

Opdracht 1 - Constraints

Verslag

Analyse 8 - Advanced Databases (2014-2015)

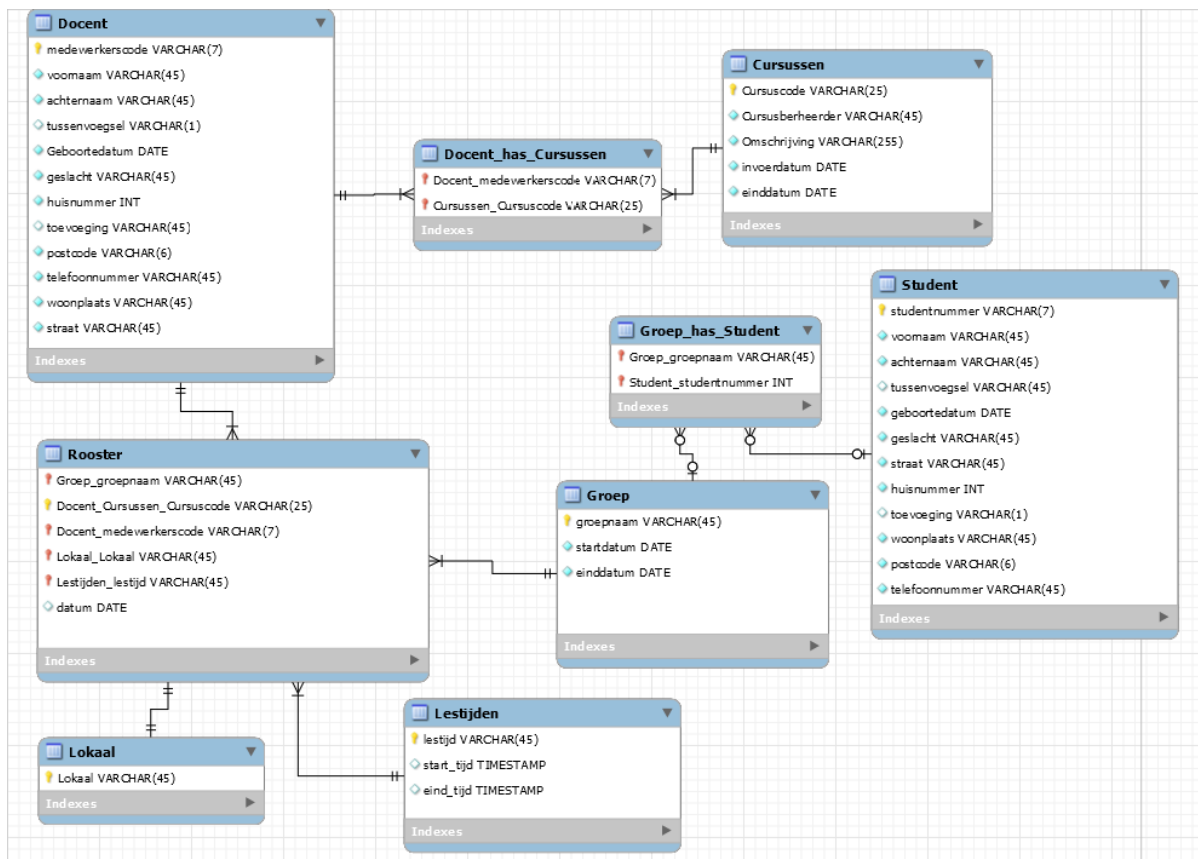
Namen:	HoYe Lam, Rinesh Ramadhin
Studentnummer:	0876814, 0882447
Klas:	INF2D
Vak:	INFANL01-8
Opdracht:	Constraints
Datum:	16 – 05 - 2015

Inhoud

Ontwerp	2
Realisatie	3
Constraints	3
Trigger	6
Testen	7
Constraints	7
Trigger	8

Ontwerp

Hieronder ziet u het ERD van de HR database:



Na het bekijken van de opdracht en het bespreken met docent hebben wij ervoor gekozen om de volgende tabellen te maken:

- Student
- Docent
- Groep
- Cursussen
- Rooster
- Lokaal
- Lestijden

Hierbij ziet u dat de student in nul of meerdere groepen kan zitten. Ook kan een groep nul of meerdere studenten bevatten. Daarnaast heeft de docent een of meerdere cursussen en een cursus heeft een of meerdere docenten. Ons "Rooster" heeft een een-op-meer relatie met de "Docent", dus de een roostermoment heeft een docent en een docent kan meerdere roostermomenten les geven.

