de factuur.

Alle mogelijke applicaties staan beschreven. Het gaat hier om binaire, wat wil zeggen dat als een applicatie bestaat voor verschillende besturingssystemen, het ook meerdere keren voorkomt in de applicatietabel.

Installaties van applicaties op bepaalde servers worden ook geregistreerd om zo een algemeen beeld te hebben van de staat van een server.

### ## Tabel- en veldbeschrijving

#### Server

serial	int	not null	uniek serienummer voor de server [PK]
jaar	int	not null	jaar van ingebruikname
os	char(50)	not null	naam van het besturingssysteem
klantid	int		huidige klant waarvoor deze server draait
isgameserver	boolean	not null	gameservers hebben een hogere prioriteit

#### Klant

id	int	not null	uniek ID voor een klant [PK]
naam	int	not null	naam van de klant

tarief	double	not null	tarief per eenheid voor deze klant
korting	double		kortingspercentage voor speciale klanten

#### Interventie

serial	int	not null	[PK]
personeelslid	int	not null	
datum	datetime	not null	
opmerkingen	char(255)	not null	
eenheden	int	not null	
factuurnr	int	not null	

#### Factuur

factuurnr	int	not null	
klantid	int	not null	
datum	date	not null	
bedrag	double	not null	
isbetaald	boolean	not null	

#### Applicatie

id	int	not null	uniek ID voor een applicatie
naam	char(50)	not null	verschillende versies van een app hebben dezelfde naam (ook voor verschillende OS)
versie	char(10)		
os	char(50)	not null	

#### Installatie

applicatieid	int	not null	[PK]
serial	int	not null	[PK]
datum	datetime	not null	
personeelslid	int	not null	personeelslid dat de installatie verrichte

#### Personeelslid

id	int	not null	[PK]
naam	char(60)	not null	naam van het personeelslid

## ERD

-----

-----

```

| server          |          | klant          |          | -----
|-----|          |-----|          | factuur        | | | |
| serial INT [PK] |          | id INT [PK]    |          |-----|
| jaar INT        | >|-----|| | naam INT        |          | factuurnr INT [PK] |
| os CHAR(50)     |          | tarief DOUBLE  | ||---|< | klantnr INT      |
| klantid INT     | ||--          | korting DOUBLE |          | datum DATE        |
| isgameserver BOOL |          |-----          | bedrag DOUBLE     |
|-----          |          | isbetaald BOOL   |
=          |          |-----          |-----
|          | | interventie      |          | =
|          | |-----|          |
|          | | serial INT [PK]  |          |
|          | | personeelslid INT | >|-----
|          |--|< | datum DATETIME    |
|          | opmerkingen CHAR(255) |-----
|          | eenheden INT        | >|-- | personeelslid |
-          | factuurnr INT        | | |-----|
^          |-----          |--|| | id INT [PK]    |
|-----          | naam CHAR(60) |-----
| installatie     |          |----- | applicatie    | |
|-----|          |          | =          |-----|
| applicatieid INT [PK] |          |          | id INT [PK]    |
| serial INT        | >|-----          | naam CHAR(50) |
| datum DATETIME    |          | versie CHAR(10) |
| personeelslid INT | >|-----|| | os CHAR(50)    |
|-----          |-----

```

# 13 Januari 2016

## De Mazière

### THEORIE

1) Mondeling: geef het verschil tussen gecorreleerde en niet-gecorreleerde subqueries. Illustreer dit aan de hand van een voorbeeld. Geef ook de voor- en nadelen van beide queries (performantie-gewijs ...)

a) Bijvraag: Wat zijn de 4 soorten subqueries?

b) Hoe kan je de performantie van een subquery testen?

2) Wat zijn de verschillen tussen hierarchisch en netwerk model?

3) Hoe worden relaties gemaakt in een NoSQL database? wat is het verschil met een rdbms hierin? geef ook een voorbeeld. Wat heeft dit voor gevolgen?

## OEFFENINGEN

1) Je kreeg een query:

```
SELECT t1.nummer, t1.kolom1, (SELECT count(*) from tabel2 WHERE tabel2.nummer =  
t1.nummer)
```

```
FROM tabel1 as t1, tabel2 as t2
```

```
WHERE t1.nummer < 3
```

Tabel1

nummer	kolom1
1	1
2	2
3	3
4	4
5	5
6	6

Tabel2

nummer	kolom1
1	10
2	20
3	30
4	40

a) Geef de output:

b) Kan de query optimaler? Zoja, maak ze.

2) Gegeven een blad met een aantal tabellen over bakkers en broden.



- a) Toon alle bakkers met maximaal 4 generaties in de familie die bakker zijn. (Kan je doen met `with_recursive` of alle tabellen joinen met allemaal `left outer joins` of 4 aparte queries voor 4-3-2-1 generaties en deze met `union` samensmelten)
- b) Toon alle bakkers van "Janssens Jan" langs vaders kant. Output moet `[regnr persoon] - [rregnr vader]` zijn. (Dit moest met `with_recursive` gedaan worden).
- c) Toon voor de bakker die het meest aantal broden gebakken heeft in januari 2015, de klantnaam met de kleinste bestelling (op basis van de prijs) aan wie de bakker ooit heeft geleverd.
- d) Geef een stored procedure die prijs van de broden aanpaste.
- e) Sorteer de bakkers aflopend op basis van het aantal klanten in 2016. Vermeld per bakker het aantal broden gebakken voor dat jaar en de gemiddelde prijs per bestelling.
- 

Revision #2

Created 31 October 2021 21:33:07 by Jasper G.

Updated 3 December 2021 22:13:18 by Jasper G.