

**ANKARA UNIVERSITY**  
**COMPUTER ENGINEERING**  
**COM2058/COM258**  
**MIDTERM**  
**(120 min.)**

You need to place your answers into a single pdf file and upload them. Make sure your answers are readable. Name your file as StudentNumber.pdf. Read the questions carefully.



ANKARA ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ



**SINAVLAR VE ÖDEVLER İÇİN ŞEREF SÖZÜ**

Bir Ankara Üniversitesi öğrencisi olarak;

- -Bu ödevde/sınavda yardım almadığımı ya da hiç kimseye yardım etmediğimi,
- -Başkasına ait olan bir çalışmayı kendi çalışmam olarak sunmadığımı,
- -Sınav/ödev sorularının çözümü için hiç kimseden (öğrenci, öğretim üyesi ya da arkadaş) yardım istemediğimi,
- -Problemin çözümünü bulmak için interneti ya da çevrimiçi ya da basılı herhangi bir belgeyi kullanmadığımı beyan ederim.

Yukarıdaki ifadelere uymadığımın tespit edilmesi durumunda sınavdan/ödevden sıfır alacağımı ve hakkımda **Ankara Üniversitesi Öğrenci Disiplin Yönetmeliği** çerçevesinde soruşturma açılacağını biliyorum.

**HONOR CODE FOR EXAMS and ASSIGMENTS**

As an Ankara University student, I agree that;

- I have neither given nor received unauthorized assistance on this exam or assignment.
- I have not represented the work of another as my work.
- I have not asked someone else (student, teacher, and friends) to help with this assignment or exam questions.
- I have not used the internet or any online or printed document to find problem solutions

I understand that failure to comply with the statements above will result in receiving a zero from this exam/assignment and being reported for academic dishonesty by the **disciplinary policies of Ankara University**.

1. (30 p) Assume that the following data definition statements for defining the COMPANY schema (CompanySchema.pdf) are given.

```
CREATE TABLE EMPLOYEE (Fname VARCHAR(15), NOT NULL, Minit CHAR, Lname VARCHAR(15) NOT NULL, Ssn CHAR(9) NOT NULL, Bdate DATE, Address VARCHAR(30), Sex CHAR, Salary DOUBLE, Super_ssn CHAR (9), Dno INT NOT NULL DEFAULT 1,
PRIMARY KEY (Ssn),
FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
ON DELETE SET NULL ON UPDATE CASCADE,
FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE DEPARTMENT (Dname VARCHAR(15) NOT NULL, Dnumber INT NOT NULL, Mgr_ssn CHAR(9) NOT NULL DEFAULT '888665555', Mgr_start_date DATE,
PRIMARY KEY (Dnumber),
FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE DEPT_LOCATIONS (Dnumber INT NOT NULL, Dlocation VARCHAR(15) NOT NULL,
PRIMARY KEY (Dnumber, Dlocation),
FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE PROJECT (Pname VARCHAR(15) NOT NULL, Pnumber INT NOT NULL, Plocation VARCHAR(15), Dnum INT NOT NULL DEFAULT 1,
PRIMARY KEY (Pnumber),
FOREIGN KEY (Dnum) REFERENCES DEPARTMENT (Dnumber)
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE WORKS_ON (Essn CHAR(9) NOT NULL, Pno INT NOT NULL DEFAULT 20, Hours DOUBLE NOT NULL,
PRIMARY KEY (Essn, Pno),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn)
ON DELETE CASCADE ON UPDATE CASCADE);
FOREIGN KEY (Pno) REFERENCES EMPLOYEE(Pnumber)
ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE DEPENDENT (Essn CHAR(9) NOT NULL,
Dependent_name VARCHAR(15) NOT NULL, Sex CHAR, Bdate DATE, Relationship VARCHAR(8),
PRIMARY KEY (Essn, Dependent_Name),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(ssn)
ON DELETE CASCADE ON UPDATE
```

Suppose that each of the following update operations is applied independently to the database state given in CompanySchema.pdf. For each operation, write the name of all constraints if any. Also, write the reason for the violation. Write the new state of each table after each operation if the operation is not restricted. Show only the updated tables.

- Insert <'ProductA',4,'Bellaire',2> into PROJECT
- Insert <'Production',4,'943775543','1998-10-01'> into DEPARTMENT
- Insert <'677678989',NULL,'40.0'> into WORKS\_ON
- Delete the PROJECT tuple with Pname='ProductX'
- Delete the DEPARTMENT tuple with Dnumber=4
- Modify the Dnumber attribute of the DEPARTMENT tuple with Dnumber=4 to 2.

2. (24 puan) Write the following queries in SQL on the relational database schema for COMPANY database given in CompanySchema.pdf.

1. Find the last name of employees who have salary greater than 30 000.
2. Find the average salary of employees who works on the projects which are located in 'Stafford'.
3. Find the total working hours of each manager.
4. Find the name of the projects on which no employee works.

3. (30 p) Using tables given below, show the results of the following operations given in relational algebra.

**A**

a1	a2	a3	a4
2	3	4	5
1	8	12	3
5	6	7	4

**B**

b1	b2	b3
2	4	3
4	12	11

**C**

c1	c2
4	4
6	4
4	12
4	7
7	6
7	12
6	7
7	4

- a)  $A \times (\sigma_{B.b2 > 10} B)$
- b)  $A \bowtie_{A.a4=B.b3} B$
- c)  $A \bowtie_{(A.a1=B.b1 \text{ AND } A.a3=B.b2)} B$
- d)  $B \cup (\pi_{(a1, a2, a3)} A)$
- e)  $\rho_{(d1, d2, d3)}(c1 \text{ } \mathcal{S}_{\text{COUNT } c2, \text{ SUM } c2} C)$
- f)  $C \div (\rho_{(c2)} (\pi_{(a3)} A))$

**STUDENT**

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

**COURSE**

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

**SECTION**

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

**GRADE\_REPORT**

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

**PREREQUISITE**

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

4. (16 p) Show the result of the following queries as it would apply to the database state given above.

1. **SELECT DISTINCT** Name **FROM** STUDENT **WHERE** Major='CS';
2. **SELECT** GRADE\_REPORT.Student\_number, **SUM**(Credit\_hours), **COUNT**(\*) **FROM** GRADE\_REPORT, SECTION, COURSE **WHERE** GRADE\_REPORT.Section\_identifier=SECTION.Section\_identifier **AND** SECTION.Course\_number=COURSE.Course\_number **GROUP BY** GRADE\_REPORT.Student\_number;
3. **SELECT** Name, Course\_name, C.Course\_number, Credit\_hours, Semester, Year, Grade **FROM** STUDENT ST, COURSE C, SECTION S, GRADE\_REPORT G **WHERE** Class=1 **AND** Major='CS' **AND** ST.Student\_number=G.Student\_number **AND** G.Section\_identifier=S.Section\_identifier **AND** S.Course\_number=C.Course\_number;
4. **SELECT** Name, Major **FROM** STUDENT **WHERE NOT EXISTS** (**SELECT** \* **FROM** GRADE\_REPORT **WHERE** Student\_number= STUDENT.Student\_number **AND** Grade<>'B');