

Extra oefeningen H 9 – 10

Voor je tabellen kan creëren moet je eerst volgend doen:

Log in als admin: >conn sys/pxl as sysdba

dan volgend commando: >ALTER USER student QUOTA UNLIMITED ON USERS;

Oefening 15 kan als lesvoorbeeld gebruikt worden om kolomconstraints en tabelconstraints uit te leggen. Zit hier in, omdat oefening 16 anders onmogelijk is voor studenten die niet in de les zijn.

1. In ons bedrijf worden er interne cursussen georganiseerd die gedoceerd worden door eigen medewerkers. Alle medewerkers kunnen hiervoor vrijblijvend inschrijven. De cursussen kunnen meermaals georganiseerd worden en elke medewerker kan eenzelfde cursus slechts één maal volgen. Om deze gegevens te kunnen registreren is het nodig dat volgende tabellen worden aangemaakt. Doe dit in een script (crecursus.sql). De gegevens moeten pas achteraf worden ingegeven.

CURSUSSEN

CURS_CODE	CURS_OMSCHR	CURS_TYPE	CURS LENGTE
SQL1	Introductie SQL	BASIC	2
SQL2	SQL Advanced	ADV	3
SQL3	PL/SQL	PROGR	3

Bepaal zelf de primary key en kies de meest correcte datatypes en lengtes.

Zorg ervoor dat de lengte enkel tussen 1 en 10 kan liggen. Alle kolommen behalve CURS_LENGTE moeten steeds ingevuld zijn. CURS_TYPE moet ingegeven worden in hoofdletters.

UITVOERINGEN

CURSUS	BEGIN DATUM	DOCENT	LOCATIE
--------	-------------	--------	---------

SQL1	1/6/2014	107	Brussel
SQL1	8/6/2014	107	Hasselt
SQL2	4/6/2014	201	Brussel
SQL3	10/6/2014	201	Brussel

Bepaal zelf de primary key en kies de meest correcte datatypes en lengtes.

De CURSUS moet voor komen in de tabel CURSUSSEN en de DOCENT moet terug te vinden zijn in de tabel EMPLOYEES. Eenzelfde cursus kan meermaals georganiseerd worden maar altijd met een andere begindatum. Een docent kan verschillende cursussen geven. LOCATIE kan achteraf nog ingevuld worden.

INSCHRIJVINGEN

CURSIST	CURSUS	BEGIN DATUM	EVALUATIE
103	SQL1	1/6/2014	
176	SQL1	1/6/2014	
202	SQL2	4/6/2014	
124	SQL2	4/6/2014	
143	SQL2	4/6/2014	
205	SQL1	8/6/2014	

Bepaal zelf de primary key en kies de meest correcte datatypes en lengtes.

De CURSUS moet voor komen in de tabel UITVOERINGEN en de CURSIST moet terug te vinden zijn in de tabel EMPLOYEES. Alle kolommen behalve EVALUATIE moeten een waarde krijgen. De evaluatie gebeurt op een schaal van 5.

Script zit in de map van de oefeningen: crecursus.sql

REM =====

REM Tabel CURSUSSEN

REM =====

```
-- drop table cursussen cascade constraints;

create table cursussen
( curs_code      VARCHAR2(4) constraint C_PK
                        primary key
, curs_omschr    VARCHAR2(50) constraint C_OMSCHR_NN
                        not null
, curs_type      CHAR(3)   constraint C_TYPE_NN
                        not null
, curs_lengte    NUMBER(2)  constraint C LENGTE_CHK
                                check (curs_lengte between 1 and 10)
, constraint C_TYPE_CHK check
                        (curs_type = upper(curs_type))
) ;
```

```
REM =====
REM Tabel UITVOERINGEN
REM =====
```

```
-- drop table uitvoeringen cascade constraints;

create table uitvoeringen
( cursus    VARCHAR2(4) constraint U_CURSUS_NN
                        not null
                        constraint U_CURSUS_FK
                        references cursussen(curs_code)
, begindatum DATE        constraint U_BEGIN_NN
                        not null
, docent     NUMBER(4)   constraint U_DOCENT_FK
                        references employees(employee_id)
, locatie    VARCHAR2(20)
, constraint U_PK      primary key
                        (cursus,begindatum)
```

```
) ;
```

```
REM =====
```

```
REM Tabel INSCHRIJVINGEN
```

```
REM =====
```

```
-- drop table inschrijvingen cascade constraints;
```

```
create table inschrijvingen
```

```
( cursist    NUMBER(4)  constraint I_CURSIST_NN
```

```
            not null
```

```
            constraint I_CURSIST_FK
```

```
            references employees(employee_id)
```

```
, cursus    VARCHAR2(4) constraint I_CURSUS_NN
```

```
            not null
```

```
, begindatum DATE      constraint I_BEGIN_NN
```

```
            not null
```

```
, evaluatie  NUMBER(1)  constraint I_EVAL_CHK
```

```
            check (evaluatie in (0,1,2,3,4,5))
```

```
, constraint I_PK      primary key
```

```
            (cursist,cursus)
```

```
, constraint I_UITV_FK  foreign key
```

```
            (cursus,begindatum)
```

```
            references uitvoeringen
```

```
) ;
```

