Stored Procedures - Practice

EduBD

student(**student_id**, first_name, last_name, dob, gender, address, note, *class_id*)

subject(**subject_id**, name, credit, percentage_final_exam)

lecturer(**lecturer_id**, first_name, last_name, dob, gender, address, email)

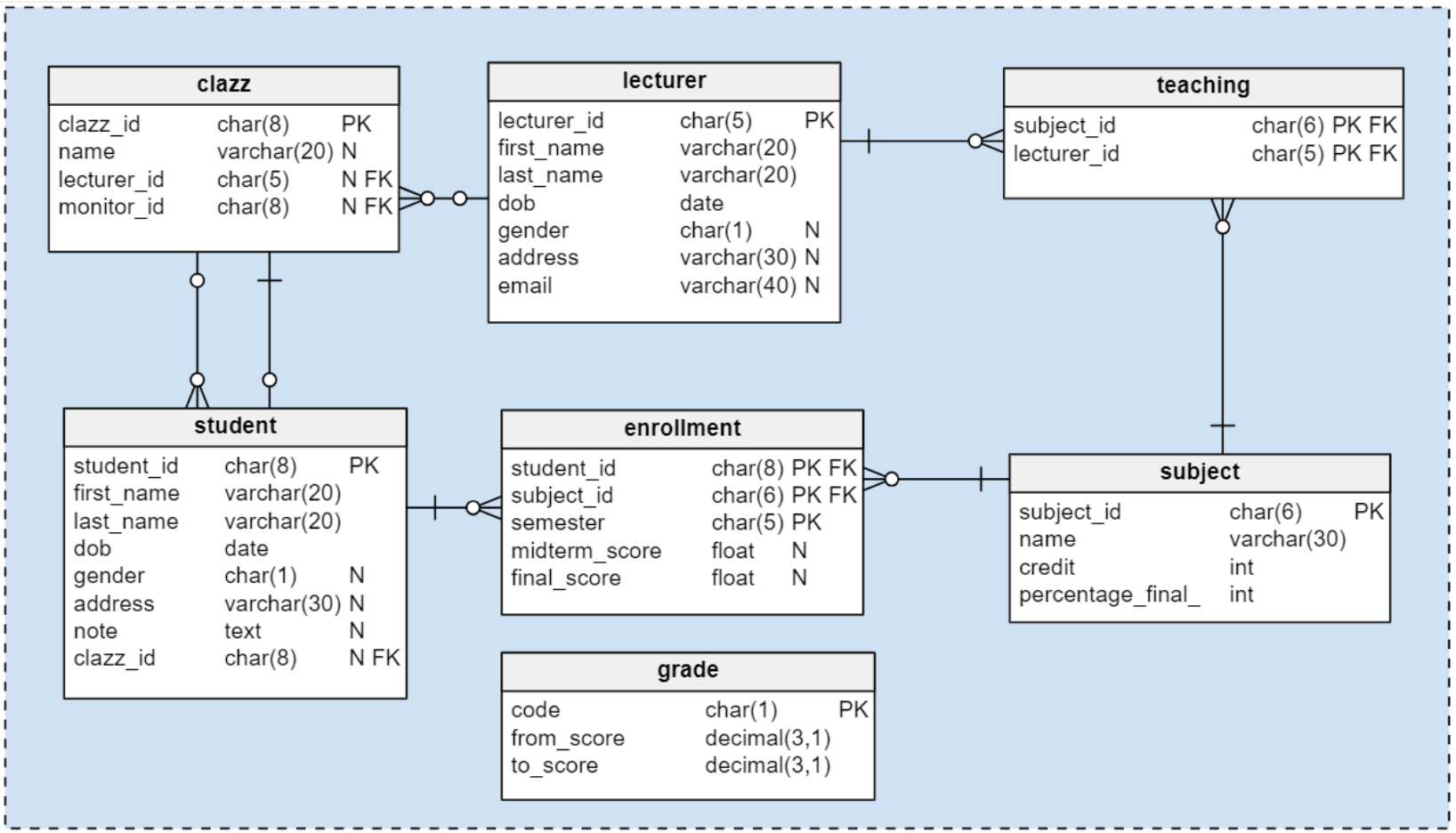
teaching(***subject_id, lecturer_id***)

grade(**code**, fromScore, toScore)

clazz(**clazz_id**, name, *lecturer_id*, *monitor_id*)

enrollment(***student_id, subject_id, semester,*** midterm_score, final_score)

EduBD


clazz
clazz_id char(8) PK
name varchar(20) N
lecturer_id char(5) N FK
monitor_id char(8) N FK
lecturer
lecturer_id char(5) PK
first_name varchar(20)
last_name varchar(20)
dob date
gender char(1) N
address varchar(30) N
email varchar(40) N
teaching
subject_id char(6) PK FK
lecturer_id char(5) PK FK
student
student_id char(8) PK
first_name varchar(20)
last_name varchar(20)
dob date
gender char(1) N
address varchar(30) N
note text N
clazz_id char(8) N FK
enrollment
student_id char(8) PK FK
subject_id char(6) PK FK
semester char(5) PK
midterm_score float N
final_score float N
subject
subject_id char(6) PK
name varchar(30)
credit int
percentage_final_ int
grade
code char(1) PK
from_score decimal(3,1)
to_score decimal(3,1)

Define a function

1) Given a classID, write a function, named : number_of_students, that calculates the number of students in this class.

- Try to call this function from superuser account

Execute a function with normal role

- Define an user role named : ***joe***; pass: ***12345678***
- Login with **joe** account ➔**execute** the function number_of_students(a_class_id)
- Login with superuser => **Grant** execute privilege on this function to joe
- Login with **joe** account => **Re-execute** the function number_of_students(a_class_id)

Exercices

2) Add a new attribute (named: number_students, data type: integer) on clazz table to store number of students in class.

- Define a function (named *update_number_students()*) that computes the number of students for each class and update the correct value for number_students attribute.

- Check values in this attribute before and after calling the function *update_number_students()*

Exercices

3) Create a new table to store GPA (float), CPA (float) of students in each semester

student_results(***student_id, semester***, GPA, CPA)

- Define a function to update GPA, GPA of a student in a semester. Student_id and semester are input variables ot the function

  updateGPA_student(studentid, semester)

- Define a function to update GPA, GPA for all students in the semester indicated by a input variable.

  updateGPA(semester)

- Check whether your functions work correctly or not