

4/01/2023

University Database

1.

We have created database tables according to the working weight of the courses and lecturers within the university system. Based on these tables, we choose the instructor who will give the most appropriate lesson on behalf of a course. We keep the relationship of the teachers with the courses and their fields, the number of theses they have done in our data set. In order to find the most competent instructor for a course, we put our theses, which are separated according to their fields, in a ranking, and the instructor who gives the most theses in the field is found to be the competent person for that course.

2.

A

Our goal is to expand the given university model and associate it with tables in the database system and increase the functionality of the model. In addition, we have added functions in our data to find the most competent staff for a field in the Instructors' Dataset and kept a record of them

B

DEPARTMENT, STUDENT, INSTRUCTOR, FACULTY MEMBER (PROFESSOR,
 ASSOCIATE PROFESSOR, ASSISTANT PROFESSOR, INSTRUCTOR,
 RESEARCH ASSISTANT) , SECTION,THESES,
 CURRICULUM,BESTINAREAS

C

DEPARTMENT{ Dname, Dcode, Doffice, Dphone, Dcountry}

COLLEGE{Coffice, Cphone, Cname}

INSTRUCTOR{Inslid, Rank, Ioffice, Iname, Iphone}

THESES{Theseld, TheseLevel, ResearchArea}

STUDENT{ SID,DOB,Sname,Major,Adress,Phone}

SECTION{InsID,SecID,SecNo,Sem,Year,Croom,DaysTime}

Course{CDesc,CCode,CoName,Credits,Level}

Curriculum{CulID}

BestInAreas {Area,DoneRearches,Best_Ins_Area

Faculty Member{Researches}

D

COLLEGE DEAN INSTRUCTOR

COLLEGE ADMINS DEPT

DEPT OFFERS COURSE

DEPT HAS STUDENT

DEPT EMPLOYS INSTRUCTOR

DEPT CHAIR CHAIR

SAVED DEPT HAS CURRICULUM

CURRICULUM HAS COURSEWITHCURRICULUM

BEST_AREAS HAS EXPERT INSTRUCTOR

BEST_AREAS HAS INCLUDES COURSE

COURSE SECS SECTION

SECTION TAKES STUDENT

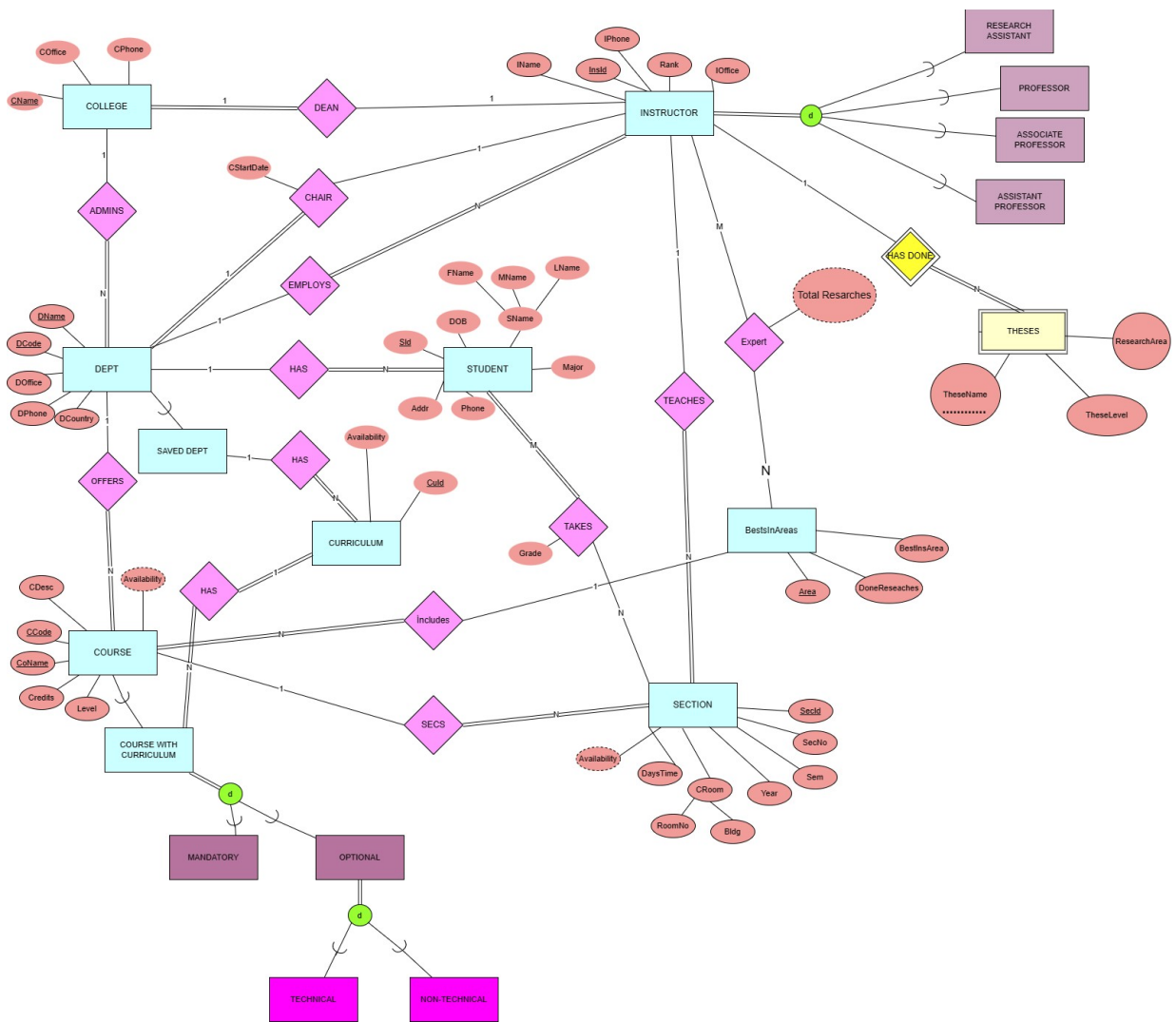
SECTION TEACHES INSTRUCTOR

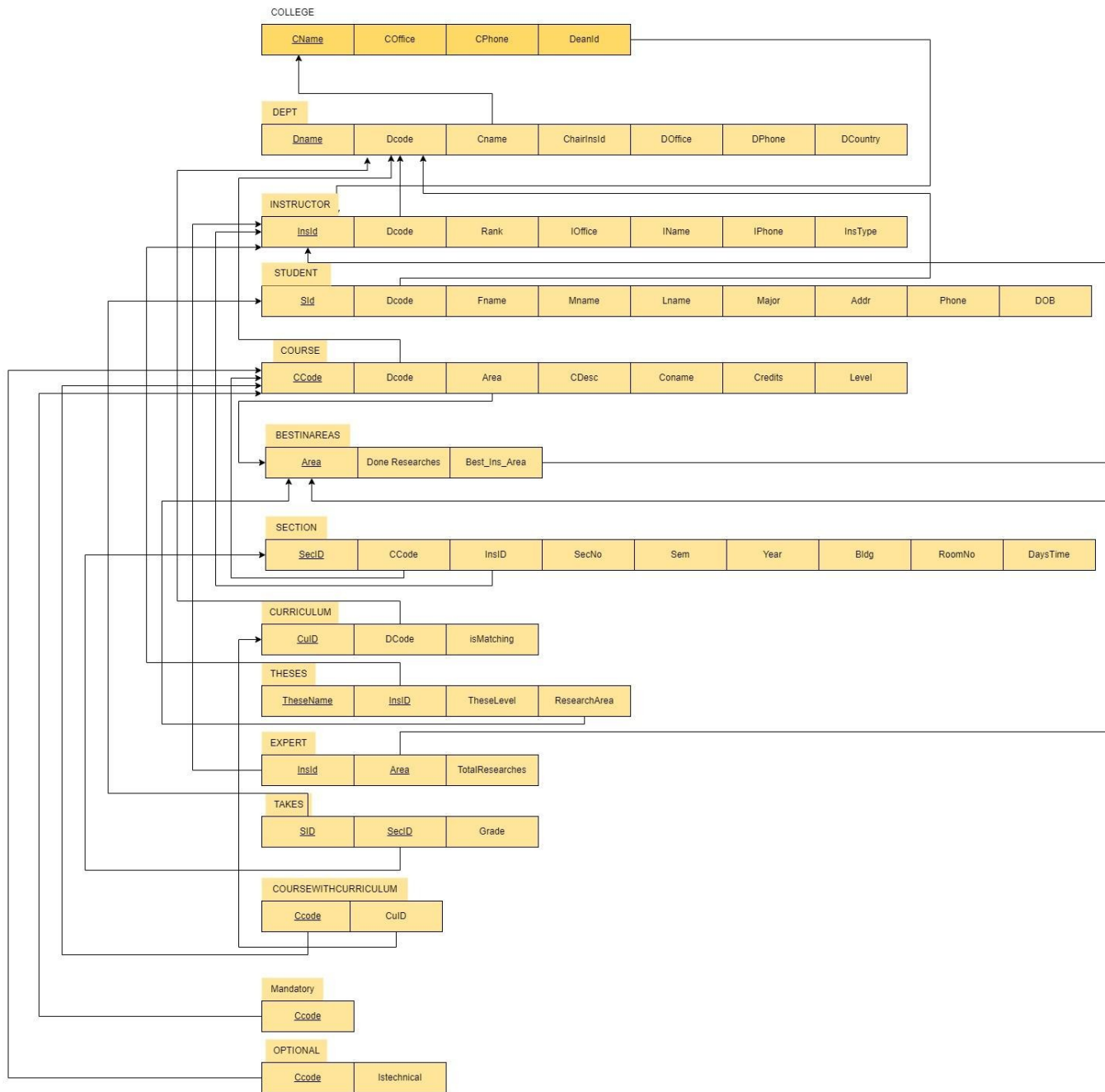
FACULTY MEMBERS HAS DONE THESES

E

- · *College must have at least one Dean from Instructor*
- · *College can admins many Departments. Departments related to one College*