2. Vul de 3 nieuwe tabellen met de voorbeeld gegevens uit opdracht 1.

```
INSERT INTO cursussen
```

```
VALUES ('&code','&omschrijving','&type','&duur);
```

```
SAVE vulcursus.sql
```

```
INSERT INTO uitvoeringen
```

```
VALUES ('&code','&begin','&docent','&locatie');
```

```
SAVE vuluitvoering.sql
```

```
INSERT INTO inschrijvingen (cursist,cursus,begindatum)
```

```
VALUES (&cursist,'&cursus','&begin');
```

```
SAVE vulinschrijf.sql
```

Voer deze scripts zo vaak uit als nodig om alle gegevens in de tabellen te zetten.

3. Wijzig de locatie van de cursus SQL2 van 4/6/2014 naar Leuven.

```
UPDATE uitvoeringen
```

```
SET Initcap(locatie)='Leuven'
```

```
WHERE UPPER(cursus)='SQL2' AND begindatum=to_date('04-JUN-14','dd-MON-yy');
```

4. Alle medewerkers die Steven King als baas hebben moeten met behulp van 1 commando ingeschreven worden voor de cursus SQL1 van 8/6/2014.

```
INSERT INTO inschrijvingen(cursist,cursus,begindatum)
```

```
(SELECT employee_id, 'SQL1',TO_DATE('08/06/2014','dd/mm/yyyy')
```

```
FROM employees
```

```
WHERE manager_id = (SELECT employee_id FROM employees WHERE  
UPPER(last_name)='KING' and UPPER(first_name)='STEVEN'));
```

5. Doe de nodige aanpassing opdat de cursus SQL3 zal gedoceerd worden door de medewerker met als job PROGRAMMER en die het langst in dienst is.

```
UPDATE uitvoeringen
```

```
SET docent = (SELECT employee_id
```

```
FROM employees
```

```
WHERE job_id = (SELECT job_id
```

```
FROM jobs
```

```
WHERE LOWER(job_title) = 'programmer')
```

```
AND hire_date = (SELECT min(hire_date)
```

```
FROM employees
```

```
WHERE job_id = (SELECT job_id
```

```
FROM jobs
WHERE LOWER(job_title)='programmer'))
WHERE cursus='SQL3';
```

6. Verwijder uit de tabel inschrijvingen alle cursisten die inschreven voor een cursus met als locatie Hasselt.

```
DELETE FROM inschrijvingen
WHERE( cursus,begindatum) = (SELECT cursus, begindatum
FROM uitvoeringen
WHERE UPPER( locatie )='HASSELT');
```

7. Zorg ervoor dat deze verwijdering niet wordt uitgevoerd.

```
ROLLBACK;
```

8. Maak een nieuwe tabel CURS_HISTORY met exact dezelfde structuur en inhoud als de tabel CURSUSSEN.

```
CREATE TABLE curs_history
AS (SELECT * FROM cursussen);
```

9. Verwijder de tabel CURS_HISTORY.

```
DROP TABLE curs_history;
```

10. Maak een tabel King_medewerkers met alle medewerkers die King als baas hebben (op dit moment). Hierin zitten de kolommen employee_id, last_name, first_name, departement_id, datum_registratie (dit is de eerste dag van de huidige maand).

```
CREATE TABLE king_medewerkers
AS SELECT employee_id, first_name, last_name, TO_DATE('01-' ||
TO_CHAR(sysdate,'mm-yyy'),'dd-mm-yyy') as datum_registratie
FROM employees
WHERE manager_id=(SELECT employee_id
FROM employees
```

```
WHERE Initcap(last_name)='King');
```

11. Doe een aanpassing in de tabel King_medewerkers: bij personen waar in de last_name een spatie voorkomt, wordt deze verwijderd.

