

Introduction to Databases

 tuyaux.winak.be/index.php/Introduction_to_Databases

Introduction to Databases (Databases & XML)

Richting	<u>Informatica</u>
----------	--------------------

Jaar	<u>2BINF</u>
------	--------------

Bespreking

Theorie (2015- ...)

De droge theorielessen en het gebrek aan ervaring met Jon Sneyers maakt dat hier weinig over te zeggen valt. Al de slides die in de lessen gebruikt worden, zijn terug te vinden op BlackBoard. Het praktijkgedeelte is het voornamelijk hetzelfde gebleven.

Theorie (2012-2014)

De theorielessen van dit vak worden gegeven door een van duitstalige professor die les geeft in het Engels. Als je de lessen wil meepikken moet je dus wat extra moeite doen want opletten gaat niet met hetzelfde gemak als je gewoon zou zijn. Op het moment van schrijven was de cursus nog in de maak en waren de slides dus het enige referentiepunt. Het is aan te raden om het boek eens uit te lenen dat de prof aanraadt aangezien quasi alles dat in de les behandeld wordt op dit boek gebaseerd is. Ongeveer alles is even belangrijk maar je zou je niet te hard mogen focussen op de details van één hoofdstuk (zoals bijvoorbeeld Datalog of XPath).

Praktijk

De oefeningenlessen zijn een aanrader om bij te wonen aangezien ze niet zomaar de theorie inoefenen maar er ook verder op ingaan. Zo geven ze ook een goede indruk van wat je voor het examen moet kunnen. Om aan het examen te mogen deelnemen moet je twee keer individueel een oefening voorbereiden en presenteren voor de klas. Op de ochtend voor de presentatie wordt iedereen die de oefening heeft voorbereid geacht deze af te geven. De eigenaar van de beste voorbereiding mag die oefening dan presenteren. Dit is ook het ideale moment om feedback te krijgen; zelfs als je niet presenteert zal de assistent al je vragen beantwoorden. Dit kan handig zijn aangezien niet alles dat in de oefeningen voorkomt in de theorielessen wordt gedekt. Als je moeite hebt met de oefeningen kan je best het boek uitlenen dat de professor aanraadt. Hier staat werkelijk alles in om de oefeningen tot een goed einde te brengen. De oplossingen worden na de les meestal op BlackBoard geplaatst.

Examen

Het examen dekt alle leerstof vrij goed. Je mag zelf kiezen of je het in het Engels of Nederlands aflegt, zo kies je je eigen examiner. Het is ook niet al te moeilijk: als je je oefeningen begreep geraak je al een heel eind verder. Het is dus geen examen waarbij wordt verwacht dat je louter reproduceert. Verwacht je aan een ingewikkelde SQL query, een relationeel data schema en misschien iets met XQuery. Je zou normaal niet al te veel van buiten moeten leren; een aantal dingen worden op het examen gegeven ter informatie, afhankelijk van de vragen. Relationele operaties zoals de verschillende joins vallen hier echter buiten en zal je je eigen moeten maken. Dit wil niet zeggen dat de theorie buiten beschouwing blijft, maar alle theorie die gevraagd wordt heb je gebruikt tijdens de oefeningen en zal dus niet te ingewikkeld of onverwacht zijn.

Puntenverdeling

De theorie en praktijk staan ongeveer op evenveel punten.

Examenvragen

Academiejaar 2019 - 2020 - 2de zittijd

[Media:IDB_Augustus_2020.pdf](#)

Academiejaar 2019 - 2020 - 1ste zittijd

[Media:IDB_Januari_2020.pdf](#)

Academiejaar 2018 - 2019 - 2de zittijd

[Media:IDB_Augustus_2019.pdf](#)

Academiejaar 2017 - 2018 - 1ste zittijd

[Media:Db1718.pdf](#)

Academiejaar 2016 - 2017 - 1ste zittijd

[Media:DBI-exam-Jan17-v2EersteZit.pdf](#)

Academiejaar 2014 - 2015 - 1ste zittijd

[Media:Db1415.pdf](#)