- · *Department can offers many course . Courses related to one Department*
- · *Department can has many Student. Studens can be in one Department*
- · *Department superclass of Saved.*
- · *Saved Dept can has many CoursewithCurriculumS. Each CoursewithCurriculum related to be saved dept.*
- · *Course Secs Sections. Each Section related to one Course.*
- · *Course has Curriculum.*
- · *Students can have many sections. Section must have at least 5 Students..*
- · *Instructor can teach many section but each section related to one instructor*
- · *Instructor has to be chair or faculty member.*
- · *Faculty member can be Professor, Associate Professor, Assistant Professor, Instructor, Research Assistant*
- · *Faculty Member has done many thesis. Thesis related to one Faculty Member*
- · *Chair manage one Department. Department related to Chair*
- · *Curriculum can has many CoursewithCurriculum. CoursewithCurriculum can related many Corruculums.*
- · *CoursewithCurriculum can be Mandotory and Optional.*
- · *Optional can be Technical or Non technical*

ER DÍAGRAM





DATA REQUIREMENTS

Each college has a unique Cname,Coffice and cPhone

Each College has more admins and it is related to exactly one instructor

Each Dept has a unique Dname,Dcode,Doffice,Dphone and Dcountry

Each dept has more employs instructor , offers course and has studens. It is also related to exactly one instructor

Each Course Unique(CoName,Ccode),Cdesc,Credits,level, isMatching. It is also related to exactly one BestInAreas

Each Course has more secs section

Each Section Unique SecID,SecNo,Sem,Year,Croom,DaysTimes,Avaiblity.

Each Section has five or more takes student.It is also related to exactly one Course and instructor

Each Insturctor has a unique InsID,Iname,Rank,Ioffice and iphone and is also related to exactly one employs
dept,Chair dept,Dean College and to at least one BestINAreas,Theses

Each BestInAreas has a unique Area,Done researches,bestInArea

Each BestIn Areas has zero or more Includes Course and teaches instructor

Each Student has a unique Sid,Dob,Fname,Mname,Lname,Major,Phone,Dept.it is related to exactly one Dept and to
