

# Databanken 1

- [2015 formularium Niels & Simon](#)
- [2015 januari examen](#)
- [2015 samenvatting Niels en Simon](#)
- [2016 januari examen](#)
- [2017 januari examen](#)
- [2018 augustus examen](#)
- [2020 augustus examen](#)
- [2020 januari examen](#)
- [2021 januari examen](#)
- [algemeen](#)

# 2015 formularium Niels & Simon

Met dank aan Niels Vermeiren & Simon Cek: [Formularium 2015 DB1 Niels Vermeiren & Simon Cek.pdf](#)

# 2015 januari examen

## 20/01

**THEORIE** 1) bespreek 1 op 1 of 0 relaties, hoe je de referentiële integriteit kan bewaren, geef een voorbeeld met een tabel en schrijf in SQL code

2) geef 4 duidelijk verschillende manieren van gegevensbeveiliging en hun SQL code

3) welke van de 2 is de meest performante code (2 codes gegeven met hun explain eronder)

4) netwerkdatabase vraag over RETENTION

5) vergeten

**OEFENINGEN** database over een bank gegeven

1) gegeven query = `select * from` 8 verschillende kolommen. Wat wordt er geselecteerd?

2) Verander alle rekeningen met naam Berchtem naar Bertels en zorg dan dat met 1 instructie (niet dezelfde) alles terug ongedaan kan worden

3) geef alle info over rekeningen die afgelopen 2 dagen zijn aangepast (laatste is laatste 2 dagen bewerkingen niet vanaf vandaag hier bestaat een tabel over (maar niet gegeven) WTF)

4) geef alle vrouwelijke eigenaars van een rekening die begint met een cijfer, een % teken bevat in het midden en eindigt op CH of CF weer

5) vergeten

## 12/01

**THEORIE**

1) Gegeven tabel met per rij (land, goals, hematocrietgemiddelde) :

a) Wat is de uitvoer van `SELECT COUNT(*), MAX(k.hematocrietgemiddelde) FROM wk w cross join wk k GROUP BY w.goals`

b) Bedenk een zinvolle output en geef hiervoor alternatieve query

2) (mondeling) Leg uit: transacties: voordelen/nadelen, hoe werkt het, voorbeeld..

3) Gegeven was code en model van een db (Hiërarchische?), hiervan moest je geven welk soort GBS dit was en van de code moest je de uitvoer geven.

4) ORDBMS: geef de 4 verschillende soorten indexen, indien deze niet bestaan, wat gebruikt men i.p.v. deze indexen

## **OEFENINGEN**

Databank met geiten/wedstrijden/jaarproductie/behandelingen..

1) Gegeven query, klopt dit? Zoja, optimaliseer, zonee, wat klopt er niet?

2) Geef alle geiten welke een behandeling hebben gehad in de afgelopen week en waar hun ras begint met 'melk'

3) Geef de geit(en) dewelke aan alle wedstrijden hebben meegedaan

4) Schrijf procedure om alle geiten die minder dan 10 L (kg) opbrengst per jaar hebben te verwijderen

5) Geef de hele boom van geiten beginnende vanaf geit met nr 1. (Recursieve functie in de stijl van vader uit cursus)

# 2015 samenvatting Niels en Simon

Met dank aan Niels Vermeiren & Simon Cek: [Samenvatting 2015 DB1 Niels Vermeiren & Simon Cek.pdf](#)

# 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.

# 2017 januari examen

## 9 Januari

(Van wat ik mij herinner)

Volledig schriftelijk, 2u tijd (1.5u gerekend)

- 1) In welke SELECT componenten kun je gebruik maken van gecorreleerde subqueries? (10min)
- 2) Code voor het creëren van een tabel met een aantal attributen gegeven. Wat is hier goed/slecht aan (inhoudelijk, fysisch en sql) (10min)
- 3) Code voor een hiërarchische DB programma gegeven. Wat voor DB is dit? Wat geeft dit als output? (10min) (zelfde programma als vorig jaar)

```
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
```

- 4) 3 verschillende queries gegeven die hetzelfde doen (een user teruggeven op basis van één van zijn eigenschappen), adhv 6 j/f vragen bepalen welke de snelste is. (kwam ongeveer op het volgende neer denk ik: ) (10min)

> a) select \*

from users

where id="123546"

1) Deze query is de snelste omdat id een uniek attribuut is.

2) Deze query is niet de snelste omdat het id zeer hoog is,  
dus hij waarschijnlijk vrij ver in de de lijst van id's gaat moeten zoeken om deze user te vinden.

> b) select \*

from users

where hash="ea02e3a20e3ae23265"

1) Deze query is de snelste omdat het een hash is.

2) Deze query is niet de snelste omdat een hash niet per se uniek is (en hier ook niet).

> c) select \*

from users

where joindate="05-12-06" (met de eerste user gejoind op "05-12-06")

1) Deze query is de snelste omdat deze eerst in de lijst staat (eerste user dan gejoined).

2) Deze query is niet de snelste omdat timestamp een complex attribuut is.

