- -- Created by Vertabelo (http://vertabelo.com)

-- Last modification date: 2023-01-24 10:52:25.005

-- tables

-- Table: Class

CREATE TABLE Class (

id int NOT NULL,

name varchar2(50) NOT NULL,

startDate date NOT NULL,

Teacher\_id int NOT NULL,

Course\_id int NOT NULL,

CONSTRAINT Class\_pk PRIMARY KEY (id)

) ;

-- Table: Class\_Student

CREATE TABLE Class\_Student (

Student\_id int NOT NULL,

Class\_id int NOT NULL,

CONSTRAINT Class\_Student\_pk PRIMARY KEY (Student\_id,Class\_id)

) ;

-- Table: Course

CREATE TABLE Course (

id int NOT NULL,

mathLessons int NOT NULL,

description varchar2(300) NOT NULL,

Math\_Topics\_id int NOT NULL,

Level\_Math\_id int NOT NULL,

CONSTRAINT Course\_pk PRIMARY KEY (id)

) ;

-- Table: Level\_Math

CREATE TABLE Level\_Math (

id int NOT NULL,

name varchar2(50) NOT NULL,

levels varchar2(5) NOT NULL,

CONSTRAINT Level\_Math\_pk PRIMARY KEY (id)

) ;

-- Table: Math\_Topics

CREATE TABLE Math\_Topics (

id int NOT NULL,

name varchar2(50) NOT NULL,

Teacher\_id int NOT NULL,

CONSTRAINT Math\_Topics\_pk PRIMARY KEY (id)

) ;

-- Table: Person\_Account

CREATE TABLE Person\_Account (

id int NOT NULL,

login varchar2(50) NOT NULL,

password varchar2(50) NOT NULL,

firstName varchar2(50) NOT NULL,

lastName varchar2(20) NOT NULL,

birthDate date NOT NULL,

email varchar2(50) NOT NULL,

CONSTRAINT Person\_Account\_pk PRIMARY KEY (id)

) ;

-- Table: Student

CREATE TABLE Student (

id int NOT NULL,

Person\_Account\_id int NOT NULL,

CONSTRAINT Student\_pk PRIMARY KEY (id)

) ;

-- Table: Teacher

CREATE TABLE Teacher (

id int NOT NULL,

Person\_Account\_id int NOT NULL,

CONSTRAINT Teacher\_pk PRIMARY KEY (id)

) ;

-- foreign keys

-- Reference: Class\_Course (table: Class)

ALTER TABLE Class ADD CONSTRAINT Class\_Course

FOREIGN KEY (Course\_id)

REFERENCES Course (id);

-- Reference: Class\_Student\_Class (table: Class\_Student)

ALTER TABLE Class\_Student ADD CONSTRAINT Class\_Student\_Class

FOREIGN KEY (Class\_id)

REFERENCES Class (id);

-- Reference: Class\_Student\_Student (table: Class\_Student)

ALTER TABLE Class\_Student ADD CONSTRAINT Class\_Student\_Student

FOREIGN KEY (Student\_id)

REFERENCES Student (id);

-- Reference: Class\_Teacher (table: Class)

ALTER TABLE Class ADD CONSTRAINT Class\_Teacher

FOREIGN KEY (Teacher\_id)

REFERENCES Teacher (id);

-- Reference: Course\_Level\_Math (table: Course)

ALTER TABLE Course ADD CONSTRAINT Course\_Level\_Math

FOREIGN KEY (Level\_Math\_id)

REFERENCES Level\_Math (id);

-- Reference: Course\_Math\_Topics (table: Course)

ALTER TABLE Course ADD CONSTRAINT Course\_Math\_Topics

FOREIGN KEY (Math\_Topics\_id)

REFERENCES Math\_Topics (id);

-- Reference: Math\_Topics\_Teacher (table: Math\_Topics)

ALTER TABLE Math\_Topics ADD CONSTRAINT Math\_Topics\_Teacher

FOREIGN KEY (Teacher\_id)

