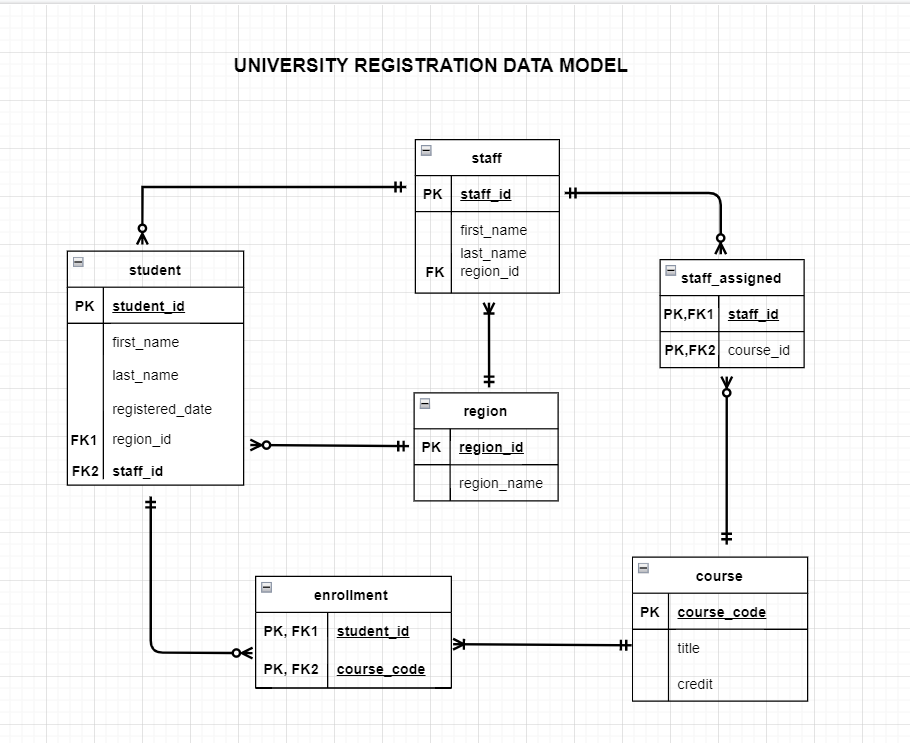
**DBMD LECTURE**

**UNIVERSITY DATABASE PROJECT**

1. DESIGN UNIVERSITY REGISTRATION DATA MODEL ACCORDING TO THE CROW'S FOOD MODEL BY USING DRAW.IO. THE MODEL HAS SEVERAL PARTS, BEGINNING WITH AN ERD AND FOLLOWED BY A WRITTEN DESCRIPTION OF ENTITY TYPES, CONSTRAINTS, AND ASSUMPTIONS.



**Entity :**

course (course\_code, title, credit)

student (student\_id, first\_name, last\_name, registered\_date, region\_id, staff\_id)

staff (staff\_id, first\_name, last\_name, region\_id)

region(region\_id, region\_name )

enrollment (student\_id, course\_code)

staff\_assigned (staff\_id, course\_code)

**Constraints :**

A staff member may only tutor or counsel students who are located in the same region as the staff member.

Students are constrained in the number of courses they can be enrolled in at any one time. Students may not enroll for more than 180 points worth of courses at any one time.

The attribute Credit (of Course) has a value of 15 or 30 points.

A student has at most one enrollment in a course as only current enrollments are recorded.

The student's region and the counselor's region must be the same.

**Assumptions :**

A student is not required to enroll in any courses at registration; enrollment in a course can happen at a later time.

Each student has one counselor, allocated at registration, who supports the student throughout his or her university career.

1. CREATE DATABASE TABLES ACCORDING TO YOUR DATA MODEL. CREATE THE NECESSARY INDEXES IN THE TABLES. BUILD RELATIONSHIPS BETWEEN TABLES USING THE KEYS. USE DATA TYPES FOR TABLE FIELDS. CREATE THE APPROPRIATE CONSTRAINTS IN THE TABLES.

**--CREATE TABLES**

**-- create table 'course'**

CREATE TABLE course(

course\_code INT NOT NULL IDENTITY (1,1)

CONSTRAINT Course\_pk PRIMARY KEY (course\_code),

title VARCHAR (256) NOT NULL,

credit INT NOT NULL

CONSTRAINT check\_credit CHECK (credit=15 OR credit=30)

);

INSERT INTO course

VALUES ('Fine Arts',15),

('German',15),

('Chemistry',30),

('French',30),

('Physics',30),

('History',30),

('Music',30),

('Psychology',30),

('Biology', 15)

**-- create table 'student'**

CREATE TABLE student (

student\_id INT NOT NULL IDENTITY (1,1)

CONSTRAINT student\_pk PRIMARY KEY (student\_id),

first\_name VARCHAR (256),

last\_name VARCHAR (256),

register\_date DATE NOT NULL,

region\_id INT NOT NULL ,

staff\_id INT NOT NULL,

CONSTRAINT fk1\_region\_no FOREIGN KEY (region\_id) REFERENCES region (region\_id) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT fk1\_staff\_no FOREIGN KEY (staff\_id) REFERENCES staff (staff\_id)ON UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO student

VALUES ('Alec','Hunter','12.05.2020',1,7),

('Bronwin', 'Blueberry','12.05.2020',2, 8),

('Charlie', 'Apricot','12.05.2020', 3, 2),

('Ursula', 'Douglas','12.05.2020', 2, 1),

('Zorro', 'Apple', '12.05.2020', 3, 6),

('Debbie', 'Orange','12.05.2020', 1, 9)