Academiejaar 2013 - 2014 - 1ste zittijd

[Media:Db1314.pdf](#)

Academiejaar 2012 - 2013 - 1ste zittijd

[Media:Db1213.pdf](#)

Theorie

1. Gegeven $R(A,B,C)$ en $F=A \rightarrow B, B \rightarrow C$. Hoeveel functional dependencies volgen uit F ? Verklaar!
2. Formuleer in Nederlands:

```
SELECT DISTINCT Drinker FROM Visits WHERE bar IN (SELECT bar FROM Serves
WHERE beer = 'Duvel') AND bar NOT IN (SELECT bar FROM Serves WHERE beer <>
'Duvel')
```

3. Leg de formele definitie van de renaming op schema- en instanceniveau uit. (slide 101 was gegeven)
4. Schrijf een LISP functie die na gaat of alle tuples voor de relatie REL dezelfde lengte hebben. (gegeven was een functie LENGTH die de lengte van een tuple teruggaf)

Praktijk

1. Gegeven een ERM-diagram, verbeter en verklaar & motiveer je verbeteringen.
2. Gegeven volgend database schema

```
Kind(ID, naam, adres, geslacht)
```

```
Ouder(ID, naam, adres, geslacht)
```

```
KindVan(Kind_ID, Ouder_ID)
```

1. Druk uit in SQL: "Geef alle ouders die geen zoon hebben en minstens 2 dochters"
 2. Druk uit in relationele algebra: "Geef alle kinderen waarvan enkel de moeder in de databank zit"
 3. Druk uit in domein calculus: "Geef alle kinderen waarvan enkel de moeder in de databank zit"
3. Gegeven $R(A,B,C,D,E,F)$ en $F=A \rightarrow BC, AC \rightarrow DF, B \rightarrow CD, E \rightarrow B$, bewijs of geef een tegenvoorbeeld:
 1. $F \vdash A \rightarrow F$
 2. $F \vdash DE \rightarrow BC$
 3. $F \vdash CE \rightarrow DF$
 4. Schrijf een LISP functie die met twee lijsten als input (a_1, \dots, a_n) en (b_1, \dots, b_n) de output $(a_1, b_2, a_3, b_4, \dots, a_{n-1}, b_n)$ geeft met n even. Als beide lijsten van verschillende lengte zijn, moet een error getoond worden.

Theorie

1. Als $R(A,B)$ en $S(B,C)$ twee relaties zijn en je beschouwt de vraag: 'Geef voor elke C-waarde cc het aantal verschillende A-waarden aa zodat er een B-waarde bb bestaat met (a,b) in RR en (b,c) in SS '.

```
SELECT C, COUNT(A)
FROM R, S
WHERE R.B = S.B
GROUP BY C;
```

Welke van de volgende drie verklaringen is juist? Bewijs telkens of geef een tegenvoorbeeld.

1. Bovenstaande query geeft altijd als resultaat het antwoord van bovenstaande vraag.
 2. Bovenstaande query geeft als resultaat het antwoord van bovenstaande vraag als de functionele afhankelijkheid $A \rightarrow B$ of de functionele afhankelijkheid $C \rightarrow B$ geldt.
 3. Als bovenstaande query als resultaat het antwoord van bovenstaande vraag geeft dan geldt de functionele afhankelijkheid $A \rightarrow B$ of de functionele afhankelijkheid $C \rightarrow B$.
-

2. De definitie van F-SORT is

```
(DEFUN F-SORT (RELATION I J)
; quicksort of RELATION on its Ith attribute, if J=1 then ASC else DESC
(COND ((NULL RELATION) NIL)
      (T (LET ((R1 (F-SIGMA RELATION
                          (LIST (LIST I) '< (NTH (CAR RELATION) I ))))
                (R2 (F-SIGMA RELATION
                          (LIST (LIST I) '> (NTH (CAR RELATION) I ))))
                (R3 (F-SIGMA RELATION
                          (LIST (LIST I) '= (NTH (CAR RELATION) I )))))
          (APPEND (F-SORT (COND ((= J 1) R1) (T R2)) I J)
                  R3
                  (F-SORT (COND ((= J 1) R2) (T R1)) I J))))
      )
)
```

