	Hogeschool PXL Departement IT Academiejaar 2018-2019 Examen Big Data – DEEL 1
Vak	Big Data
Resultaat	DEEL 1 /10p + DEEL 2 /10 p = /20
Periode	1 ^{ste} zit 2019
Datum	20 juni 2019
Tijdstip	08:30 uur
Klassen	1 TIN – 1TIW
Lector	Carina Medats – Kerstin Nys – Isabelle Godfrind

Studentengegevens	
Naam student	
Voornaam student	
Klas	
Lector	

Samenstelling bundel		
Onderdelen (*)	Deel 1	Deel 2
Inhoud	Schriftelijk (theorie en MongoDB)	Laptop
Pagina's	8 p.	3 p.
Puntenverdeling	(10/20)	(10/20)
Digitaal beginbestand		
Digitale indiening		X
Toegelaten hulpmiddelen:	NIETS	
* rekenmachine		
* laptop		X
* internet		
* cursusmateriaal		X
Opmerkingen:	Deel 1 duurt maximum 1.5 uur. Achterzijde kan als kladblad worden gebruikt. Het formularium (laatste blad) mag worden losgemaakt.	Studenten mogen gebruik maken van: - alle materiaal dat op de laptop aanwezig is - 1 cursus DATA - SQL
Aanvangsuur examen – deel 1: 8u30		Einde examen – deel 1: 10u



**DE HOGESCHOOL
MET HET NETWERK**

Checklist voor het afleggen van examen

- ☐ GSM's/smartphones liggen uitgeschakeld op tafel
- ☐ Studentenkaart ter beschikking houden
- ☐ Jassen en tassen vooraan in het lokaal
- ☐ GEEN GSM's of smartphones in jassen of tassen
- ☐ Bij laptopexamen strikt de opgelegde regels volgen
- ☐ Examenbundels blijven steeds samengeniet
- ☐ Toegestaan: 1 droog koekje en 1 drankje in hersluitbaar flesje
- ☐ Iedereen zwijgt tijdens het examen

-
- ☐ Afgeven mag niet voor 9:00/14:00 uur

- ☐ Niet vergeten op je examenkopij te vermelden:

Naam van de lector

Eigen naam

Klas

MD5Hash



Theorie Big Data (2 punten)

Geen voorbeeld ter beschikking

MongoDB (8 punten) - voor dit gedeelte mag je enkel het bijgevoegde formularium gebruiken en geen laptop

Veronderstel dat een databank 'sport' bestaat met daarin een collection 'cyclists'.

De databank 'sport' is de actieve databank.

De collection 'cyclists' bevat een groot aantal documenten, die er “kunnen” uitzien zoals volgende voorbeelden:

```
{ "_id" : "ZS1", "firstname" : "Zdenek", "lastname" : "Stybar", "nickname" : "Styby", "birthday" :  
ISODate("1985-12-11T00:00:00Z"), "birthplace" : { "city" : "Plana", "country" : "Tsjecho-Slowakije"  
}, "nationality" : "Tsjechische", "length" : 183, "languages" : [ "Dutch", "English", "Tsjech" ],  
"victories" : [ { "race" : "Strade Bianche", "year" : 2015 }, { "race" : "Omloop Het Nieuwsblad", "year"  
: 2019 }, { "race" : "E3 Harelbeke", "year" : 2019 } ], "UCI_ranking" : 1533, "cyclocross" : true }
```

```
{ "_id" : "GV7", "firstname" : "Greg", "lastname" : "Van Avermaet", "nickname" : "Golden Greg",  
"birthday" : ISODate("1985-05-17T00:00:00Z"), "birthplace" : { "city" : "Lokeren", "country" :  
"Belgie" }, "team" : "CCC Team", "length" : 181, "victories" : [ { "race" : "Parijs-Tours", "year" : 2011  
}, { "race" : "Oml. Het Nieuwsblad", "year" : [ 2016, 2017 ] }, { "race" : "E3 Harelbeke", "year" : 2017  
} ], "UCI_ranking" : 2760 }
```

```
{ "_id" : "PS4", "firstname" : "Peter", "lastname" : "Sagan", "UCI_ranking" : 2408 }
```

Opgave 1 (1p)

Toon de wielrenners die ooit de wedstrijd 'omloop Het Nieuwsblad' gewonnen hebben. Merk op dat de schrijfwijze niet overal dezelfde is.