Daarnaast heeft ons rooster ook een een-op-meer relatie met "Groep", zodat een roostermoment een groep heeft en een groep meerdere roostermomenten kan les volgen.

Tot slot hebben wij aparte tabellen gemaakt voor lokaal en lestijden. De reden hiervoor is, omdat je dan kijken of het lokaal en de lestijd vrij/bezet is voor de groep en de docent.

Realisatie

Constraints

Hieronder vindt U het PostgreSQL Script dat we hebben geschreven om automatisch de database en tabellen te creëren. Ook vind u hierin de constraints die we hebben toegepast.

```
-- Database: "Week1"

--DROP DATABASE "Week1"

DROP SCHEMA public CASCADE;
CREATE SCHEMA public;

CREATE DATABASE "Week1"
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       LC_COLLATE = 'Dutch_Netherlands.1252'
       LC_CTYPE = 'Dutch_Netherlands.1252'
       CONNECTION LIMIT = -1;
```

Hier maken we de database aan. Ook wordt vooraf even de SCHEMA verwijderd zodat we iedere keer van een schone start kunnen beginnen.

```
CREATE TABLE Student(
  studentnummer          VARCHAR(7) NOT NULL PRIMARY KEY CHECK (length(studentnummer) = 7)
  , voornaam              VARCHAR(45) NOT NULL
  , achternaam            VARCHAR(45) NOT NULL
  , tussenvoegsel         VARCHAR(45)
  , geboortedatum        DATE NOT NULL CHECK (geboortedatum <= current_DATE)
  , geslacht              VARCHAR(45) CHECK (Geslacht = 'man' OR Geslacht = 'Vrouw' OR Geslacht = 'Onbepaald')
  , straat                VARCHAR(45) NOT NULL
  , huisnummer            INT NOT NULL
  , toevoeging            VARCHAR(1)
  , woonplaats            VARCHAR(45) NOT NULL
  , postcode              VARCHAR(6) NOT NULL
  , telefoonnummer        VARCHAR(45) NOT NULL CHECK (telefoonnummer ~ '^([0-9]\." "()-
]*)$')
);
```

De tabel “student” wordt aangemaakt. De primaire key (Studentnummer) wordt aangegeven. Er wordt bij enkele rijen aangegeven dat er data MOET worden ingevoerd. Ook worden er enkele constraints gezet. Zo mag de studentnummer maximaal 7 tekens bevatten, de voornaam, achternaam, tussenvoegsels, straat, woonplaats en telefoonnummer maximaal 45 tekens. Ook wordt er gecheckt of de geboortedatum niet in de toekomst ligt. Tevens wordt er ook gekeken of het geslacht precies “man”, “vrouw”, “onbepaald”. Als het geslacht niet bekend is kan er niks worden ingevoerd. Uiteindelijk wordt er bij het telefoonnummer gecheckt of de ingevoerde waarde alleen toegelaten tekens bevat.

```

CREATE TABLE Docent(
    medewerkerscode    VARCHAR(7)    NOT NULL    PRIMARY KEY    CHECK (medewerkerscode~'^[0-9A-Z]{7}$')
    , voornaam          VARCHAR(45)    NOT NULL
    , achternaam        VARCHAR(45)    NOT NULL
    , tussenvoegsel      VARCHAR(45)
    , geboortedatum     DATE           NOT NULL    CHECK (geboortedatum <= current_DATE)
    , geslacht           VARCHAR(45)    CHECK (Geslacht = 'man' OR Geslacht = 'Vrouw' OR Geslacht = 'Onbepaald')
    , straat            VARCHAR(45)    NOT NULL
    , huisnummer        INT            NOT NULL
    , toevoeging        VARCHAR(45)
    , woonplaats        VARCHAR(255)    NOT NULL
    , postcode          VARCHAR(6)     NOT NULL
    , telefoonnummer    VARCHAR(255)    NOT NULL    CHECK (telefoonnummer~'^[0-9\." '()-]*$')
);