-- create table ‘region'

CREATE TABLE region(

region\_id INT PRIMARY KEY NOT NULL,

region\_name VARCHAR (256) NOT NULL

);

INSERT INTO region

VALUES (1,'Wales'),(2, 'Scotland'),(3, 'England'),(4, 'Northern Ireland')

**-- create table ‘staff'**

CREATE TABLE staff (

staff\_id INT NOT NULL IDENTITY(1,1)

CONSTRAINT staff\_pk PRIMARY KEY (staff\_id),

first\_name VARCHAR (256),

last\_name VARCHAR (256),

region\_id INT NOT NULL

FOREIGN KEY (region\_id) REFERENCES region (region\_id)

);

INSERT INTO staff

VALUES ('Neil','Mango',2),

('Harry','Smith',3),

('Yavette','Berry',4),

('Tom','Garden',4),

('Margeret','Nolan',3),

('Kellie', 'Pear',3),

('October', 'Lime',1),

('Ross', 'Island', 2),

('Victor', 'Fig',1)

**-- create table 'enrollment'**

CREATE TABLE enrollment(

student\_id INT NOT NULL,

course\_code INT NOT NULL

Constraint PK\_Enrollment PRIMARY KEY (student\_id , course\_code),

FOREIGN KEY (student\_id) REFERENCES student (student\_id),

FOREIGN KEY (course\_code) REFERENCES course (course\_code));

INSERT INTO enrollment

VALUES (1, 1),

(1, 2),

(2, 1),

(2, 2),

(3, 1),

(3, 2),

(4, 1),

(4, 2)

--- **create table 'staff\_assigned'**

CREATE TABLE staff\_assigned(

staff\_id INT NOT NULL,

course\_code INT NOT NULL,

Constraint PK\_staff\_assigned PRIMARY KEY (staff\_id, course\_code),

FOREIGN KEY (staff\_id) REFERENCES staff (staff\_id),

FOREIGN KEY (course\_code) REFERENCES course (course\_code)

);

INSERT INTO staff\_assigned

VALUES (1,1),

(2,2),

(5,3),

(6,4),

(5,4),

(2,5),

(6,5),

(3,9)

**--CONSTRAINTS**

--1. Students are constrained in the number of courses they can be enrolled in at any one time.

--They may not take courses simultaneously if their combined points total exceeds 180 points.

create function enrollment\_constraint (

@student\_id int,

@course\_code int

)

returns varchar(10)

as

begin

declare @total\_credit int

declare @new\_credit int

declare @result varchar(10)

select @total\_credit = sum(c.credit) from course c join enrollment e on e.course\_code = c.course\_code JOIN student s ON s.student\_id = e.student\_id where e.student\_id = @student\_id

select @new\_credit = credit from course where course\_code = @course\_code

if @total\_credit + @new\_credit > 180

set @result = 'False'

else

set @result = 'True'

return @result

end

ALTER TABLE enrollment

ADD CONSTRAINT check\_enrollment CHECK(dbo.enrollment\_constraint(student\_id,course\_code) = 'True');

--2. The student's region and the counselor's region must be the same.

CREATE FUNCTION check\_region()

RETURNS INT

AS

BEGIN

DECLARE @region int

IF EXISTS(SELECT \*

FROM region r JOIN student s ON r.region\_id=s.region\_id JOIN staff p ON p.region\_id = r.region\_id

WHERE s.region\_id != p.region\_id)

SELECT @region =1 ELSE SELECT @region = 0;

RETURN @region;

END;

ALTER TABLE staff

ADD CONSTRAINT check\_region CHECK(dbo.check\_region() = 0);

1. **ADDITIONAL TASKS**

--1. Test the credit limit constraint.

INSERT INTO course VALUES('History', 15)

The statement has been terminated with error message.

--2. Test that you have correctly defined the constraint for the student counsel's region.

UPDATE staff SET region\_id = 6 WHERE staff\_id=1

The statement has been terminated with error message.

--3. Try to set the credits of the History course to 20. (You should get an error.)

UPDATE course SET credit = 20 WHERE title='History'

The statement has been terminated with error message.

--4. Try to set the credits of the Fine Arts course to 30.(You should get an error.)

UPDATE course SET credit = 30 WHERE title='Fine Arts'

The statement has been terminated with error message.

--5. Debbie Orange wants to enroll in Chemistry instead of German. (You should get an error.)

--6. Try to set Tom Garden as counsel of Alec Hunter (You should get an error.)

UPDATE student SET staff\_id = 4 WHERE student\_id=1

--7. Swap counselors of Ursula Douglas and Bronwin Blueberry.

UPDATE student Set staff\_id = 8 where last\_name ='Douglas'

UPDATE student Set staff\_id = 1 where first\_name ='Blueberry'

--8. Remove a staff member from the staff table.

-- If you get an error, read the error and update the reference rules for the relevant foreign key.

DELETE FROM staff WHERE staff\_id=5

ALTER TABLE student DROP CONSTRAINT fk1\_region\_no

ALTER TABLE student DROP CONSTRAINT fk1\_staff\_no

ALTER TABLE student

ADD CONSTRAINT fk1\_region\_no FOREIGN KEY (region\_id) REFERENCES region (region\_id)

ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT fk1\_staff\_no FOREIGN KEY (staff\_id) REFERENCES staff (staff\_id)

ON UPDATE CASCADE ON DELETE CASCADE