SOLUTION

Question 1

A.

Lock and Unlock instructions:

T_{31}:	T_{32}:
lock-S(A)	**lock-S**(B)
read(A)	**read**(B)
lock-X(B)	**lock-X**(A)
read(B)	**read**(A)
if $A = 0$	**if** $B = 0$
then $B := B + 1$	**then** $A := A + 1$
write(B)	**write**(A)
unlock(A)	**unlock**(B)
unlock(B)	**unlock**(A)

Execution of these transactions can result in deadlock. For example, consider the following partial schedule:

T_{31}	T_{32}
lock-S(A)	
	lock-S(B)
	read(B)
read(A)	
lock-X(B)	
	lock-X(A)

The transactions are now deadlocked

B. It is relatively simple to implement, imposes low rollback overhead because of cascadeless schedules, and usually allows an acceptable level of concurrency.

Question 2

Solution:

A. Transactions T3 and T1 get rolled back.

B. As transaction T2 reads a value from T1, it gets rolled back as well.

C. All the write operations by transactions T1, T2, and T3 will be undone.

D. All the write operations by T2 are guaranteed to have taken place on disk.

E. None. All of the write operations could have already taken place on the disk.

Questions 3:

A.

Table – 2: Schedule S2

Time	T11	T12	T13	System Log	Database

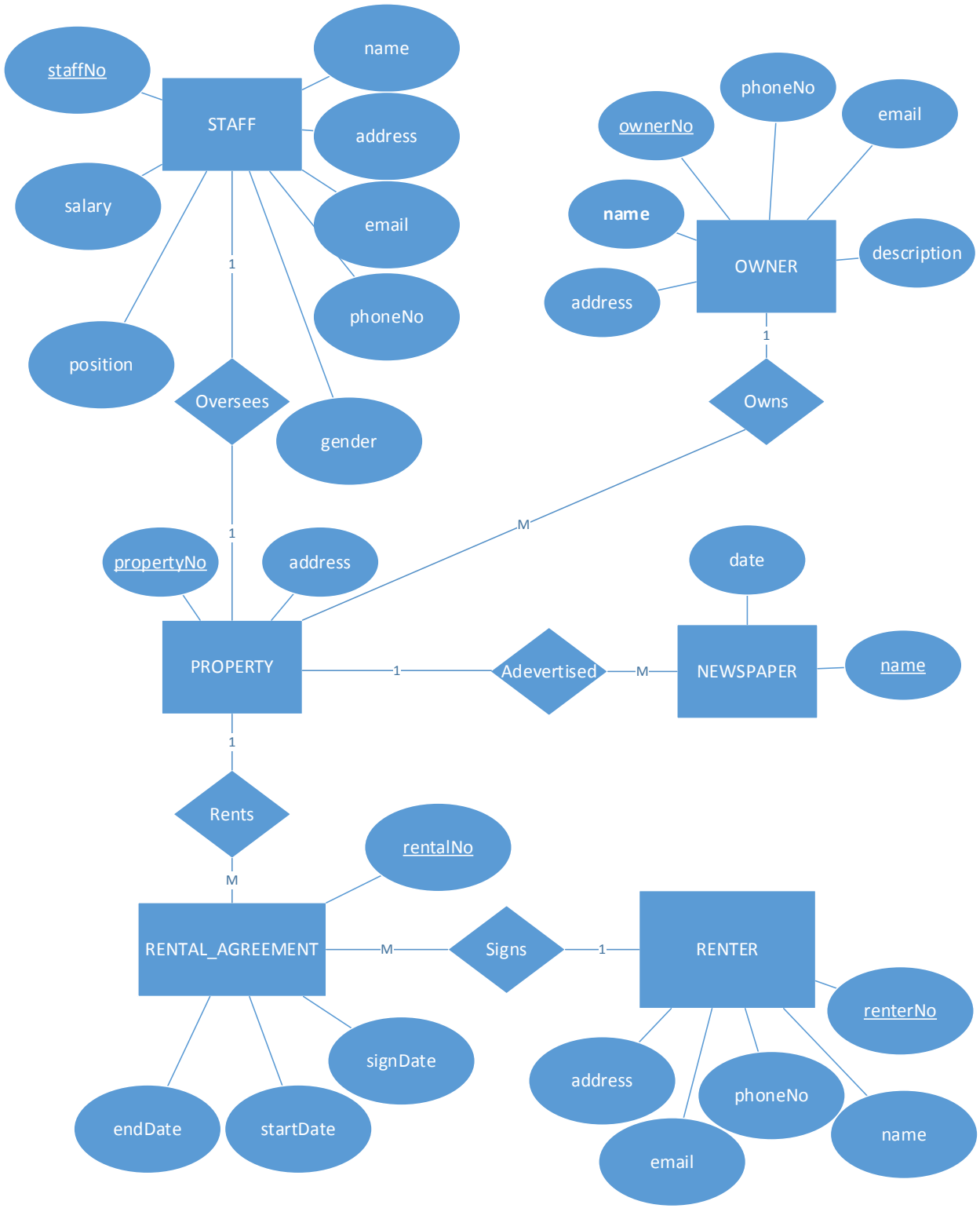
t_1		begin_transaction			X=100, Y=50, Z=25, Sum = 0
t_2	begin_transaction	Sum = 0			X=100, Y=50, Z=25, Sum = 0
t_3	read(X)	Read(X)			X=100, Y=50, Z=25, Sum = 0
t_4	X = X – 10	Sum = Sum + X			X=100, Y=50, Z=25, Sum = 0
t_5	Write (X)	Read(Y)		**X_{11} = 90**	X=100, Y=50, Z=25, Sum = 0
t_6	Read(Z)	Sum = Sum + Y			X=100, Y=50, Z=25, Sum = 0
t_7	Z = Z+10				X=100, Y=50, Z=25, Sum = 0
t_8	Write(Z)		Begin_transaction	**Z_{11} = 35**	X=100, Y=50, Z=25, Sum = 0
t_9	Commit	Read(Z)	Read(X)		X = **90**, Y=50, Z=**35**
t_{10}		Sum = Sum + Z	X = X+10		X = 90, Y=50, Z=35
t_{11}			Write (X)	**X_{13} = 110**	X = 90, Y=50, Z=35
t_{12}		Read(X)			X = 90, Y=50, Z=35
t_{13}		X = X – 10			X = 90, Y=50, Z=35
t_{14}		Write(X)	Rollback	**X_{12} = 80**	X = 90, Y=50, Z=35
t_{15}		Commit			X = **8**0, Y=50, Z=35