at least one Section

Courses with a Curriculum must either be mandatory or must be Optional

Optional courses should either be Technical or non-technical,A technical course must be an optional course

an instructor must either be a research assistant or a professor or an Associate professor or an assistant
professor

a professor must be a Instructor,a Assocaciate proffesor must be a Instructor,a research assistant must be a
Instructor,a Assistant Professor must be a Instructor

LOGICAL DESIGN

ITERATION 1

1) Regular Entity

COLLEGE{Cname, Coffice, CPhone}

INSTRUCTOR{InsId, Rank, IOffice, IName, IPhone}

DEPT{DName, DCode, DOffice, DPhone, DCountry} DName should be unique

STUDENT{SId, DOB, Fname, Mname, Lname, Major, Addr, Phone}

COURSE{CDesc, CCode, CoName, Credits, Level}

BESTINAREAS{Area, Done Researches, Best_Ins_Area}

SECTION{isMatching, DaysTime, RoomNo, Bldg, Year, Sem, SecNo, SecID}

CURRICULUM(isMatching, CuID)

2) Weak Entity

THESES(TheseName, TheseLevel, ResearchArea, InsID)

3) 1:1 RELATION

COLLEGE{CName, Office, CPhone, DeanId} Dean relationship

DEPT{DName,Dcode, DOffice, DPhone, DCountry, CStartDate, ChairId} chair relationship

4) 1:N RELATION

DEPT{DName,Dcode, Cname, ChairInsId, CStartDate, DOffice, DPhone, DCountry} Admins relationship

INSTRUCTOR{InsId, DCode, Rank, IOffice, IName, IPhone} employs relationship

SECTION{isMatching, DaysTime, RoomNo, Bldg, Year, Sem, SecNo, SecID, InsId} teaches

SECTION{isMatching, DaysTime, RoomNo, Bldg, Year, Sem, SecID, SecNo, InsID, CCode} secs relationship

COURSE{CDesc, CCode, DCode, CoName, Credits, Level} offers

COURSE{CDesc, CCode, Area, DCode, CoName, Credits, Level} includes

STUDENT{SId, DCode, DOB, Fname, Mname, Lname, Major, Addr, Phone} has

5) M:N RELATION

EXPERT{InsId, Area, TotalResearches}

TAKES{SID, SecId, Grade}

6-7) —

8)

A)

COURSEWITHCURRICULUM{Ccode}

C)

INSTRUCTOR{InsId, DCode, Rank, IOffice, IName, IPhone, InsType}

D)

DEPT{DName, Dcode, Cname, ChairInsId, CStartDate, Doffice, Dphone, Dcountry, IsSaved}

2. ITERATION

1)--

2)--

3)--

4)--

5) CURRICULUM{DCode, isMatching}

COURSEWITHCURRICULUM{CCode}

6) –

7) –

8.