Dus: De Haan wordt DeHaan

```
UPDATE king_medewerkers  
SET last_name=replace(last_name, ' ','');
```

12. Schrijf een scriptfile die bij uitvoering het loon van de mensen die in dezelfde afdeling zitten als Jones, gelijkstelt aan het laagste loon van afdeling 30.

Voordat je deze query uitvoert, zorg je ervoor dat je een punt voorziet, zodat je deze wijziging later ongedaan kan maken.

```
SAVEPOINT oefening12  
UPDATE employees  
SET salary = ( SELECT MIN(salary)  
                FROM employees  
                WHERE department_id=30)  
WHERE department_id=(SELECT department_id  
                      FROM employees  
                      WHERE LOWER(last_name)='jones');
```

13. Na uitvoeren van oef 12, toon je de naam en het loon van de mensen die in de afdeling van Jones zitten.

```
SELECT last_name, salary  
FROM employees  
WHERE department_id=(SELECT department_id  
                     FROM employees  
                     WHERE LOWER(last_name)='jones');
```

14. Toon nu ook het laagste loon van afdeling 30. Als het loon in oef 13 gelijk is aan het laagste loon uit afdeling 30 is je query uit oef 12 juist
Zorg er nu voor dat enkel de wijziging uit oef 12 ongedaan gemaakt

wordt en niet de andere wijzigingen door eerdere oefeningen.

```
SELECT MIN(salary)
FROM employees
WHERE department_id=30;
```

ROLLBACK TO oefening12

15. Maak een tabel artikels, met een id (tekst van max 10 karakters, PK), naam (tekst, 30 kaarakters, moet altijd een waarde hebben), prijs (decimaal getal bestaande uit 6 cijfers voor en 2 cijfers na de komma), datumvan bestaan en land van fabricatie (2 karakters), die moet verwijzen naar de PK van de tabel countries.

Maak deze oefening 1x met kolomconstraints en 1 keer met tabelconstraints waar mogelijk.

met kolomconstraints:

```
create table artikels
(artikel_id    varchar2(10) constraint art_pk primary key
,artikel_naam  varchar2(30) constraint art_nm_nn not null
,prijs         number(6,2) constraint art_pr  check (prijs > 0.5)
,bestaat_sinds date    default sysdate
,land_fabricage char(2) constraint art_land_fk references countries
);
```

met tabelconstraints:

```
create table artikels
(artikel_id    varchar2(10)
,artikel_naam  varchar2(30) constraint art_nm_nn not null
,prijs         number(6,2)
,bestaat_sinds date    default sysdate
,land_fabricage char(2)
```



```
, constraint art_pk primary key(artikel_id)
, constraint art_pr check (prijs > 0.5)
, constraint art_land_fk foreign key (land_fabricage) references countries
);
```

NOT NULL kan niet via een tabelconstraint!

16. Schrijf een scriptfile dat bij uitvoering een tabel Magazijnen creëert die de volgende eigenschappen heeft: magazijnnummer, naam magazijn, straat, huisnummer en gemeente. De primaire sleutel is de magazijnnummer. Ook wordt er een tabel Magazijn_Artikels aangemaakt. Deze tabel bevat het nummer van het magazijn, het artikelnummer en het aantal items van dat artikel in dat magazijn. Zorg voor de Primary key en nodige Foreign keys ,zodat de juiste relaties gelegd worden.

```
CREATE TABLE magazijn
(magazijn_id NUMBER(2) CONSTRAINT magazijn_id_pk PRIMARY KEY
,magazijn_naam VARCHAR2(15)
,straat VARCHAR2(25)
,huisnummer NUMBER(4)
,gemeente VARCHAR2(25));
```

```
CREATE TABLE magazijn_artikels
(magazijn_id NUMBER(2) constraint mag_art_magid_fk REFERENCES
magazijn (magazijn_id)
,artikel_id NUMBER(6) constraint mag_artid_fk REFERENCES
artikels(artikel_id)
,hoeveelheid NUMBER(4)
, CONSTRAINT mag_art_magid_artid_pk PRIMARY KEY (magazijn_id,
artikel_id));
```

17. Extra oefening voetbal.

- A. Creëer volgende 3 tabellen met de nodige constraints en sla de SQL-commando's die je daarvoor gebruikt op in het bestand crevoetbal.sql zodat je uiteindelijk een script bekomt voor de creatie van de 3 tabellen.

