

## Data & Informatie Toets 3

6 juni 2014, 13:45-15:30

### Triggers, Transactions, Web Applications, Web Services and XML

De toets bevat 4 opgaven (13 vragen) op 4 pagina's die gezamenlijk 10,0 punten opleveren. Bij het tentamen mogen geen boeken, aantekeningen of elektronische apparaten worden gebruikt.

#### Opgave 1

Voor de administratie van een huisartsenpraktijk ziet een deel van het databaseschema er als volgt uit:

Patient (patientnr , adres, naam, ... )  
Factuur (patientnr , datum, bedrag, ... )

Een *patientnr* identificeert een patiënt uniek, en er worden in de administratie geen *patientnrs* gebruikt die niet in *Patient* voorkomen. Een factuur wordt uniek geïdentificeerd door *patientnr* en *datum* samen. We willen dat iedere wijziging van de databasetoestand zich houdt aan de volgende regels:

1. De gegevens van een patiënt worden niet uit *Patient* verwijderd als zijn *patientnr* nog in Factuur voorkomt.
2. Het *patientnr* in Factuur verandert op gelijke wijze als in *Patient* wanneer in *Patient* een wijziging wordt aangebracht.

**Vraag a (1,0 punt)** Voeg key constraints toe aan het schema zodat wijzigingen van de databasetoestand zich automatisch houden aan bovenstaande regels; voeg geen overbodige constraints toe. Ter herinnering, key constraints zien er als volgt uit:

primary key (*x*<sub>1</sub>, *x*<sub>2</sub>, ... ),

foreign key (*x*<sub>1</sub>, *x*<sub>2</sub>, ... ) references *T* (*y*<sub>1</sub>, *y*<sub>2</sub>, ... ) on *event1* *action1* on *event2* *action2* ...

Hierbij staat *x* voor een attribuutnaam, *T* voor een tabelnaam, *event* voor een gebeurtenis (delete, update) en *action* voor een actie (set null, set default, no action, cascade).

**Vraag b (1,0 punt)** In plaats van key constraints toe te voegen kan de wens ook automatisch door het databasesysteem gerealiseerd worden door geschikte triggers toe te voegen aan het databaseschema. Ter herinnering, de syntaxis van een trigger creatie luidt als volgt:

```
create trigger trigger-name
{before | after} {insert | delete | update [ of column-name-list ] } on table-name
[ referencing [ old as var ][ new as var ] [ old table as var ][ new table as var ] ]
[ for each { row | statement } ]
```

[ when (*precondition*) ]  
*statement-list*

Hierbij staat { $x$  |  $y$  | ... } voor “een keuze uit  $x$ ,  $y$ , ...”; en [ $x$ ] staat voor “optioneel  $x$ ”. Geef de trigger create statements voor regel 1 en 2 van de opgave.

## Opgave 2

Beschouw de twee volgende transacties:

$T_1 = \text{read}_1(x); \text{write}_1(y); \text{write}_1(x);$   
 $T_2 = \text{read}_2(y); \text{read}_2(x); \text{write}_2(y);$

Hierbij staan  $x$  en  $y$  voor verschillende data items, en  $\text{read}_i()$  en  $\text{write}_i()$  voor respectievelijk een voor een lees- en schrijfactie van transactie  $i$ . De isolatieniveaus van transacties  $T_1$  en  $T_2$  worden op SERIALIZABLE gesteld, en in de implementatie wordt dit met het locking protocol gerealiseerd. De transacties worden tegelijkertijd uitgevoerd.

**Vraag a (1,0 punt)** Geef alle schedules van transacties  $T_1$  en  $T_2$  die beginnen met:

$\text{read}_1(x); \text{read}_2(y); \text{read}_2(x); \dots$

en waarin iedere transactie met een "commit $_i$ " eindigt (waarbij  $i$  het transactienummer 1 of 2 is).