A)

Mandatory(CCode)

Optional(CCode)

3. ITERATION

1-7)--

8)

C)OPTIONAL {CCode, is technical}

CODE

–CREATIONS–

```
CREATE TABLE COLLEGE(  
  
    CName nvarchar(100) NOT NULL,  
  
    COffice nvarchar(100),  
  
    CPhone nvarchar(13),  
  
    DeanId int,  
  
    PRIMARY KEY(CName)  
  
);
```

```
CREATE TABLE DEPT(  
  
    DName nvarchar(100) UNIQUE,  
  
    Dcode int NOT NULL,  
  
    Cname nvarchar(100) NOT NULL,  
  
    ChairInsId nvarchar(100),  
  
    CStartDate Date,  
  
    DOffice nvarchar(100),  
  
    DPhone nvarchar(13),  
  
    DCountry nvarchar(100),  
  
    IsSaved boolean default 0,  
  
    PRIMARY KEY(Dcode)  
  
);
```

```
CREATE TABLE INSTRUCTOR(  
  
    InsId int NOT NULL,  
  
    Dcode int NOT NULL,  
  
    IOffice nvarchar(100),  
  
    IName nvarchar(100),  
  
    IPhone nvarchar(13),  
  
    InsType nvarchar(100),  
  
    PRIMARY KEY(InsId),  
  
    check ((STRCMP(InsType, 'Research Assistant')=0)  
  
            OR(STRCMP(InsType, 'Professor')=0)  
  
            OR (STRCMP(InsType, 'Associate Professor')=0)  
  
            OR (STRCMP(InsType, 'Assistant Professor')=0))  
  
);
```

```
CREATE TABLE STUDENT(  
  
    SId int,  
  
    Dcode int NOT NULL,  
  
    Fname nvarchar(100),  
  
    Mname nvarchar(100),  
  
    Lname nvarchar(100),  
  
    Major nvarchar(100),  
  
    Addr nvarchar(100),
```

```
Phone nvarchar(13),

DOB DATE,

PRIMARY KEY(SId)

);

CREATE TABLE BESTINAREAS(

Areaa nvarchar(100) NOT NULL,

DoneResearches int NOT NULL default 0,

Best_Ins_Area int,

PRIMARY KEY(Areaa)

);

CREATE TABLE COURSE(

Ccode int NOT NULL,

Dcode int NOT NULL,

CArea nvarchar(100) NOT NULL,

CDesc nvarchar(100),

CoName nvarchar(100),

Credits nvarchar(100),

CLevel nvarchar(100),

IsMatching boolean default true NOT NULL,

PRIMARY KEY(Ccode)

);
```

CREATE TABLE SECTION(

SecID int NOT NULL,

CCode int NOT NULL,

InsID int NOT NULL,

SecNo int,

Sem nvarchar(100),

Yearr int,

Bldg nvarchar(100),

RoomNo nvarchar(100),

DaysTime date,

PRIMARY KEY(SecID),

check ((STRCMP(Sem,'Güz')=0)

OR(STRCMP(Sem,'Bahar')=0))

);

CREATE TABLE CURRICULUM(

CulID int NOT NULL,

DCode int NOT NULL,

IsMatching boolean default true NOT NULL,

PRIMARY KEY(CuID)

);

CREATE TABLE THESES(

TheseName nvarchar(100) NOT NULL,

InsID int NOT NULL,

TheseLevel nvarchar(100),

ResearchArea nvarchar(100) NOT NULL,

PRIMARY KEY(TheseName,InsID),

check ((STRCMP(TheseLevel,'Lisans')=0)

OR(STRCMP(TheseLevel,'Yüksek Lisans')=0)

OR(STRCMP(TheseLevel,'Doktora')=0))

);

CREATE TABLE EXPERT(

InsID int NOT NULL,

Areaa nvarchar(100) NOT NULL,

TotalResearches int NOT NULL default 0,

PRIMARY KEY(InsID,Areaa)

);

CREATE TABLE TAKES(

SID int NOT NULL,

SecID int NOT NULL,

GRADE nvarchar(100),

PRIMARY KEY(SID,SecID)

);

CREATE TABLE COURSEWITHCURRICULUM(

CCode int NOT NULL,

CuID int NOT NULL,

PRIMARY KEY(CCode)

);

CREATE TABLE MANDATORY(

CCode int Primary key

);

CREATE TABLE OPTIONAL(

CCode int Primary key,

Istechnical boolean default true NOT NULL,

FOREIGN KEY (CCode) REFERENCES COURSEWITHCURRICULUM(CCode)

);

CREATE TABLE log(

Message varchar(40)

);

ALTER TABLE COLLEGE

ADD FOREIGN KEY

(DeanId) REFERENCES INSTRUCTOR(InsId);

ALTER TABLE DEPT

ADD FOREIGN KEY

(Cname) REFERENCES COLLEGE(CName);

ALTER TABLE INSTRUCTOR

ADD FOREIGN KEY

(Dcode) REFERENCES DEPT(Dcode);

ALTER TABLE STUDENT

ADD FOREIGN KEY

(Dcode) REFERENCES DEPT(Dcode);

ALTER TABLE COURSE

ADD FOREIGN KEY

(Dcode) REFERENCES DEPT(Dcode),

ADD FOREIGN KEY

(CArea) REFERENCES BESTINAREAS(Areaa);

ALTER TABLE BESTINAREAS

ADD FOREIGN KEY

(Best_Ins_Area) REFERENCES INSTRUCTOR(InsId);