SPELERS

spelersnr - numeriek - 5 lang - primaire sleutel

naam - alfanum - 20 lang - verplicht - moet ingegeven worden in hoofdletters

voorletter - alfanum - 3 lang

geslacht - alfanum - 1 lang - verplicht en standaard "M"

gebdatum - datumveld - moet ingevuld worden

ploegnr - numeriek - 3 lang - moet voorkomen in de tabel TEAMS

TEAMS

teamnr - numeriek - 3 lang - primaire sleutel

teamnaam - alfanum - 20 lang - moet in hoofdletters

kapitein - numeriek - 5 lang - moet voorkomen in de tabel

SPELERS - moet ingevuld zijn

BOETES

boetenr - numeriek - 3 lang - primaire sleutel

spelersnr - num.- 5 lang - moet voorkomen in de tabel spelers

datum - datumveld - verplicht

bedrag - numeriek - 5 cijfers voor en 2 na de komma - verplicht -
moet minimaal 100 € zijn

Vul deze tabellen ook op met enkele gegevens!!

create table spelers

```
(spelersnr number(5) constraint S_SNR_PK primary key  
,naam varchar(20) constraint S_NM_NN not null  
,voorletter varchar(3)  
,geslacht varchar(1) default 'M'  
constraint S_GESL_NN not null  
,gebdatum date constraint S_GBDAT_NN not null  
,ploegnr number(3) )
```

LET OP : veld geslacht !

Default waarde + constraint

De default waarde moet eerst staan, dan pas de constraint

insert into spelers

```
values(1,'Jansen','J','V', to_date('29-05-1984','dd-mm-yyyy'),100)
```

LET OP: de constraint dat het ploegnr moet voorkomen in teams kan nog niet, want teams bestaat niet.

create table teams

```
(teamnr number(3) constraint T_TNR_PK primary key  
,teamnaam char(20) constraint T_NAAM_CHK check (teamnaam =  
upper(teamnaam))  
,kapitein number(5) constraint T_KAP_FK references spelers(spelersnr)
```

```
constraint T_KAP_NN not null )
```

```
insert into teams values (100,'DE KAMPIOENEN',1)
```

Stel dat je de kolom kapitein pas later toevoegt:

```
alter table teams
```

```
add kapitein number(5) constraint T_KAP_FK references spelers
```

```
constraint T_KAP_NN not null
```

```
alter table spelers
```

```
add constraint S_PLNR_FK foreign key (ploegnr) references teams(teamnr)
```

LET OP : dit kan enkel als de huidige records nu al aan deze eis voldoen.

```
create table boetes
```

```
(boetenr number(3) constraint B_BNR_PK primary key
```

```
, spelersnr number(5) constraint B_SNR_FK references spelers(spelersnr)
```

```
,datum date constraint B_DAT_NN not null
```

```
,bedrag number(7,2) constraint B_BEDR_NN not null)
```

```
alter table boetes
```

```
add constraint B_DAT_CH check(bedrag >= 100)
```

```
insert into boetes values(10,1,'10-04-2000',500)
```

```
insert into spelers
```

```
values(&spelersnr,&naam,&voorletter,&geslacht,&gebdatum,&ploegnr);
```

Enter value for spelersnr: 3

Enter value for naam: Lemmens

Enter value for voorletter: P

Enter value for geslacht: M

Enter value for gebdatum: 01-01-1960

Enter value for ploegnr: 100

- B. Probeer nu een speler die een boete heeft te verwijderen. Waarom lukt dat niet? Vanwege de foreign key constraint bij boetes foutmelding: child record found
- C. Maak een tabel BOETES_BIS met dezelfde structuur en dezelfde inhoud als de tabel BOETES.
`CREATE TABLE boetes_bis`
`AS (SELECT * FROM boetes);`
`INSERT INTO boetes_bis`
`(SELECT * FROM boetes);`
- D. Maak de tabel BOETES_BIS leeg. Controleer of dit gebeurd is.
`DELETE boetes_bis;`
`SELECT *`
`FROM boetes_bis;`
- E. Herstel de vorige verwijdering.
`ROLLBACK;`
- F. Verwijder de BOETES_BIS-tabel.
`DROP TABLE boetes_bis;`
- G. Maak voor de tabel TEAMS het synonim PLOEG.
`CREATE SYNONYM ploeg FOR teams;`