```

Hierna wordt de tabel “Docent” gecreëerd. Deze is identiek aan “Student”, met uitzondering van de Medewerkerscode. Hier wordt er gekeken of de medewerkerscode geen spaties bevat.

```

CREATE TABLE Groep(
    groepnaam          VARCHAR(45)    NOT NULL    PRIMARY KEY
    , startdatum       DATE           NOT NULL
    , einddatum        DATE           NOT NULL    CHECK (einddatum >= startdatum)
);

```

Hier wordt de tabel “Groep” gecreëerd. Als speciale checkt wordt er gekeken of de einddatum gelijk is of hoger dan de startdatum is.

```

CREATE TABLE Groep_has_Student(
    groep_groepnaam    VARCHAR(45)    NOT NULL    REFERENCES Groep(groepnaam)
    , student_studentnummer VARCHAR(7)    NOT NULL    REFERENCES Student(studentnummer)
);

```

Een koppeltabel wordt gecreëerd hier.

```

CREATE TABLE Cursussen(
    cursuscode          VARCHAR(25)    NOT NULL    PRIMARY KEY    CHECK (cursuscode = UPPER(cursuscode))
    , cursusbeheerder   VARCHAR(45)    NOT NULL
    , omschrijving      VARCHAR(255)    NOT NULL
    , invoerdatum       DATE           NOT NULL
    , einddatum        DATE           NOT NULL    CHECK (Einddatum > invoerdatum)
);

```

De “Cursussen” tabel wordt gecreëerd. Er wordt gekeken of de Cursuscode in uppercase is. Tevens wordt er ook gecheckt of de einddatum gelijk is / na de invoerdatum is.

```
CREATE TABLE Docent_has_Cursussen(
    docent_medewerkerscode VARCHAR(7) NOT NULL REFERENCES Docent(medewerkerscode)
    , cursussen_Cursuscode VARCHAR(25) NOT NULL REFERENCES Cursussen(cursuscode)
);
```

Nog een koppeltabel

```
CREATE TABLE Lokaal(
    lokaal VARCHAR(45) NOT NULL PRIMARY KEY
);
```

De “Lokaal” tabel wordt gemaakt.

```
CREATE TABLE Lestijden(
    lestijd VARCHAR(45) NOT NULL PRIMARY KEY
    , start_tijd TIME NOT NULL
    , eind_tijd TIME NOT NULL CHECK (eind_tijd > start_tijd)
);
```

De “Lestijden” tabel wordt gecreëerd. Er wordt ook gekeken of de eindtijd na de starttijd is.

```
CREATE TABLE Rooster(
    groepnaam VARCHAR(45) NOT NULL REFERENCES Groep(groepnaam)
    , lokaal VARCHAR(45) NOT NULL REFERENCES Lokaal(lokaal)
    , medewerkerscode VARCHAR(7) NOT NULL REFERENCES Docent(medewerkerscode)
    , Docent_Cursussen_cursuscode VARCHAR(25) NOT NULL REFERENCES Cursussen(cursuscode)
    , lestijd VARCHAR(45) NOT NULL REFERENCES Lestijden(lestijd)
    , datum DATE NOT NULL
);
```

De tabel “Rooster” wordt gecreëerd. Er wordt gekeken of eindles na de startles is.