REFERENCES Teacher (id);

-- Reference: Student\_Person\_Account (table: Student)

ALTER TABLE Student ADD CONSTRAINT Student\_Person\_Account

FOREIGN KEY (Person\_Account\_id)

REFERENCES Person\_Account (id);

-- Reference: Teacher\_Person\_Account (table: Teacher)

ALTER TABLE Teacher ADD CONSTRAINT Teacher\_Person\_Account

FOREIGN KEY (Person\_Account\_id)

REFERENCES Person\_Account (id);

-- End of file.

INSERT INTO Person\_Account(id, login, password, firstName, lastName, birthDate, email) VALUES ( 201, 'Beyzuk', 'shc637gq' , 'Beyzuk', 'Sonmez', TO\_DATE('2003-07-01','YYYY-MM-DD'), 'beyzu7.gmail.com' );

INSERT INTO Person\_Account(id, login, password, firstName, lastName, birthDate, email) VALUES ( 202, 'Arzuk', 'c63gq' , 'Arzu', 'Kilic', TO\_DATE('2002-08-01','YYYY-MM-DD'), 'arzu123.gmail.com' );

INSERT INTO Person\_Account(id, login, password, firstName, lastName, birthDate, email) VALUES ( 203, 'Aduk', 'ac637gq' , 'Adam', 'Levis', TO\_DATE('1980-01-02','YYYY-MM-DD'), 'adam677.gmail.com' );

INSERT INTO Student(id, Person\_Account\_id) VALUES ( 25729, 201);

INSERT INTO Student(id, Person\_Account\_id) VALUES ( 25701, 202);

INSERT INTO Student(id, Person\_Account\_id) VALUES ( 20431, 203);

INSERT INTO Teacher(id, Person\_Account\_id) VALUES ( 3000, 201);

INSERT INTO Teacher(id, Person\_Account\_id) VALUES ( 3001, 202);

INSERT INTO Teacher(id, Person\_Account\_id) VALUES ( 3002, 203);

INSERT INTO Math\_Topics(id, name, Teacher\_id) VALUES ( 133, 'Limit', 3000);

INSERT INTO Math\_Topics(id, name, Teacher\_id) VALUES ( 144, 'Integral', 3001);

INSERT INTO Math\_Topics(id, name, Teacher\_id) VALUES ( 155, 'Logic', 3002);

INSERT INTO Level\_Math(id, name, levels) VALUES ( 123, 'Beginner', 'A');

INSERT INTO Level\_Math(id, name, levels) VALUES ( 124, 'Intermediate', 'B');

INSERT INTO Level\_Math(id, name, levels) VALUES ( 125, 'Advanced', 'C');

INSERT INTO Course(id, mathLessons, description, Math\_Topics\_id, Level\_Math\_id) VALUES ( 765, 30, 'Algebratci equations', 133, 123);

INSERT INTO Course(id, mathLessons, description, Math\_Topics\_id, Level\_Math\_id) VALUES ( 800, 12, 'Calculus 1', 144, 124);

INSERT INTO Course(id, mathLessons, description, Math\_Topics\_id, Level\_Math\_id) VALUES ( 986, 6, 'Discrete Math and logic', 155, 125);

INSERT INTO Class(id, name, startDate,Teacher\_id ,Course\_id) VALUES ( 1100, 'Algebra', TO\_DATE('2021-09-01','YYYY-MM-DD'), 3000, 765);

INSERT INTO Class(id, name, startDate,Teacher\_id ,Course\_id) VALUES ( 1200, 'Calculus', TO\_DATE('2022-03-05','YYYY-MM-DD'), 3001, 800);

INSERT INTO Class(id, name, startDate,Teacher\_id ,Course\_id) VALUES ( 2400, 'Discrete Math', TO\_DATE('2022-04-02','YYYY-MM-DD'), 3002, 986);