ALTER TABLE SECTION

ADD FOREIGN KEY

(CCode) REFERENCES COURSE(Ccode),

ADD FOREIGN KEY

(InsID) REFERENCES INSTRUCTOR(InsId);

ALTER TABLE CURRICULUM

ADD FOREIGN KEY

(DCode) REFERENCES DEPT(Dcode);

ALTER TABLE THESES

ADD FOREIGN KEY

(InsID) REFERENCES INSTRUCTOR(InsId),

ADD FOREIGN KEY

(ResearchArea) REFERENCES BESTINAREAS(Areaa);

ALTER TABLE EXPERT

ADD FOREIGN KEY

(InsID) REFERENCES INSTRUCTOR(InsId),

ADD FOREIGN KEY

(Areaa) REFERENCES BESTINAREAS(Areaa);

ALTER TABLE TAKES

ADD FOREIGN KEY

(SID) REFERENCES STUDENT(SId),

ADD FOREIGN KEY

(SecID) REFERENCES SECTION(SecID);

ALTER TABLE COURSEWITHCURRICULUM

ADD FOREIGN KEY

(CCode) REFERENCES COURSE(CCode),

ADD FOREIGN KEY

(CuID) REFERENCES CURRICULUM(CuID);

ALTER TABLE MANDATORY

ADD FOREIGN KEY

(CCode) REFERENCES COURSEWITHCURRICULUM(CCode);

ALTER TABLE OPTIONAL

ADD FOREIGN KEY

(CCode) REFERENCES COURSEWITHCURRICULUM(CCode);

INSERTIONS

start transaction ;

SET foreign_key_checks = 0;

INSERT INTO college (CName, COffice,CPhone,DeanId)

VALUES ('Fen Edebiyat Ve Mühendislik Fakültesi', 'İzmir', '+905365622933',1),

('Fen Ve Mühendislik Fakültesi', 'Antalya', '+905336180073',4);

INSERT INTO instructor (InsId, Dcode,IOffice,IName,IPhone,InsType)

VALUES (1, 10102,'Bornova','Necdet', '+905435434567', 'Professor'),

(2,10101,'Mavi Sehir','Osman Unalir', '+905126718161', 'Professor'),

(3,100,'Tepecik','Ahmet Bırhani', '+905454558747', 'Professor'),

(4,100,'Kızılcık','Özkan Özlenen', '+905237659322', 'Professor'),

(5,10101,'Bornova','Mustafa Kaya', '+905237646212', 'Research Assistant'),

(6,10101,'Bornova','Fatma Yaman', '+905416246376', 'Associate Professor'),

(7,10101,'Bornova','Nazan Nazli', '+905234219542', 'Professor'),

(8,100,'Konyaalti','Hamdi Özmen', '+90552345624', 'Professor'),

(9,100,'Konyaalti','Samet Yazici','+905456754298','Assistant Professor');

INSERT INTO dept (Dname, Dcode, Cname, ChairInsId, CStartDate, DOffice, DPhone, DCountry, IsSaved)

VALUES ('Bilgisayar Mühendisliği', 10101, 'Fen Edebiyat Ve Mühendislik Fakültesi', 2, '2022-11-29', 'Bornova', '+905555555555', 'Türkiye', true),

('Felsefe', 10102, 'Fen Edebiyat Ve Mühendislik Fakültesi', 1, '2020-05-29', 'Evka 3', '+905314323232', 'Türkiye', 0),

('Yazılım Mühendisliği', 100, 'Fen Ve Mühendislik Fakültesi', 3, '2021-11-10', 'Atatürk Mahallesi', '+905314323232', 'İtalya', 0);

INSERT INTO Student (SId, Dcode, Fname, Mname, Lname, Major, Addr, Phone, DOB)

VALUES (265, 10101, 'Osman', '', 'OKSUZ', 'Bilgisayar Mühendisi', 'İzmir Mevlana', '+905365622933', '1999-07-29'),

(439, 10101, 'Aleyna', 'Tuğçe', 'Güzel', 'Bilgisayar Mühendisi', 'İzmir Erzene', '+905454556772', '2002-04-25'),

(15, 10101, 'Nehir', '', 'DEMİR', 'Bilgisayar Mühendisi', 'İzmir Mevlana', '+905987593578', '2001-07-29'),

(20, 10101, 'Nisan', '', 'ÇELİK', 'Bilgisayar Mühendisi', 'İzmir Mevlana', '+905748561039', '2001-07-29'),

(25, 10101, 'Derya', '', 'TAŞ', 'Bilgisayar Mühendisi', 'İzmir Mevlana', '+905575813194', '2002-07-29'),

(30, 10102, 'Berrak', '', 'BAKIR', 'Filozof', 'İzmir Mevlana', '+905145443841', '2003-07-29'),

(35, 10102, 'Su', '', 'KAYA', 'Filozof', 'İzmir Mevlana', '+905800758901', '2002-07-29'),

(40, 10102, 'Yağmur', '', 'ADIGÜZEL', 'Filozof', 'İzmir Mevlana', '+905559352090', '2005-07-29'),

(45, 100, 'Duru', '', 'BOLUĞUR', 'Yazılım Mühendisi', 'İtalya Atatürk Mah', '+905966330683', '2003-07-29'),