Trigger

Hieronder vindt de functie en de trigger die we hebben aangemaakt om zo aan de eisen van het rooster te voldoen. Dit gedeelte komt onderaan het Script te staan, net onder de CREATE Rooster gedeelte.

```
CREATE FUNCTION Rooster_check() RETURNS trigger AS $Rooster_check$
BEGIN
    IF ((SELECT COUNT(groepnaam) FROM Rooster WHERE groepnaam = NEW.groepnaam AND datum = NEW.datum
AND lestijd = NEW.lestijd) = 1) THEN
        RAISE EXCEPTION '% heeft al les op dat tijdstip', NEW.groepnaam;
    END IF;

    IF ((SELECT COUNT(medewerkerscode) FROM Rooster WHERE medewerkerscode = NEW.medewerkerscode AND
datum = NEW.datum AND lestijd = NEW.lestijd) = 1) THEN
        RAISE EXCEPTION '% geeft al les op dat tijdstip', NEW.medewerkerscode;
    END IF;

    IF ((SELECT COUNT(lokaal) FROM Rooster WHERE lokaal = NEW.lokaal AND datum = NEW.datum AND
lestijd = NEW.lestijd) = 1) THEN
        RAISE EXCEPTION '% is al bezet op dat tijdstip', NEW.lokaal;
    END IF;

    RETURN NEW;
END
$Rooster_check$ LANGUAGE plpgsql;
```

De functie “Rooster_check” wordt aangemaakt. Hierin wordt er gekeken of een nieuwe INSERT of UPDATE wel mag worden uitgevoerd. Er worden achter mekaar drie checks uitgevoerd. Allereerst wordt er gekeken of de groepnaam (Klas) al in de tabel voorkomt op dezelfde datum en lestijd. Hierna wordt er hetzelfde gedaan voor de medewerkerscode (docent). Uiteindelijk wordt er ook hetzelfde gedaan voor het lokaal.

```
CREATE TRIGGER before_Rooster_insert
BEFORE INSERT OR UPDATE
ON Rooster
FOR EACH ROW
EXECUTE PROCEDURE Rooster_check();
```

Hier wordt de trigger aangemaakt. Deze zorgt ervoor dat voor elke INSERT of UPDATE in de Rooster tabel de bovenvermelde functie wordt uitgevoerd.

Testen

Constraints

Voor het testen hebben wij een script geschreven met enkele simpele inserts. Wij hebben deze zo geschreven dat het elke keer een andere constraint test. Ook wordt er met comments aangegeven wat de geteste constraint is. Hieronder bevindt het script.

```
-- Test inserts

INSERT INTO Student VALUES
('0882447','Rinesh','Ramadhin','','19930926','man','Reling',211','','Barendrecht','2993DR','0643643698');
INSERT INTO Student VALUES
('08824457','Joost','Ruiter','de','19950916','man','Wijnhaven',107','','Rotterdam','1345DA','06484839933'); --
studentnummer te lang
INSERT INTO Student VALUES
('08457','Joost','Ruiter','de','19950916','man','Wijnhaven',107','','Rotterdam','1345DA','06484839933'); --
studentnummer te kort
INSERT INTO Student VALUES
('0123445','Jan','Bakker','de','19970916','Man','Wijnhaven',107','','Rotterdam','1345DA','06484839933'); --
Geslacht verkeerd.
INSERT INTO Student VALUES
('0832445','hoye','lam','','19950916','man','Wijnhaven',107','','Rotterdam','1345DA','gmail.com'); --
studentnummer te lang

INSERT INTO Docent VALUES
('TJANG','dgg','dfgh','ddgvf','19970926','man','Wijnhaven',123','','Barendrecht','2147AA','78907');
INSERT INTO Docent VALUES ('344
556','dfgh','dfwvsg','19971026','man','Wijnhaven',123','','Barendrecht','2993DR','78907'); -- fout
medewerkerscode
INSERT INTO Docent VALUES
('47979','fdghfgh','fdwgh','dcvf','20180926','man','Wijnhaven',123','','Barendrecht','2993DR','78907'); --
geboortedatum
INSERT INTO Docent VALUES
('887979','dffhgh','dfgfrfh','fff','19970926','gijian','Wijnhaven',123','','Barendrecht','2993DR','78907'); --
foutive geslacht
INSERT INTO Docent VALUES
('887979','dfgh','dfgh','dvf','19970926','man','Wijnhaven',123','','Barendrecht','2993DR','06094a509'); -- fout
telefoonnummer

INSERT INTO Groep VALUES ('INF2D','2013-01-01','2013-10-10');
INSERT INTO Groep VALUES ('INF1D','2013-01-01','2013-10-10');
INSERT INTO Groep VALUES ('INF2D','2013-10-01','2013-01-10'); -- einddatum na invoerdatum

INSERT INTO Cursussen VALUES ('INFONZ01-1','TJANG','bablalbla','2013-01-01','2013-10-10');
INSERT INTO Cursussen VALUES ('INFONZ02-1','YOURI','bablalbla','2013-10-01','2013-01-10'); -- einddatum na
invoerdatum
INSERT INTO Cursussen VALUES ('infanl01-1','STELL','badlalbla','2013-01-01','2013-10-10'); -- kleine letters bij
cussuscode

INSERT INTO Rooster VALUES ('INF2D','H.4.002','TJANG','INFONZ01-1','2014-10-01 12:00:00','2014-11-10 12:00:00');
INSERT INTO Rooster VALUES ('INF2D','H.4.002','TJANG','INFONZ01-1','20141010120000','20140110120000');
```