**Vraag b (1,0 punt)** Is er een equivalent serieel schedule voor onderstaande schedule van transacties  $T_3$  en  $T_4$ ? (deze vraag gaat dus niet langer over transacties  $T_1$  en  $T_2$ )

$\text{read}_4(z); \text{read}_3(x); \text{read}_4(y); \text{read}_3(y); \text{write}_4(y); \text{write}_3(x);$

Leg uit hoe je tot je antwoord komt. (Als het antwoord ja is, geef dan aan met welke volgorde van transacties  $T_3$  en  $T_4$  de schedule equivalent is.)

## Opgave 3

**Vraag a (0,5 punt)** Wat is 'deployment' en waarom is dat nodig voor web applicaties?

**Vraag b (0,5 punt)** Geef een belangrijk voordeel van het gebruik van een MVC architectuur in de ontwikkeling van een web applicatie.

**Vraag c (0,5 punt)** Wat zou de rol moeten zijn van Servlets, Java Beans en JSP pages in een MVC architectuur voor een web applicatie als behandeld in deze module?

**Vraag d (0,5 punt)** Hoe zijn de REST CRUD (Create-Read-Update-Delete) methodes gerelateerd aan HTTP?

**Vraag e (1,0 punt)** Beschouw het volgende HTTP request bericht:

```
GET /jws.rest.first/rest/hello HTTP/1.1
Accept: text/xml
User-Agent: Java/1.6.0_25
Host: localhost:8080
Connection: keep-alive
```

Dit HTTP request bericht wordt in een RESTful service implementatie door een object van de volgende klasse afgehandeld:

```
package di.rest.first;

import javax.ws.rs.Path;
import javax.ws.rs.GET;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/hello")
public class Hello {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String sayPlainTextHello() {
        return "Hello Jersey";
    }

    @GET
    @Produces(MediaType.TEXT_XML)
    public String sayXMLHello() {
        return "<xml version='1.0'?'>\n<hello>HelloJersey</hello>";
    }

    @GET
    @Produces(MediaType.TEXT_HTML)
    public String sayHTMLHello() {
        return "<html>\n" +
            "<title>Hello Jersey</title>\n" +
            "<body><h1> Hello Jersey </h1></body>\n" +
            "</html>";
    }
}
```

Welk antwoord wordt door deze service terug gestuurd? Waarom?

**NB:** Antwoorden zonder verklaring worden als fout gerekend.

## Opgave 4

We hebben een document `movies.xml` met daarin informatie over een heel aantal films. Hieronder staat een voorbeeld van zo'n document met één film erin. Neem aan dat grotere documenten simpelweg meer films bevatten op dezelfde manier gestructureerd als deze.

```
<movies>
  <movie imdbnr="0360717">
    <title>King Kong</title>
    <year>2005</year>
    <genres>
      <genre>Action</genre>
      <genre>Adventure</genre>
      <genre>Drama</genre>
      <genre>Thriller</genre>
    </genres>
    <plot>In 1933 New York, ...</plot>
    <actors>
      <actor>Naomi Watts</actor>
    </actors>
  </movie>
</movies>
```

```
<actor>Jack Black</actor>
<actor>Adrien Brody</actor>
</actors>
</movie>
</movies>
```

Geef een XQuery voor de onderstaande vragen. Maak ze zo kort mogelijk.

**NB:** Je query's moeten niet alleen een correct antwoord geven voor het voorbeelddocument, maar voor alle documenten die op dezelfde manier gestructureerd zijn.

**Vraag a (0,75 punten)** Produceer een lijst met alle jaren waarin een film getiteld “King Kong” is gemaakt. Je hoeft hierbij geen Full-Text-faciliteiten te gebruiken; neem maar aan dat de titel precies “King Kong” is in alle gevallen.

**Vraag b (0,75 punten)** Produceer een overzicht met daarin voor elke actor in hoeveel films hij/zij gespeeld heeft en ook een lijstje met die films.

**Vraag c (0,75 punten)** Update het document door het genre “Romance” aan alle films genaamd “King Kong” toe te voegen.

**Vraag d (0,75 punten)** Vind alle films mbv de Full-Text faciliteiten waarbij in de ‘plot’ hetzelfde jaar genoemd wordt als in het ‘year’-element.