INSERT INTO Class\_Student(Student\_id, Class\_id) VALUES ( 25729, 1100);

INSERT INTO Class\_Student(Student\_id, Class\_id) VALUES ( 25701, 1200);

INSERT INTO Class\_Student(Student\_id, Class\_id) VALUES ( 20431, 2400);

DROP TABLE Class\_Student;

DROP TABLE Class;

DROP TABLE Course;

DROP TABLE Level\_Math;

DROP TABLE Math\_Topics;

DROP TABLE Student;

DROP TABLE Teacher;

DROP TABLE Person\_Account;

--TRIGGER1

set serveroutput on;

create or replace trigger numOfMathLessoTooLow

before insert

on Course

for each row

begin

if inserting then

if :new.mathLessons < 6 then

raise\_application\_error(-20100, 'lesson count is too low');

end if;

end if;

end;

DROP TRIGGER numOfMathLessoTooLow;

insert into course values ( 987, 2, 'Elemntary Algebra', 133, 123);

--TRIGGER2

set serveroutput on;

create or replace trigger newstudentmessage

after insert or update

on student

for each row

when (new.id > 0)

begin

dbms\_output.put\_line ('New student details inserted. Welcome to the course!');

end;

INSERT INTO Student(id, Person\_Account\_id) VALUES ( 8, 201);

PROCEDURE 1

-----------------------------

SET serverOutput ON;

CREATE OR REPLACE PROCEDURE addnewlevelmath(mathLevelId INTEGER, mathLevelName VARCHAR2, mathLevelLevels VARCHAR2)

AS

foundLevelMaths INTEGER;

BEGIN

SELECT COUNT(1) INTO foundLevelMaths FROM Level\_Math WHERE id=mathLevelId OR name=mathLevelName OR levels=mathLevelLevels;

IF foundLevelMaths=0 THEN

INSERT INTO Level\_Math (id, name, levels) VALUES (mathLevelId, mathLevelName, mathLevelLevels);

END IF;

END;

Drop procedure addnewlevelmath;

Call addnewlevelmath( 126,'Master', 'd');

PROCEDURE WITH CURSOR

------------------------

SET serverOutput ON;

CREATE OR REPLACE PROCEDURE UpdateNumOfMathLessons(maxNumberMathLessons INTEGER, minNumberMathLessons INTEGER)

AS

id\_in INTEGER; mathLessons\_in INTEGER; description\_in VARCHAR2(50); mathTopicsId\_in INTEGER; levelMathId\_in INTEGER;

newMathLessons INTEGER;

CURSOR numCursor IS SELECT id, mathLessons, description, Math\_Topics\_id, Level\_Math\_id FROM Course;

BEGIN

OPEN numCursor;

LOOP FETCH numCursor INTO id\_in, mathLessons\_in, description\_in, mathTopicsId\_in, levelMathId\_in;

EXIT WHEN numCursor%NOTFOUND;

IF mathLessons\_in < maxNumberMathLessons THEN

newMathLessons := mathLessons\_in + 1;

UPDATE Course SET mathLessons = newMathLessons WHERE id = id\_in;

dbms\_output.put\_line('Course ' || id\_in || ' numof classes increased from ' || mathLessons\_in || ' to ' || newMathLessons);

ELSIF mathLessons\_in > minNumberMathLessons THEN

newMathLessons := mathLessons\_in - 1;

UPDATE Course SET mathLessons = newMathLessons WHERE id = id\_in;

dbms\_output.put\_line('Course ' || id\_in || ' numof classes decreased from ' || mathLessons\_in || ' to ' || newMathLessons);

END IF;

END LOOP;

CLOSE numCursor;

END;

call UpdateNumOfMathLessons(7, 29);

drop procedure UpdateNumOfMathLessons;

Vertabelo

The database keeps data about math classes given by teachers, students, class descriptions, class numbers. One teacher can give multiple classes. The account information for students and teachers should be kept in the database. The level and topic of the course should be described as well.