Toon van deze renners het id en de familienaam.

Opgave 2 (1p)

Toon wielrenners die zowel Dutch als Tsjech praten en aan cyclocross doen.

Toon alle gegevens, behalve hun victories.

Opgave 3 (1,5p)

Toon alle wielrenners die in het CCC Team zitten en al overwinningen behaalden + alle wielrenners die aan cyclocross doen.

Opgave 4 (1,5p)

Toon van alle wielrenners die een UCI-Ranking hebben tussen 2000 en 2500 (inclusief) de 5^{de} en 6^{de} hoogst genoteerde.

Opgave 5 (1p)

Schrijf code die via 1 instructie de volgende 2 documenten toevoegt (met dezelfde structuur als de voorbeelddocumenten!!):

1. Object-id: LB3
Lars Bak, geboren op 16 januari 1981
2. Object-id: laat het door MongoDB genereren
Magnus Cort, geboren te Bornholm, Denemarken

Opgave 6 (2p)

Schrijf code die een weight van 68 toevoegt bij alle wielrenners met tsjechische nationaliteit en een lengte van 181 of 185.

Formularium MongoDB

```
use test
show dbs
db
db.dropDatabase()
db.CreateCollection(Name, options)
show collections
db.collectionnaam.drop()
db.collectionnaam.insert()
db.collectionnaam.update({SELECTION_CRITERIA}, {$set:{field1:value1, ...}})

db.collectionnaam.update(..., {multi:true})

db.collectionnaam.updateMany({SELECTION_CRITERIA}, {$set:{field1:value1, ...}})

db.collectionnaam.update({SELECTION_CRITERIA}, {$push:{field1:value1}})

db.collectionnaam.save({_id:ObjectId(),NEW_DATA})
db.collectionnaam.remove()
db.collectionnaam.remove(DELETION_CRITERIA)

db.collectionnaam.find()
db.collectionnaam.find.pretty()
db.collectionnaam.findOne()
db.collectionnaam.find().sort({key:1})
db.collectionnaam.find().skip(n)
db.collectionnaam.find().count()
db.collectionnaam.find().limit(n)
db.collectionnaam.count()
```

```
db.collectionnaam.find({SELECTIE_CRITERIA},{PROJECTIE})
```

SELECTIE_CRITERIA:


```
field:'tekst'
field: n
field:{$lt:value}
$lte
$gt
$gte
$ne
$in:[value1, value2, ...]
$nin:[value1, value2, ...]
$exists
/value/
  ^ vooraan
  $ achteraan
  i
'doc_field.subdoc_field': ...
field:['value1','value2']
field:{$all:['value1','value2']}
```

'field.n': 'value'

switch naar DB test
toon alle DB
toon huidige DB
drop huidige DB
creatie collectie
toon alle collecties in de huidige DB
drop collectienaam
voegt een doc toe aan een collectie
update het eerste document dat voldoet aan
selectiecriteria van de collectie
update alle documenten die voldoen aan
selectiecriteria van de collectie
update alle documenten die voldoen aan
selectiecriteria van de collectie
voegt exact 1 value toe aan een array of 1 subdoc bij
een array van subdocumenten ; array MOET BESTAAN
vervangt doc met _id door nieuwe data
verwijdert alle documenten van de collectie
verwijdert alle documenten die voldoen aan
selectiecriteria

geeft alle info weer vd collectie, alles achter elkaar
geeft alle info weer vd collectie mooi met tabs
geeft alle info weer van één document (het eerste)
sorteert oplopend op key (-1 voor dalende volgorde)
geeft alle info na het overslaan van n documenten
telt het aantal items
geeft de eerste n documenten weer
geeft het aantal documenten weer

veld gelijk aan tekst
veld gelijk aan getal n
veld kleiner dan value
kleiner dan of gelijk aan
groter dan
groter dan of gelijk aan
niet gelijk aan
gelijk aan value1 of gelijk aan value2, ...
niet gelijk aan value1 en niet gelijk aan value2, ...
is aanwezig
value komt voor
moet beginnen met
moet eindigen met
case insensitive
beperking op veld van subdocument
veld dat exact de 2 values heeft
veld dat de 2 values heeft, ongeacht volgorde
of andere elementen
op positie n van het veld staat value

