DOCUMENTATION

DBS Project
Addition Substitution Window

Team Members
Madhav Madhoosudanan 2020A7PS0014P
Tikesh Vaishnav 2020A7PS0014P

System Requirements

Operating System:

1. Microsoft Windows Server 2012 R2 Standard.
2. Oracle Enterprise Linux 6.5 or higher.

Database:

1. Oracle Text.
2. Oracle XML DB.
3. Oracle JVM.

Memory:

1. 8GB RAM or more.

Disc Space:

1. 100MB for SQL server and an additional 200KB for Database (schema).

ARCHITECTURE & Tech stack

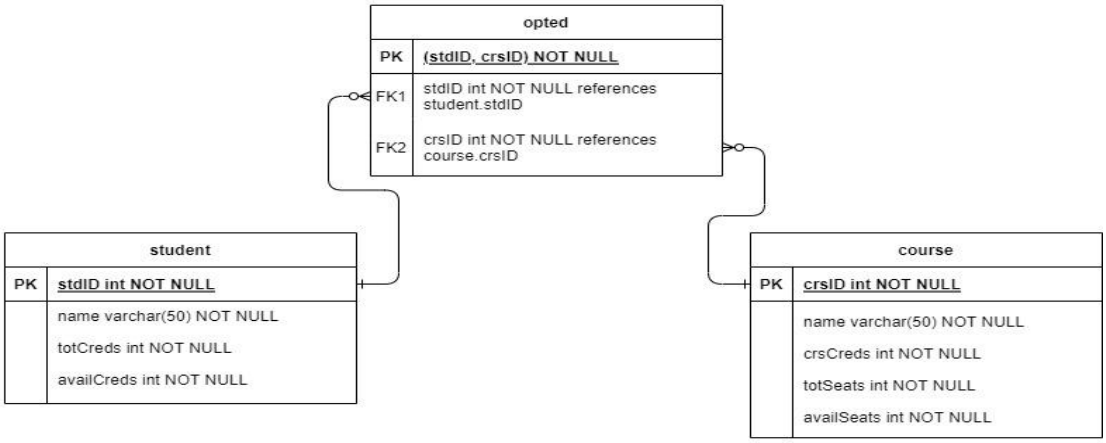
Frontend (Vuejs, html/css/js)

Backend (Go)

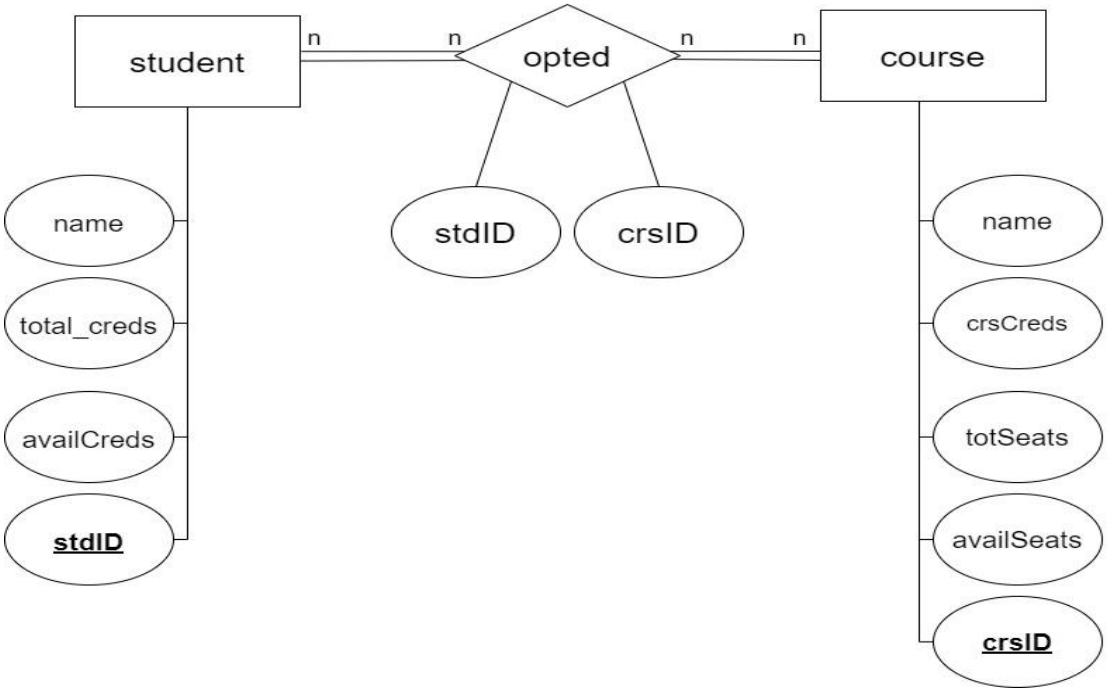
Database (mysql instance, docker container)

ER - Diagram

PHYSICAL ER DIAGRAM



CONCEPTUAL ER DIAGRAM



Normalization

Student (name, stdID , totCreds, availCreds)

Course (name, CrsID, totSeats, availSeats, EnrCreds)

Opted (StdId , CrsId)

↳ for student (A, B, C, D)

FD = { B → A, B → C, B → D }

∴ B^+ BACD

Candidate key = {B} ⟶ BCNF

prime attribute = {B} (LHS is CK)

↳ For Course (A , B , C , D, E)

FD = { B → A, B → C , B → D, B → E }

B^+ = BACDE

CK = { B } ⟶ BCNF

PA = { B } (LHS is CK)

↳ for opted (A, B)

FD = { }

AB+ = AB

CK = {AB}

PA = {A, B} ⟶ BCNF

∴ Normalization is BCNF

and since no multivalued dependency is present,

Normalization in 4th NF

List of Tables required

Student table:

1. stdID = int primary key not null
2. stdName = varchar(50) not null
3. totCreds = int not null
4. availCreds = int not null range[0, total_creds]

Course table:

1. crsID = int primary key not null
2. crsName = varchar(50) not null
3. totalSeats = int not null
4. availSeats = int not null range[0, total_seats]
5. crsCreds = int not null

Opted table:

1. stdID = foreign key not null references student.studID
2. crsID = foreign key not null references course.courseID
3. (stdID, crsID) = primary key not null

Additional Components

Procedures:

1. Addition:
    a. Inputs = int stdID, int crsID.
    b. Checks = available seats in the course, if the student is already enrolled and available credits with the students.
        i. If fails: Output ERRORMessage.
        ii. Else: commit transaction.
    c. Output = Update takes table, student table, course table.
2. Substitution:
    a. Inputs = int stdID, int toldcrsID, int tnewcrsID.
    b. Checks = available seats in the new course, if the student is already enrolled in the new course, available credits can accommodate the difference of credits from both courses.
        i. If fails: Output ERROR Message.
        ii. Else: commit transaction.
    c. Output = Update opted table, student table, course table.
3. Display Student's Courses:
    a. Inputs = int stdID
    b. Output = all courses the student is enrolled in.
4. Display Course details:
    a. Inputs = int crsID
    b. Output = available seats, total seats in course and credits of course.