(50, 100, 'Gamze', '', 'ERBİL', 'Yazılım Mühendisi', 'İtalya Atatürk Mah', '+905436805672', '2004-07-29'),

(55, 100, 'Bahar', '', 'TATLI', 'Yazılım Mühendisi', 'İtalya Atatürk Mah', '+905404593224', '2004-07-29'),

(60, 100, 'Eylül', '', 'İNCİ', 'Yazılım Mühendisi', 'İtalya Atatürk Mah', '+905935199210', '2003-07-29')

;

INSERT INTO bestinareas (Areaa)

VALUES ('Yapay Zeka'),

('Elektrik'),

('Algoritma'),

('Varoluş Problemi'),

('Felsefe Tarihi'),

('Gömülü Sistemler');

INSERT INTO course(Ccode,Dcode,CArea,CDesc,CoName,Credits,CLevel)

VALUES (1,10101,'Yapay Zeka','Yapay Zeka Alanına Popüler Bilim Açısıyla Yaklaşım','Yapay Zekaya Giriş','Beş','Lisans'),

(2,10101,'Algoritma','Algoritma Analizi','Algoritma Analizi','Dört','Lisans'),

(3,10101,'Elektrik','Elektriğin 100 Yıl Önceki Kullanım Yerleri','Elektriğe Giriş','Beş','Lisans'),

(4,100,'Yapay Zeka','Yazılım Mühendisliğinde Yapay Zeka Nasıl Etki Gösteriyor','Yapay Zekanın Etkisi','Üç','Lisans'),

(5,100,'Algoritma','İleri Seviye Algoritmalar Ve Efektif Çözümleri','Algorithm Analysis','Altı','Lisans'),

(6,10102,'Felsefe Tarihi','Felsefenin Tarihsel Gelişimi','Felsefe Tarihi','On İki','Lisans'),

(7,10102,'Varoluş Problemi','Varoluş Yüzünden Yaşanan Problemler','Varoluşsal Sıkıntılar','Yirmi','Lisans'),

(8,10101,'Gömülü Sistemler','Gömülü ve Gerçek Zamanlı Sistemler Tasarımı','Gömülü ve Gerçek Zamanlı Sistemler','Beş','Lisans'),

(9,10101,'Elektrik','Elektrik Devreleri Tasarımı','Elektrik Devreleri','Beş','Lisans'),

(10,100,'Gömülü Sistemler','Gömülü ve Gerçek Zamanlı Sistemler Tasarımı','Gömülü ve Gerçek Zamanlı Sistemler','Beş','Lisans');

INSERT INTO curriculum(CulID,Dcode)

VALUES (1,10101),

(2,100),

(3,10102);

INSERT INTO theses(TheseName,InsID,TheseLevel,ResearchArea)

VALUES ('Elektriğin Yönetime Etkisi', 1, 'Doktora','Elektrik'),

('Yapay Zeka İle Arı Kovanı Modellemesi', 2, 'Lisans','Yapay Zeka'),

('Yapay Zeka İle Şifre Kırma', 2, 'Yüksek Lisans','Yapay Zeka'),

('Varoluşsal Problemlerin Yıkımı', 2, 'Doktora','Varoluş Problemi'),

('Felsefe Tarihini Özümsemenin Açtığı Zihin Karışıklığı', 2, 'Lisans','Felsefe Tarihi'),

('IoT alanındaki güncel gelişmeler', 2, 'Doktora','Gömülü Sistemler'),

('Yazılımın Durdurulamaz Yükselişindeki Faktörlerin Antoloji İle Çözümlemesi', 3, 'Doktora','Algoritma'),

('Yapay Zeka İle Geleceğin Tahminlenmesi', 3, 'Doktora','Yapay Zeka'),

('Algoritma Analiz Yöntemleri', 3, 'Doktora','Algoritma'),

('Alternatif Akım Kaynaklarında Verimlilik', 4, 'Yüksek Lisans','Elektrik'),

('Görüntü İşleme Algoritmaları', 5, 'Yüksek Lisans','Algoritma'),

('Görüntü Analizi Algoritmaları', 5, 'Doktora','Algoritma'),

('Sivilarda İletkenliği Artıran Durumlar', 6, 'Doktora','Elektrik'),

('Sigorta Sistemlerinin Geleceği', 6, 'Yüksek Lisans','Elektrik'),

('Gömülü Sistemlerde Verimlilik', 7, 'Yüksek Lisans','Gömülü Sistemler'),

('Entegre Devrelerde Sıcaklığı Kontrol Etme Yöntemleri ', 7, 'Doktora','Gömülü Sistemler'),

('Kuantum Bilgisayar Algoritmaları', 8, 'Doktora','Algoritma'),

('Gerçek Zamanlı Sistemler ve Otomasyon', 8, 'Doktora','Gömülü Sistemler'),

('Yapay Zeka ve Veri Bilimi', 9, 'Doktora','Yapay Zeka'),

('Yapay Zeka ile Yüz Tanıma Sistemleri', 9, 'Yüksek Lisans','Yapay Zeka');

INSERT INTO coursewithcurriculum(CCode,CulD)

Values (1,1),

(2,1),

(3,1),

(8,1),

(9,1);

