# 

DAOS Mini Projekt

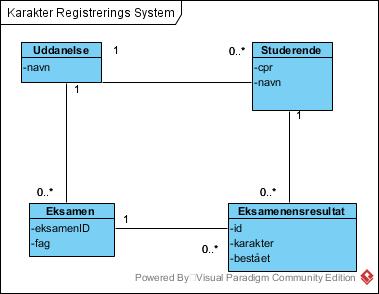
Simon K. Mortensen

Ali Kasim

John G. M. Høeg

# 

# Opgave 1



# 

# Opgave 2

|  |
| --- |
| DROP TABLE IF EXISTS EksamensResultat DROP TABLE IF EXISTS Studerende DROP TABLE IF EXISTS Eksamen DROP TABLE IF EXISTS Uddanelse  create table Uddanelse( navn varchar(30) Primary key )  create table Eksamen( eksamenID int identity(1,1) Primary key, fag varchar(30), Uddannelsesnavn varchar(30) foreign key references Uddanelse(navn) )  create table Studerende( cpr char(10) primary key, navn varchar(50), Udannelsesnavn varchar(30) foreign key references Uddanelse(navn) )  create table EksamensResultat( id int identity(1,1) primary key, karakter int check(karakter in (-3, 0, 2, 4, 7, 10, 12)), bestået bit, EksamensID int foreign key references Eksamen(eksamenID), StuderendeCPR char(10) foreign key references Studerende(cpr) ) |

# Opgave 3

|  |
| --- |
| insert into Uddanelse values ('Datamatikker'), ('Finansøkonom'), ('Økonomi og IT'), ('Jordbrugsteknolog'), ('Softwareudvikling')  insert into Eksamen values ('PRO1', 'Datamatikker'), ('DAOS', 'Datamatikker'), ('PRO2', 'Datamatikker'), ('SU1', 'Datamatikker'), ('SU2', 'Datamatikker'), ('FIT1', 'Datamatikker'), ('FIT2', 'Datamatikker')  insert into Studerende values ('0404801123', 'Simon', 'Datamatikker'), ('0404801125', 'John', 'Datamatikker'), ('0404801127', 'Ali', 'Datamatikker'), ('0404801129', 'Viggo', 'Datamatikker'), ('0404801122', 'Stine', 'Datamatikker'), ('0404801121', 'Mikkel', 'Datamatikker'), ('0404801124', 'Elizabeth', 'Datamatikker')  insert into EksamensResultat values ('0404801123', 10, 1, 2), ('0404801123', 7, 1, 1), ('0404801123', 4, 1, 3), ('0404801123', 2, 1, 6), ('0404801123', 0, 0, 7), ('0404801123', 2, 1, 7), ('0404801127', 2, 1, 1), ('0404801127', 2, 1, 2), ('0404801127', 2, 1, 3), ('0404801127', 2, 1, 4), ('0404801127', 2, 1, 5), ('0404801127', 2, 1, 6), ('0404801127', 2, 1, 7), ('0404801125', 12, 1, 2), ('0404801129', 12, 1, 1), ('0404801129', 12, 1, 2), ('0404801129', 12, 1, 3), ('0404801129', 12, 1, 4), ('0404801129', 12, 1, 5), ('0404801129', 12, 1, 6), ('0404801129', 12, 1, 7), ('0404801122', -3, 0, 2), ('0404801122', 0, 0, 2), ('0404801122', 12, 1, 2), ('0404801121', 2, 1, 2), ('0404801124', 7, 1, 2) |

