Quiz 4, 5, 6, 7, 8 Fall 18

Test Directions: On your honor, you agree to neither give assistance nor receive assistance throughout the exam. Any suspicion of cheating will be dealt with swiftly and harshly. Answer each question to the best of your ability. For all coding questions, partial credit is available. Therefore it is in your best interest to attempt to answer every question.
There are no intentional syntax mistakes in the code.
Name: _____AML140830_____________

Note: We will be taking random attendance 5 times for the rest of the semester. Failing to be present in any of those days will deduct 100 points as if you are not in class to take the quiz.

1. (14.31) Consider the following relation for published books:
   BOOK (Book_title, Authorname, Book_type, Listprice, Author_affil, Publisher)
   Author_affil referes to the affiliation of the author. Suppose the following dependencies exist:
   Book_title -> Publisher, Book_type
   Book_type -> Listprice
   Author_name -> Author-affil

   (a) What normal form is the relation in? Explain your answer.
   (b) Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

   (a)The key for this relation is Book_title,Authorname. This relation is in 1NF and not in 2NF as no attributes are FFD on the key. It is also not in 3NF.

   (b) 2NF decomposition:

   Book0(Book_title, Authorname)

   Book1(Book_title, Publisher, Book_type, Listprice)

   Book2(Authorname, Author_affil)

   This decomposition eliminates the partial dependencies.

   3NF decomposition:

   Book0(Book_title, Authorname)

   Book1-1(Book_title, Publisher, Book_type)

   Book1-2(Book_type, Listprice)

   Book2(Authorname, Author_affil)
   This decomposition eliminates the transitive dependency of Listprice

2. (14.33) Consider the following relation: R (Doctor#, Patient#, Date, Diagnosis, Treat_code, Charge)

   In this relation, a tuple describes a visit of a patient to a doctor along with a treatment code and daily charge. Assume that diagnosis is determined (uniquely) for each patient by a doctor. Assume that each treatment code has a fixed charge (regardless of patient). Is this relation in 2NF? Justify your answer and decompose if necessary. Then argue whether further normalization to 3NF is necessary, and if so, perform it.

   From the question's text, we can infer the following functional dependencies:

   {Doctor#, Patient#, Date}®{Diagnosis, Treat_code, Charge}
   {Treat_code}®{Charge}

   Because there are no partial dependencies, the given relation is in 2NF already. This however is not 3NF because the Charge is a nonkey attribute that is determined by another nonkey attribute, Treat_code. We must decompose further:

   R (Doctor#, Patient#, Date, Diagnosis, Treat_code)

   R1 (Treat_code, Charge)

We could further infer that the treatment for a given diagnosis is functionally dependent, but we should be sure to allow the doctor to have some flexibility when prescribing cures.

3. (Section 20.3) List and briefly explain the ACID properties provided by a Database Management System.

   Atomicity.
   guarantees that each transaction is treated as a single "unit"

   Consistency.
   ensures that a transaction can only bring the database from one valid state to another

   Isolation.
   ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially

   Durability.
   guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure

4. (Chapter 8) Describe briefly what is a query tree and what are its components?

   A tree data structure representing a relational algebra expression. The tables of the query are represented as leaf nodes. The relational algebra operations are represented as the internal nodes. The root represents the query as a whole.

5. (Chapter 20) Briefly describe the basic Two-Phase Locking (2PL) protocol.

1. **Expanding phase (aka Growing phase): gathering of locks, this section only allows getting more locks**
2. **Shrinking phase (aka Contracting phase): releasing of locks, this section only allows letting go of locks**

6. (16.34) Consider a disk with the following characteristics (these are not parameters of any particular disk unit): block size B=512 bytes, interblock gap size G=128 bytes, number of blocks per track=20, number of tracks per surface=400. A disk pack consists of 15 double-sided disks.