	Hogeschool PXL Departement IT Academiejaar 2018-2019 Examen Big Data – DEEL 2
Vak	Big Data
Resultaat	DEEL 1 /10p + DEEL 2 /10 p = /20
Periode	1 ^{ste} zit 2019
Datum	20 juni 2019
Tijdstip	08:30 uur
Klassen	1 TIN – 1TIW
Lector	Carina Medats – Kerstin Nys – Isabelle Godfrind

Studentengegevens	
Naam student	
Voornaam student	
Klas	
Lector	

Samenstelling bundel		
Onderdelen (*)	Deel 1	Deel 2
Inhoud	Schriftelijk (theorie en MongoDB)	Laptop
Pagina's	8 p.	3 p.
Puntenverdeling	(10/20)	(10/20)
Digitaal beginbestand		
Digitale indiening		X
Toege laten hulpmiddelen:	NIETS	
* rekenmachine		
* laptop		X
* internet		
* cursusmateriaal		X
Opmerkingen:	Deel 1 duurt maximum 1.5 uur. Achterzijde kan als kladblad worden gebruikt. Het formularium (laatste blad) mag worden losgemaakt.	Studenten mogen gebruik maken van: - alle materiaal dat op de laptop aanwezig is - 1 cursus DATA - SQL

Aanvangsuur examen – deel 1: 8u30

Einde examen – deel 2: 12u

Overzicht databank

PERFORMANCES

theatre_performance
season
producer
description
price_ticket

PERFORMANCE_ACTORS

theatre_performance
season
actor_id

ACTORS

actor_id
actor_name
actor_firstname
known_from

PERFORMANCE_DATES

theatre_performance
date_time
season

RESERVATIONS

reservation_id
subscriber_id
comments

SUBSCRIBERS

subscriber_id
subscriber_name
subscriber_firstname
address
zip_code
telephone
email

RESERVATION_SPECTATORS

reservation_id
spectator_id

REHEARSALS

theatre_performance
date_starttime
date_endtime
location_id

LOCATIONS

location_id
address
zip_code

ZIP_CODES

zip_code
city

REHEARSAL_ACTORS

theatre_performance
date_starttime
actor_id

RESERVATION_PERFORMANCES

reservation_id
theatre_performance
date_time

SPECTATORS

spectator_id
spectator_name
spectator_firstname

PL/SQL: Algemene Richtlijnen:

- Maak een map aan met als naam 'naam_voornaam' (zonder spaties),
vb Els Van den Broecke maakt een map met als naam VandenBroecke_Els
- Aan het einde van het examen maak je van de map een zip-file, die je via FileZilla uploadt.
- **Belangrijke opmerking:** Elke oplossing moet altijd werken ongeacht de inhoud kleine of hoofdletters of een mix bevat en ongeacht het datumformaat.

Opgave (10p)

Maak een procedure die een chronologisch overzicht geeft van alle repetities van een bepaalde maand in een bepaald jaar.

Volgende gegevens worden getoond: de naam van het optreden, de datum en het uur van de repetitie, alsook de namen van de acteurs die aanwezig moeten zijn op die repetitie.

De invoerparameters:

- Maand als een getal; vb: 12
- Jaar als een getal; vb: 2018

Voorbeeld output (fictieve gegevens!!):

```
De directeur slaat op hol! 01-12-2018 16:10
---> Darcis Jef
---> Croonburg Jacky
---> Geris Erna
De directeur slaat op hol! 05-12-2018 09:00
---> Wilmots Gert
---> Croonburg Jacky
De directeur slaat op hol! 05-12-2018 14:00
---> Geris Erna
---> Darcis Jef
Koninklijk DNA 10-12-2018 09:00
---> Wilmots Gert
---> Bex Gary
De directeur slaat op hol! 11-12-2018 19:00
---> Croonburg Jacky
.....
```

Bewaar deze oplossing als : naam_voornaam.sql in de voordien gemaakte map.