INSERT INTO section(SecID,CCode,InsID,SecNo,Sem,Yearr,Bldg,RoomNo,DaysTime)

VALUES (1,1,2,150,'Bahar',2023,'Ana Bina','Amfi 5','2023-01-04 10:30:00'),

(2,2,2,200,'Güz',2022,'Ana Bina','Sinan Yılmaz Dersliği','2023-02-04 08:00:00'),

(3,3,2,250,'Bahar',2023,'Ana Bina','Elektrik Labı','2023-02-04 10:00:00'),

(4,4,3,300,'Güz',2022,'Ana Bina','Derslik 10','2023-02-04 15:00:00'),

(5,5,3,350,'Bahar',2023,'Ana Bina','Derslik 10','2023-03-04 18:00:00'),

(6,6,2,400,'Güz',2020,'Ana Bina','Derslik 11','2023-03-04 18:00:00'),

(7,8,2,450,'Güz',2020,'Ana Bina','Derslik 12','2023-03-05 14:00:00'),

(8,8,2,451,'Güz',2020,'Ana Bina','Derslik 13','2023-04-05 18:00:00'),

(9,8,2,452,'Güz',2020,'Ana Bina','Derslik 14','2023-04-06 18:00:00'),

(10,8,2,453,'Güz',2020,'Ana Bina','Derslik 15','2023-04-07 18:00:00'),

(11,9,2,500,'Bahar',2023,'Ana Bina','Amfi 3','2023-01-08 10:30:00'),

(12,10,2,550,'Bahar',2023,'Ana Bina','Amfi 4','2023-01-09 10:30:00'),

(13,10,2,551,'Güz',2023,'Ana Bina','Derslik 7','2023-01-10 10:30:00')

;

INSERT INTO mandatory(CCode)

Values (1),

(2),

(3);

INSERT INTO optional(CCode,Istechnical)

Values

(8,0),

(9,1);

INSERT INTO takes(SID, SecID, GRADE)

Values (265,1,70),

(265,2,40),

(265,3,80),

(265,7,55),

(265,8,100),

(439,1,40),

(439,2,80),

(439,3,70),

(439,6,50),

(439,9,65),

(15,1,40),

(15,2,95),

(15,7,73),

(15,8,60),

(20,2,33),

(20,3,80),

(20,8,65),

(20,10,80),

(25,1,60),

(25,2,90),

(25,7,45),

(25,10,65),

(30,3,50),

(30,6,55),

(30,7,42),

(30,10,36),

(35,2,62),

(35,3,23),

(35,8,68),

(35,9,82)

;

SET foreign_key_checks = 1;

rollback;

TRIGGERS

DELIMITER \$\$

CREATE TRIGGER optional_course BEFORE INSERT

ON optional

FOR EACH ROW

BEGIN

IF EXISTS(

*SELECT **

FROM mandatory

WHERE NEW.CCode = mandatory.CCode

)

THEN

*SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'INSERT WARNING - COURSE EXISTS IN
MANDATORY TABLE';*

```
END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER log_update

BEFORE UPDATE

ON log FOR EACH ROW

BEGIN

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'Log tablosu update edilemez.';

END$$

DELIMITER ;

/*drop trigger log_update;*/

/* drop trigger Section_sayi_kontrol_del*/

DELIMITER $$

CREATE TRIGGER Section_sayi_kontrol_del BEFORE DELETE

ON TAKES

FOR EACH ROW

BEGIN

    IF EXISTS(

        SELECT COUNT(*)
```

```

FROM TAKES

GROUP BY TAKES.SecId

HAVING COUNT(*) = 5

)

THEN

    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = '5 ten az öğrenci olan section olamaz.';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER mandatory_course BEFORE INSERT

ON mandatory

FOR EACH ROW

BEGIN

    IF EXISTS(

        SELECT *

        FROM optional

        WHERE NEW.CCode = optional.CCode

    )

    THEN

        SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'INSERT WARNING - COURSE EXISTS IN
OPTIONAL TABLE';

    END IF;

```

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE TRIGGER expertis BEFORE INSERT

ON theses

FOR EACH ROW

BEGIN

IF EXISTS(

SELECT *

FROM expert

WHERE NEW.InsId = expert.InsID AND new.ResearchArea = expert.Areaa

)

THEN

update expert

set TotalResearches=TotalResearches+1

where NEW.InsId = expert.InsID AND new.ResearchArea = expert.Areaa;

ELSE

Insert INTO expert(InsID,Areaa,TotalResearches)

VALUES(NEW.InsId,new.ResearchArea,1);

END IF;

END\$\$

DELIMITER ;

/*drop trigger IsMatched;*/

DELIMITER \$\$

CREATE TRIGGER IsMatched BEFORE INSERT

ON section

FOR EACH ROW

BEGIN

IF EXISTS(

SELECT *

FROM course

WHERE NEW.CCode = course.Ccode AND course.IsMatching = true

)

THEN

SELECT CArea INTO @Areaa FROM course where new.Ccode=course.Ccode;

SELECT TotalResearches INTO @enyuksek FROM expert where

STRCMP(@Areaa,expert.Areaa)=0 order by TotalResearches Desc Limit 1 ;