Gebruik makend van de functies F-SIGMA en EVALCOND:

```
(DEFUN EVALCOND (TUPLE CND)
; evaluates the infix condition CND on TUPLE
; an example of CND is '((NOT((1) < (2)))) OR ((2) = 5) AND ((1) >= (3))
; (i) means the ith attribute, i the number i
(COND ((NUMBERP CND) CND)
      ((EQ CND T) T)
      ((EQ CND NIL) NIL)
      ((= (LENGTH CND) 1) (NTH TUPLE (CAR CND)))
      ((= (LENGTH CND) 2) (EVAL (LIST (CAR CND)
                                         (EVALCOND TUPLE (CADR CND)))))
      (T (EVAL (LIST (CADR CND)
                     (EVALCOND TUPLE (CAR CND))
                     (EVALCOND TUPLE (CADDR CND)))))
      )
)
```

```
(EVALCOND '(3 5 7 4 1) '((NOT((1) < (2))) or ((2) = 5)))
T
```

```
(DEFUN F-SIGMA (RELATION CND)
; gives the selection of the condition CND on RELATION
(COND ((NULL RELATION) NIL)
      ((EVALCOND (CAR RELATION) CND)
       (CONS (CAR RELATION) (F-SIGMA (CDR RELATION) CND)))
      (T (F-SIGMA (CDR RELATION) CND))
      )
)
```

```
(F-SIGMA R '(((1) >= (2)) AND (2 <= (3))))
((4 4 5) (5 4 2))
```

Met S gedefinieerd als volgt:

```
;S(2)
(SETF S '(
(1 2)
(3 3)
(2 4)
(4 1)
(4 4)
(2 4) ))
```

(F-SORT S 2 2) geeft als resultaat:

```
((2 4) (4 4) (2 4) (3 3) (1 2) (4 1))
```

Wat wordt het resultaat van (F-SORT S 2 2) als men de APPEND in F-SORT vervangt door

```
(APPEND R3
  (F-SORT (COND ((= J 1) R1) (T R2)) I J)
  (F-SORT (COND ((= J 1) R2) (T R1)) I J))))
```

3. Stel dat $X, Y, Z \subseteq \Omega$, $X, Y, Z \subseteq \Omega$ met $XUYUZ = \Omega XUYUZ = \Omega$.

Laat $F = \{X-Z \rightarrow Z-Y, Y \rightarrow X\}$, een verzameling van twee functionele afhankelijkheden.

1. Is XX een supersleutel? Zo ja, bewijs via de Armstrong axioma's; zoniet, geef een tegenvoorbeeld.
2. Is YY een supersleutel? Zo ja, bewijs via de Armstrong axioma's; zoniet, geef een tegenvoorbeeld.
3. Is ZZ een supersleutel? Zo ja, bewijs via de Armstrong axioma's; zoniet, geef een tegenvoorbeeld.

Praktijk

1. Deze vraag is gebaseerd op een muziekdatabank die er zo uitziet.

In de databank worden er van de liedjes de titel, de lengte en de release datum bijgehouden. Van de artiesten wordt de artiestennaam, de (gewone) naam, het adres en het telefoonnummer bijgehouden. Van platenfirma's houden we de naam, het adres en het telefoonnummer bij. Verder moet het mogelijk zijn om te weten welk genre een bepaald liedje van een bepaalde artiest bij een zekere platenfirma heeft. Verder kan je de merchandise opzoeken die een artiest in samenwerking met zijn platenfirma op de markt brengt. Van het merchandise artikel in kwestie vind je het nummer en de omschrijving terug in de databank. Verder zitten er ook winkels in de databank, met hun btwnummer, naam, adres en telefoonnummer, die bepaalde hoeveelheden van de merchandise verkopen.

