**Extra oefeningen H 7 – 9 – 10 - 11**

1. Los oef3 uit de extra oefeningen H6 op met een subquery..

**SELECT** j.EMPLOYEE\_ID, e.LAST\_NAME, j.JOB\_ID, f.JOB\_TITLE

**FROM** JOB\_HISTORY j

**JOIN** EMPLOYEES e **ON** (j.EMPLOYEE\_ID = e.EMPLOYEE\_ID)

**JOIN** JOBS f **ON** (j.JOB\_ID = f.JOB\_ID)

**WHERE** j.JOB\_ID = (**SELECT** JOB\_ID

**FROM** EMPLOYEES

**WHERE** EMPLOYEES.LAST\_NAME = 'Abel');

1. Los oef5 uit de extra oefeningen H6 op met een subquery.

**SELECT** *UPPER*(FIRST\_NAME) || ' ' || *UPPER*(LAST\_NAME)

**FROM** EMPLOYEES

**WHERE** MANAGER\_ID != (**SELECT** MANAGER\_ID

**FROM** DEPARTMENTS

**WHERE** EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID);

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 | BEGINDATUM | 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 | BEGINDATUM | 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 CURSUSSEN 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.

**DROP TABLE** CURSUSSEN **CASCADE CONSTRAINTS**;

**DROP TABLE** UITVOERINGEN **CASCADE CONSTRAINTS**;

**DROP TABLE** INSCHRIJVINGEN **CASCADE CONSTRAINTS**;

**CREATE TABLE** CURSUSSEN

(

CURS\_CODE **VARCHAR2**(4)

,CURS\_OMSCHRIJVING **VARCHAR2**(10) **CONSTRAINT** C\_CO\_NN **NOT NULL**

,CURS\_TYPE **VARCHAR2**(10) **CONSTRAINT** C\_CT\_CH **CHECK** (CURS\_TYPE = *upper*(CURS\_TYPE))

,CURS\_LENGTE **NUMBER**(1)

,**PRIMARY KEY** (CURS\_CODE)

);

**CREATE TABLE** UITVOERINGEN

(

CURSUS **VARCHAR2**(4) **CONSTRAINT** U\_C\_FK **REFERENCES** CURSUSSEN(CURS\_CODE)

,BEGINDATUM **DATE**

,DOCENT **NUMBER**(3) **CONSTRAINT** U\_D\_FK **REFERENCES** EMPLOYEES(EMPLOYEE\_ID)

**CONSTRAINT** U\_D\_NN **NOT NULL**

,LOCATIE **VARCHAR2**(10)

,**PRIMARY KEY** (CURSUS, BEGINDATUM)

);

**CREATE TABLE** INSCHRIJVINGEN

(

CURSIST **NUMBER**(3) **CONSTRAINT** I\_CS\_FK **REFERENCES** EMPLOYEES(EMPLOYEE\_ID)

,CURSUS **VARCHAR2**(4) **CONSTRAINT** I\_CU\_FK **REFERENCES** CURSUSSEN(CURS\_CODE)

,BEGINDATUM **DATE**

,EVALUATIE **NUMBER**(1) **CONSTRAINT** I\_E\_CH **CHECK** (EVALUATIE <= 5)

**CONSTRAINT** I\_E\_CH2 **CHECK** (EVALUATIE >= 0)

,**PRIMARY KEY** (CURSIST, CURSUS, BEGINDATUM)

);

1. Vul de 3 nieuwe tabellen met de gegevens in opdracht 3.

**INSERT INTO** CURSUSSEN **VALUES** ('SQL1', 'Introducti', 'BASIC', 2);

**INSERT INTO** CURSUSSEN **VALUES** ('SQL2', 'SQL Advanc', 'ADV', 3);

**INSERT INTO** CURSUSSEN **VALUES** ('SQL3', 'PL/SQL', 'PROGR', 3);

**INSERT INTO** UITVOERINGEN **VALUES** ('SQL1', *TO\_DATE*('1/6/2014', 'DD/MM/YYYY'), 107, 'Brussel');

**INSERT INTO** UITVOERINGEN **VALUES** ('SQL1', *TO\_DATE*('8/6/2014', 'DD/MM/YYYY'), 107, 'Hasselt');

**INSERT INTO** UITVOERINGEN **VALUES** ('SQL2', *TO\_DATE*('4/6/2014', 'DD/MM/YYYY'), 201, 'Brussel');

**INSERT INTO** UITVOERINGEN **VALUES** ('SQL3', *TO\_DATE*('10/6/2014', 'DD/MM/YYYY'), 201, 'Brussel');

**INSERT INTO** INSCHRIJVINGEN **VALUES** (103, 'SQL1', *TO\_DATE*('1/6/2014', 'DD/MM/YYYY'), **NULL**);

**INSERT INTO** INSCHRIJVINGEN **VALUES** (176, 'SQL1', *TO\_DATE*('1/6/2014', 'DD/MM/YYYY'), **NULL**);

**INSERT INTO** INSCHRIJVINGEN **VALUES** (202, 'SQL2', *TO\_DATE*('4/6/2014', 'DD/MM/YYYY'), **NULL**);

**INSERT INTO** INSCHRIJVINGEN **VALUES** (124, 'SQL2', *TO\_DATE*('4/6/2014', 'DD/MM/YYYY'), **NULL**);

**INSERT INTO** INSCHRIJVINGEN **VALUES** (143, 'SQL2', *TO\_DATE*('4/6/2014', 'DD/MM/YYYY'), **NULL**);

**INSERT INTO** INSCHRIJVINGEN **VALUES** (205, 'SQL1', *TO\_DATE*('8/6/2014', 'DD/MM/YYYY'), **NULL**);

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

**UPDATE** UITVOERINGEN

**SET** LOCATIE = 'Leuven'

**WHERE** CURSUS = 'SQL2' **AND** BEGINDATUM = *TO\_DATE*('4/6/2014', 'DD/MM/YYYY');

1. 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

(**SELECT** EMPLOYEE\_ID, 'SQL1', *TO\_DATE*('1/6/2014', 'DD/MM/YYYY'), **NULL**

**FROM** EMPLOYEES

**WHERE** MANAGER\_ID = (**SELECT** EMPLOYEE\_ID

**FROM** EMPLOYEES

**WHERE** LAST\_NAME = 'King' **AND** FIRST\_NAME = 'Steven'));

1. 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** (**SELECT** *MIN*(HIRE\_DATE)

**FROM** EMPLOYEES

**WHERE** JOB\_ID = 'IT\_PROG') = HIRE\_DATE)

**WHERE** CURSUS = 'SQL3';

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

**SAVEPOINT s01;**

**DELETE FROM** INSCHRIJVINGEN

**WHERE** CURSUS = (**SELECT** CURSUS

**FROM** UITVOERINGEN

**WHERE** LOCATIE = 'Hasselt') **AND** BEGINDATUM = (**SELECT** BEGINDATUM

**FROM** UITVOERINGEN

**WHERE** LOCATIE = 'Hasselt');

1. Zorg ervoor dat deze verwijdering niet wordt uitgevoerd.

**ROLLBACK TO** s01;

1. Maak een nieuwe tabel CURS\_HISTORY met exact dezelfde structuur en inhoud als de tabel CURSUSSEN.

**CREATE TABLE** CURS\_HISTORY

**AS** (**SELECT** \*

**FROM** CURSUSSEN);

1. Verwijder de tabel CURS\_HISTORY.

**DROP TABLE** CURS\_HISTORY;