B)

***Phantom Read**, T12 get two different values of X;*

***Inconsistent summary**: Sum is summarized wrongly*

Question 4:


staffNo
name
STAFF
address
salary
email
phoneNo
position
Oversees
gender
1
1
ownerNo
phoneNo
email
name
OWNER
description
address
1
Owns
M
propertyNo
address
PROPERTY
1
Adevertised
M
date
NEWSPAPER
name
1
Rents
M
rentalNo
RENTAL_AGREEMENT
M
Signs
1
RENTER
renterNo
signDate
endDate
startDate
address
phoneNo
email
name

Question 5:

Solution:

1. **Find the agency names for agencies that located in the same city as passenger with passenger id 123.**

   Π aname (agency ⋈ acity = pcity (σ pid = 123 (passenger)))

2. **Get the details of flights that are scheduled on both dates 01/12/2020 and 02/12/2020 at 16:00 hours.**

   (σ fdate = 01/12/2020 ^ time = 16:00 (flight)) ∩ (σ fdate = 02/12/2020 ^ time = 16:00 (flight))

3. **Find the agency names for agencies who do not have any bookings for passenger with id 123.**

   Π aname (agency ⋈ (Π aid (agency) – Π aid (σ pid = 123 (booking)))

4. **Find the details of all male passengers who are associated with Jet agency.**

   Π passengers.pid, pname, pcity (σ pgender = "Male" ^ aname = 'Jet' (passengers ⋈ booking ⋈ agency)

5. **Find all the names of the passenger who are using the same flight and agency as passenger id 123.**

   Π passengers.pname, fid, aid (passengers ⋈ booking ⋈ agency⋈ flight) Π fid, aid (σ pid = 123 (passengers ⋈ booking ⋈ agency⋈ flight))

Question 6:

1st Normal Form

StudentNo	StudentName	Major	CourseNo	InstructorNo	InstructorName	CourseName	Grade	InstructorLocation
1	Ali	CS	1001	1	Anila,	Programming Fundamentals,	A	Karachi, Fast,
2	Abdullah	CS	1002	2	Huzaifa,	Calculus,	A+	Karachi, Fast,
3	Babar	SE	1003	3	Yousuf	Basic Electrical Engineering	A	Karachi, Fast,
1	Ali	CS	1002	2	Huzaifa	Calculus	A+	Karachi, Fast
1	Ali	CS	1003	3	Yousuf,	Basic Electrical Engineering	B	Karachi, Fast
2	Abdullah	CS	1004	4	Azam,	English	A+	Karachi, Fast

2	Abdullah	CS	1003	3	Yousuf	Basic Electrical Engineering	A+	Karachi, Fast
3	Babar	SE	1005	5	Khalid	Applied Physics	B+	Karachi, Fast
3	Babar	SE	1006	1	Anila	Introduction to IT	A+	Karachi, Fast

FDS

Primary Key:

Student_Grade_Report (<u>StudentNo,</u> StudentName, Major, <u>CourseNo,</u> CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

StudentNo → StudentName, Major (PD)

StudentNo, CourseNo→ Grade (PD)

CourseNo→CourseName, Coursename, InstructorNo, InstructorName, InstructorLocation (PD)

InstructorNo → InstructorName, InstructorLocation (TD)

2NF

Student (StudentNo, StudentName, Major)

CourseGrade (StudentNo, CourseNo, Grade)

CourseInstructor (CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation)

3NF

Student (StudentNo, StudentName, Major)

CourseGrade (StudentNo, CourseNo, Grade)

Course (CourseNo, CourseName, InstructorNo)

Instructor (InstructorNo, InstructorName, InstructorLocation)

Question 7:

a)

Filesystems might have different copies of same datafiles Which leads to:

a. effort duplication,
b. more storage wastage
c. and same format should be used for entering same data in different files

databases minimize the redundancy by normalization and hence the above stated problems won't occur.

b)

i. Stored procedures can be invoked explicitly by the user. It's like a program that can take some input as a parameter then can do some processing and can return values.
ii. trigger is a stored procedure that runs automatically when various events happen

c)

Logical data independence

a. is the capacity to *change the conceptual schema without having to change external schemas.*
b. E.g., Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

Physical data independence

c. is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well.
d. for example, by creating additional access structures—to improve the performance of retrieval or update. changing access paths, compressing and saving data records.

logical independence is difficult to achieve