1. Stel voor deze databank het ER-diagramma op.

2. Map naar het relationele model.

2. Gegeven het volgende schema:

verkoper({ID}, naam, adres, telefoon, email)

koper({ID}, naam, adres, telefoon, email)

product({ID}, titel, jaar)

boek({product.ID}, auteur, ISBN, uitgeverij)

CD({product.ID}, performer, producer)

DVD({product.ID}, regisseur, minuten)

biedtAan({verkoper.ID}, {product.ID}, vraagprijs, startdatum, einddatum, hoeveelheid)

koopt({koper.ID}, {product.ID}, {verkoper.ID}, koopprijs, aankoopdatum, hoeveelheid)

1. Schrijf in relationele algebra: Geef de namen van verkopers die DVD's van meer dan 120 minuten langer dan 20 dagen hebben aangeboden.

2. Schrijf in SQL: Geef de namen van de kopers die meer dan 3 boeken en meer dan 5 CD's hebben gekocht.

3. Schrijf in tuple calculus: Geef de titels van boeken die door een koper A van een verkoper B gekocht werden, waar A en B hetzelfde adres hebben.

3. Gegeven $R(A, B, C, D, E, F, G, H, I)$. Is deze set van functionele afhankelijkheden een minimal cover?

$AB \rightarrow CD \rightarrow EFA \rightarrow DBG \rightarrow EF \rightarrow HH \rightarrow I$
 $AB \rightarrow CD \rightarrow EFA \rightarrow DBG \rightarrow EF \rightarrow HH \rightarrow I$

Zo ja, motiveer je antwoord (geef een lijst van eisen waaraan deze set voldoet). Zo niet, leg uit waarom.

- Schrijf een functie in LISP die, gegeven twee lijnen, een nieuwe lijst maakt door de eerste helft van de eerste lijst met de tweede helft van de tweede lijst te combineren. Bijvoorbeeld, (1 2 3 4) en (5 6 7 8) geven (1 2 7 8).

Genereer ook foutmeldingen voor foutieve input (lijsten met oneven aantal elementen en geen lijsten). Definieer hulpfuncties als nodig.

Academiejaar 2005 - 2006 - 1ste zittijd

Theorie

- Leg de volledige semantiek van recursieve queries zonder negatie (slide 65) uit. Wat loopt er mis met de negatie?
- Beschouw de relaties Student(snaam), Prof (pnaam), Hand(snaam, pnaam) die uitdrukken welke student welke prof al een hand heeft gegeven. Wat betekent:

1. Hand : Student

2. Hand

π_{snaam}

π_{snaam}

(Hand)

3. Student $\bar{\cap}$ Prof : Student

4. Student $\bar{\cap}$ Prof : Hand

Leg de formule R : S van slide 14 uit.

- Geef verschillen tussen een programmeertaal en een query-taal. Welke soorten querytalen zijn er? Waar in deze soorten kan je de relationele algebra, de relationele calculi, SQL en datalog situeren, en waarom?
- Leg de eerste 4 lijnen van de calculus expressie op slide 80 (Monotone increasing sequences) duidelijk uit.

Praktijk

- Gegeven een jeugdvereniging. De relatie R(A,B) bestaat uit tuples (a, b) als en slechts als a en b beiden lid zijn van de jeugdvereniging. Welke niet-triviale Functional Dependencies gelden er? Welke triviale en welke niet-triviale Multi-Valued Dependencies gelden er? In welke normaalvorm is R?

2. Gegeven de volgende databank

Pilot(pil#, name, salary)

Flies(pil#, pl#, hours)

Plane(pl#, plane type)

Stel volgende queries in de Relationele Algebra, in een Relationele Calculus en in SQL.

1. Geef alle piloten (naam) die niet vliegen met een vliegtuig waar "P2 mee vliegt" (P2 is een pil#)
2. Geef alle piloten (naam) die met juist 1 vliegtuig vliegen waar "P2 mee vliegt"
3. Vind de namen van de piloten die vliegen met het meest aantal vliegtuigen.