# Opgave 4

|  |
| --- |
| -- Opgave 4.A select distinct s.navn from Eksamen e, EksamensResultat er, Studerende s where e.eksamenID = er.EksamensID and er.StuderendeCPR = s.cpr and er.karakter = 12  -- Opgave 4.B select distinct s.navn from Eksamen e, EksamensResultat er, Studerende s where e.eksamenID = er.EksamensID and er.StuderendeCPR = s.cpr and er.bestået = 1 except select distinct s.navn from Eksamen e, EksamensResultat er, Studerende s where e.eksamenID = er.EksamensID and er.StuderendeCPR = s.cpr and er.bestået = 0  -- Opgave 4.C select distinct e.fag, avg(er.karakter) as Gennemsnitskarakter from Eksamen e, EksamensResultat er, Studerende s where e.eksamenID = er.EksamensID and er.StuderendeCPR = s.cpr and er.bestået = 1 group by e.fag |

# Opgave 5

|  |
| --- |
| -- Opgave 5 drop proc IF EXISTS StuderendeBeståetEksamer create proc StuderendeBeståetEksamer @CPR char(10) as begin select distinct s.navn, e.fag from Eksamen e, EksamensResultat er, Studerende s where e.eksamenID = er.EksamensID and er.StuderendeCPR = s.cpr and er.bestået = 1 and s.cpr = @CPR end  execute StuderendeBeståetEksamer '0404801127' execute StuderendeBeståetEksamer '0404801123' |

# Opgave 6

|  |
| --- |
| -- Opgave 6  go drop trigger IF EXISTS AlleredeBeståetEksamen go create trigger AlleredeBeståetEksamen on EksamensResultat instead of insert  as declare @EkID int declare @CPR char(10) set @EkID = (select EksamensID from inserted) set @CPR = (select StuderendeCPR from inserted) begin  if not exists  (select er.bestået, er.StuderendeCPR  from EksamensResultat er  where er.bestået = 1  and StuderendeCPR = @CPR  and EksamensID = @EkID  )  begin  insert into EksamensResultat  select StuderendeCPR, karakter, bestået, EksamensID from inserted  end  else   begin  raiserror('Du har allerede bestået dit fag!', 1, 1)  end end  insert into EksamensResultat values ('0404801123', 2, 1, 4) --('0404801125', 12, 1, 2)  select \* from EksamensResultat |

# Opgave 7

|  |
| --- |
| package mini\_projekt;  import java.io.BufferedReader; import java.io.InputStreamReader; import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.SQLException;  public class MainApp {  public static void main(String[] args) {  try {  BufferedReader inLine = new BufferedReader(new InputStreamReader(System.in));  System.out.println("Intast CPR nummer: ");  String CPR = inLine.readLine();  System.out.println("Intast karakter: ");  String karakter = inLine.readLine();  System.out.println("Intast FagID:(1,2,3,4,5,6,7) ");  String fagID = inLine.readLine();  Connection minConnection;  minConnection = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Karakter\_Registrering\_System;user=sa;password=8260Viby");  String sql = "insert into EksamensResultat values(?,?,?,?)";  PreparedStatement prestmt = minConnection.prepareStatement(sql);  prestmt.clearParameters();  prestmt.setString(1, CPR);  prestmt.setInt(2, Integer.parseInt(karakter.trim()));  int bestaaet = 0;  if (Integer.parseInt(karakter.trim()) < 2) {  bestaaet = 0;  } else {  bestaaet = 1;  }  prestmt.setInt(3, bestaaet);  prestmt.setInt(4, Integer.parseInt(fagID.trim()));  prestmt.executeUpdate();  System.out.println("Karakter indsat");   if (prestmt != null)  prestmt.close();  if (minConnection != null)  minConnection.close();  } catch (SQLException e) {  System.out.println("fejl: " + e.getMessage());  System.out.println("fejl: " + e.getErrorCode());  if (e.getErrorCode() == 208) {  System.out.println("Forkert database!");  } else if (e.getErrorCode() == 8152) {  System.out.println("CPR-nummer er for langt.");  } else if (e.getErrorCode() == 547) {  System.out.println("Ugyldig karakter eller fagID.");  }  }   catch (Exception e) {  System.out.println("fejl: " + e.getMessage());  }  }  } |