**SELECT TotalResearches INTO @ohocanin FROM expert where STRCMP(@Areaa,expert.Areaa)=0 and
expert.InsID=new.InsID;**

IF @enyuksek!=@ohocanin

THEN

```

        update course

        set course.IsMatching=false

where NEW.CCode = course.Ccode;

IF EXISTS(

        SELECT *

        FROM coursewithcurriculum as c

        WHERE NEW.CCode = c.Ccode

)

THEN

        SELECT CuID INTO @cuid from coursewithcurriculum where
NEW.CCode = coursewithcurriculum.Ccode;

        update curriculum

        set curriculum.IsMatching=false

        where curriculum.CuID = @cuid;

END IF;

END IF;

END$$

DELIMITER ;

/*drop trigger log_college_insert;*/

DELIMITER //

CREATE TRIGGER log_college_insert AFTER INSERT ON college

```



```
FOR EACH ROW BEGIN
```

```
    INSERT INTO log (message) VALUES ("Database'e yeni college eklendi.");
```

```
END;
```

```
//
```

```
DELIMITER ;
```

3 DELETE 3 INSERT 3 UPDATE

```
SET SQL_SAFE_UPDATES = 0;

update log

set message='lğpasdskad'

where Message="Database'e yeni college eklendi.";

SET SQL_SAFE_UPDATES = 1;


INSERT INTO Student
(SId,Dcode,Fname,Mname,Lname,Major,Addr,Phone,DOB)

VALUES (123123,10101,'Süleyman','','OKSUZ','Bilgisayar
Mühendisi','İzmir Mevlana','+905428459303','2003-07-29');

update student

set Mname='Yaşar'

where SId="123123";

delete from student

where sId=123123;
```

```

delete from takes

where SID=20 and SecID=3;

update takes

set grade=100

where SId=265 and SecID=7;

INSERT INTO takes (SID, SecID, GRADE)

Values (15,3,100);

INSERT INTO bestinareas (Areaa)

VALUES ('Kuantım Bilgisayarları');

update bestinareas

set Areaa='Kuantum Bilgisayarları '

where Areaa='Kuantım Bilgisayarları';

delete from bestinareas

where Areaa='Kuantum Bilgisayarları '

```

SELECTION 1

```

select * from student

where major='Filozof';

select * from takes

where grade>80;

select * from expert

where TotalResearches>1;

```

```
select * from college;  
  
select * from mandatory;
```

SELECTION 2

```
select *  
  
from student, dept  
  
where dept.Cname='Fen Edebiyat Ve Mühendislik Fakültesi'  
  
and dept.dcode=student.dcode;
```

```
select takes.SecID, Grade  
  
from student, takes  
  
where student.Fname='Osman'  
  
and student.SID=takes.SID;
```

```
select *  
  
from student, dept  
  
where dept.Dname='Bilgisayar Mühendisliği'  
  
and dept.dcode=student.dcode;
```

```
select *  
  
from course, section  
  
where course.CArea='Yapay Zeka'  
  
and course.Ccode=section.Ccode;
```

```
select Dept.Dname,count(*)  
  
from dept,student  
  
where dept.Dcode=student.Dcode  
  
group by dept.Dname;
```

SELECTION 2

```
select student.Fname, student.phone  
  
from college,dept,student  
  
where college.DeanId=1  
  
and college.Cname=dept.cname  
  
and student.Dcode=dept.Dcode;  
  
select Iname  
  
from dept,instructor,course  
  
where course.cArea='Yapay Zeka'  
  
and course.Dcode=dept.Dcode  
  
and instructor.InsId=dept.chairInsId;  
  
select Iname  
  
from dept,instructor,college  
  
where dept.Dname='Bilgisayar Mühendisliği'  
  
and Dept.Cname=college.Cname  
  
and instructor.InsId=college.DeanId;
```

```

select co.coName

from dept d, curriculum cu, course co

where d.Dname='Bilgisayar Mühendisliği'

and d.Dcode=cu.Dcode

and co.Dcode=cu.Dcode;

select distinct Iname

from coursewithcurriculum,section,instructor

where coursewithcurriculum.Ccode=section.Ccode

and section.InsId=instructor.InsId

```

Crit 5 Select

1.

```

select course.ccode

from section,course

where section.ccode=course.ccode

group by section.CCode

having count(*)>2

order by Course.Ccode;

```

2.

```

select distinct Iname

from instructor

```

```

where instructor.InsId in(select a.InsId from(select InsId, count(*)
                                from section

                                group by CCode, InsId

                                having count(*)>3) as a )

order by Iname ;

```

3

```

select distinct course.Coname, instructor.Iname

from theses,section,course,instructor

where section.InsId=theses.InsId

and theses.Researcharea = 'Yapay Zeka'

and theses.InsID=instructor.InsId

and course.ccode=section.CCode

order by instructor.Iname;

```

4

```

select concat(Student.Fname,' ',Student.Lname) as 'İsim Soyisim',
        avg(grade)

from student,takes

where student.SId=takes.SID

group by student.SId

order by avg(grade) desc;

```

5

```
select Yearr,sem,count(*)  
  
from section  
  
group by Yearr,sem;
```