Trigger

Hieronder vind U het script dat automatisch de trigger test. We hebben eerst een klas toegevoegd die aan alle eisen voldoet. Hierna hebben we drie keer een INSERT gedaan, waarbij elke keer een andere check wordt gedaan. Eerst een klas die al les heeft op dat tijdstip, dan hetzelfde voor de docent en een lokaal.

```
-- benodigde inserts om te testen

INSERT INTO Docent VALUES ('1234567','Youri','Tjang','','1990-01-01','man','wijnhaven',99','','Rotterdam','1234AB','0100000');
INSERT INTO Docent VALUES ('7654321','Marco','Noord','van','1995-01-01','man','wijnhaven',107','','Rotterdam','1234BA','0101000');

INSERT INTO Groep VALUES ('INF2D','2014-09-01','2015-08-01');
INSERT INTO Groep VALUES ('INF1D','2013-09-01','2014-08-01');

INSERT INTO Lokaal VALUES ('H.4.318');
INSERT INTO Lokaal VALUES ('H.4.208');

INSERT INTO Cursussen VALUES ('ANALYSE','YouriTjang','analyse vak','2000-01-01','2020-01-01');
INSERT INTO Cursussen VALUES ('PROJECT','MarcovanNoord','Project Emerging Technologies','2010-01-01','2020-01-01');

INSERT INTO Lestijden VALUES ('1','08:30','09:20');
INSERT INTO Lestijden VALUES ('2','09:20','10:10');
INSERT INTO Lestijden VALUES ('3','10:30','11:20');
INSERT INTO Lestijden VALUES ('4','11:20','12:10');
INSERT INTO Lestijden VALUES ('5','12:10','13:00');
INSERT INTO Lestijden VALUES ('6','13:00','13:50');
INSERT INTO Lestijden VALUES ('7','13:50','14:40');
INSERT INTO Lestijden VALUES ('8','15:00','15:50');

-- test trigger

INSERT INTO Rooster VALUES ('INF2D','H.4.318','1234567','ANALYSE','1','2015-01-01');
INSERT INTO Rooster VALUES ('INF2D','H.4.208','7654321','PROJECT','1','2015-01-01'); -- klas heeft al les op dat uur
INSERT INTO Rooster VALUES ('INF1D','H.4.208','1234567','ANALYSE','1','2015-01-01'); -- docent geeft al les op dat uur
INSERT INTO Rooster VALUES ('INF1D','H.4.318','7654321','PROJECT','1','2015-01-01'); -- lokaal is al bezet
```