5) Query met rollup gegeven. Bespreek hoe de uitvoer hiervan eruit gaat zien. (10min)

6) Vergeten (10min)

7) 4 queries maken (met subqueries, aggregatiefuncties en joins) (40min)



# 2018 augustus examen

THEORIE 10/20

1) Geef alternatief equivalent query zonder gebruik van CUBE

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

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

2) Waarvoor gebruikt men FILTER, maak duidelijk aan de hand van een kleine SQL

3) Hoe garandeer je integriteit van een 1-1 of 1-0 relatie tussen 2 tabellen? Illustreer en onderlijn het noodzakelijke volgens jou.

4) Wat zijn de verschillen tussen een hiërarchisch model en relationeel model

5) Geef de uitvoer van volgende 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

SQL 10/20

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

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

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

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

## 1. SERVERROOM

### 1. 1. 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.

### 1. 1. 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
```

### 1. 1. ERD

```
-----
| server      |
|-----|
| serial INT [PK] |
| jaar INT    | >|-----|
| os CHAR(50) |
| klantid INT | ||--
| isgameserver BOOL |
-----
|
=
|
| interventie
|
| serial INT [PK]
|
| personeelslid INT | >|-----|
|
--|<| datum DATETIME
|
| opmerkingen CHAR(255) |
|
| eenheden INT | >|-- | personeelslid |
-
| factuurnr INT | | |-----|
^
-----
--|| | id INT [PK] |
-----
| naam CHAR(60) |
-----
| installatie |
|-----|
=
|-----|
| applicatieid INT [PK] |
|
| id INT [PK] |
|
| serial INT | >|-----|
| naam CHAR(50) |
```

datum DATETIME		versie CHAR(10)	
personeelslid INT	> -----	os CHAR(50)	
-----		-----	

# 2020 augustus examen

/10 theorie

1) /2 schrijf deze query op 2 versch manieren `select * from x tabel_a where x not in (select x from tabel_b )`

2) /2 tabel van isolation level (not possible, possible) letterlijk invullen

3)programma over verzorgsters in hierarchisch tabel wat is GN en GNP wat doet dit programma

4) /2 een query van view create view as `select snr from spelers where blabla group by s.snr with check option`

kun je hierin muteren (insert, delete)? ja/nee/soms/optioneel

(ik had nee omdat er zo paar regels zijn wanneer je niet mag muteren en 1 daarvan is geen group by / order by of set operatoren)

5) /2 Waar kan er in een select instructie een niet gecorreleerde subquery staan, geef de naam van ja wa vervangt me die subquery. Doe dit voor amle componenten, denk aan select from where (vage vraag again...)

/10 praktijk

5 queries db over geiten

zet de 5 queries in 1 grote transactie waar concurrency hoog moet zijn (dus dirty read)

start transaction

set transaction isolation level dirty read

- query\*

end transaction

uhm er zat ook een query bij waar je cube moest gebruiken , ook like %

groetjes thuis kirito-kun x

# 2020 januari examen

De eerste paar vragen stonden op 2 punten en je krijgt 2h voor het examen.

- 1) Query met GROUP BY CUBE -> wordt gevraagd om anders te schrijven (GROUP BY CUBE -> GROUP BY ROLLUP)
- 2) Leg het begrip "Refererende Integriteit" + geef een voorbeeld van tabellen enz
- 3) Leg de begrippen (ROOT, GHU, RECORD, GORDAK) uit en zeg bij welke soort databank deze horen (netwerk of hiërarchisch)
- 4) In welke delen van de SELECT query kan je een gecorrelated subquery uitvoeren?
- 5) Gegeven: query + omschrijving -> Klopt deze query? + leg de keuze uit
- 6) Pagina vol met CREATE TABLE's => 4 query's schrijven + er werd ook gevraagd om transactions te gebruiken (eentje was een CREATE VIEW)



# 2021 januari examen

## THEORIE:

- 1: de uitvoer van 2 tabellen die gejoined worden on true.
- 2: vraag over hierarchische databank structuur. bij welke database structuur hoort deze code? wat doet de code?
- 3: welke 4 soorten subqueries zijn er? bij welker kan je exists toepassen?
- 4: vraag over collation.
- 5: verander de geit tabel zodat er in bijgehouden wordt wie de eigenaar is van een geit.

## PRAKTIJK:

- 1: geef hoeveel wedstrijden er per plaats zijn, maar enkel plaatsen waarbij de meest recente wedstrijd dit jaar was.
- 2: geef alle geiten die geen unieke geboortedatum hebben.
- 3: geef per geit de totale kg\_melk dat die geproduceert heeft, en ook het totaal onafhankelijk van de geit (dus rollup denk ik).
- 4: maak een object eigen\_geit aan zodat een boer hier aan kan en enkel zijn eigen geiten kan zien en aanpassen.

# algemeen

## W. Bertels / P. De Mazière

Lessen zijn niet verplicht, maar er worden aanwezigheden opgenomen. Als je maar 4u of minder mist, mag je extra bladen meenemen naar het examen. Zo ook voor oefeningen maken (of extra opdrachten). Maximaal 6blzn. Leerstof: SQL (subqueries, joins, vensterfuncties, indexen), Hiërarchische en Netwerk DB