   a. What is the total capacity of a track and what is its useful capacity (excluding interblock gaps)?
      Total Capacity =(512+128)*20 =12800 = 12.8 Kbytes
      Useful capacity of a track=512*20 = 10240 = 10.24 Kbytes
   b. How many cylinders are there?
      Number of cylinders = number of tracks = 400
   c. What is the total capacity and the useful capacity of a cylinder?
      Total cylinder capacity =: (512+128)*20*15*2=384000 = 384 KB
      Useful cylinder capacity =: 512*20*15*2=307200 = 307.2 KB
   d. What is the total capacity and the useful capacity of a disk pack?
      Total capacity of a disk pack =: (512+128)*20*400*15*2=153600000 =>153.6MB Useful capacity of a disk pack =: 512*20*400*15*2=122880000 =>122.88MB
   e. Suppose the disk drive rotates the disk pack at a speed of 2400 rpm (revolutions per minute); what is the transfer rate in bytes/msec and the block transfer time btt in msec? What is the average rotational delay rd in msec? What is the bulk transfer rate (see Appendix B)?
      Transfer rate tr= (12800 bytes)/(60*2400rpm)*1000msec/sec = 88.9bytes/msec block transfer time btt=B*tr=(512)/88.9= 5.76 msec average rotational delay rd ==1/2*(1/2400)*60*1000= 12.5 msec
   f. Suppose the average seek time is 30 msec. How much time does it take (on the average) in msec to locate and transfer a single block given its block address?
      average time to locate and transfer a block = btt+rd+st=5.76+12.5+30=48.26msec
   g. Calculate the average time it would take to transfer 20 random blocks and compare it with the time it would take to transfer 20 consecutive blocks using double buffering to save seek time and rotational delay.

time to transfer 20 random blocks = 20 * (s + rd + btt) = 20 * 43.5 = 870 msec time to transfer 20 consecutive blocks using double buffering = s + rd + 20*btt = 30 + 12.5 + (20*1) = 62.5 msec

7. (17.18) Consider a disk with block size B=512 bytes. A block pointer is P=6 bytes long, and a record pointer is P R =7 bytes long. A file has r=30,000 EMPLOYEE records of fixed-length. Each record has the following fields: NAME (30 bytes), SSN (9 bytes), DEPARTMENTCODE (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX (1 byte), JOBCODE (4 bytes), SALARY (4 bytes, real number). An additional byte is used as a deletion marker.
    a. Calculate the record size R in bytes.
       Record length R = (30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4) + 1 = 115 bytes
    b. Calculate the blocking factor bfr and the number of file blocks b assuming an unspanned organization.
       Blocking factor bfr = floor (B/R) = floor(512/115) = 4 records per block Number of blocks needed for file = ceiling(r/bfr) = ceiling(30000/4) = 7500
    c. Suppose the file is ordered by the key field SSN and we want to construct a primary index on SSN. Calculate
        i. (i) the index blocking factor bfr i (which is also the index fan-out fo);
           Index record size R i = (V SSN + P) = (9 + 6) = 15 bytes Index blocking factor bfr i = fo = floor(B/R i) = floor(512/15) = 34
        ii. (ii) the number of first-level index entries and the number of first-level index blocks;
           Number of first-level index entries r 1 = number of file blocks b = 7500 entries Number of first-level index blocks b 1 = ceiling(r 1 / bfr i) = ceiling(7500/34) = 221 blocks
        iii. (iii) the number of levels needed if we make it into a multi-level index;
           Number of second-level index entries r 2 = number of first-level blocks b 1 = 221 entries Number of second-level index blocks b 2 = ceiling(r 2 /bfr i) = ceiling(221/34) = 7 blocks Number of third-level index entries r 3 = number of second-level index blocks b 2 = 7 entries Number of third-level index blocks b 3 = ceiling(r 3 /bfr i) = ceiling(7/34) = 1 Since the third level has only one block, it is the top index level. Hence, the index has x = 3 levels
        iv. (iv) the total number of blocks required by the multi-level index;
           Total number of blocks for the index b i = b 1 + b 2 + b 3 = 221 + 7 + 1 = 229 blocks
        v. (v) the number of block accesses needed to search for and retrieve a record from the file--given its SSN value--using the primary index;
           Number of block accesses to search for a record = x + 1 = 3 + 1 = 4

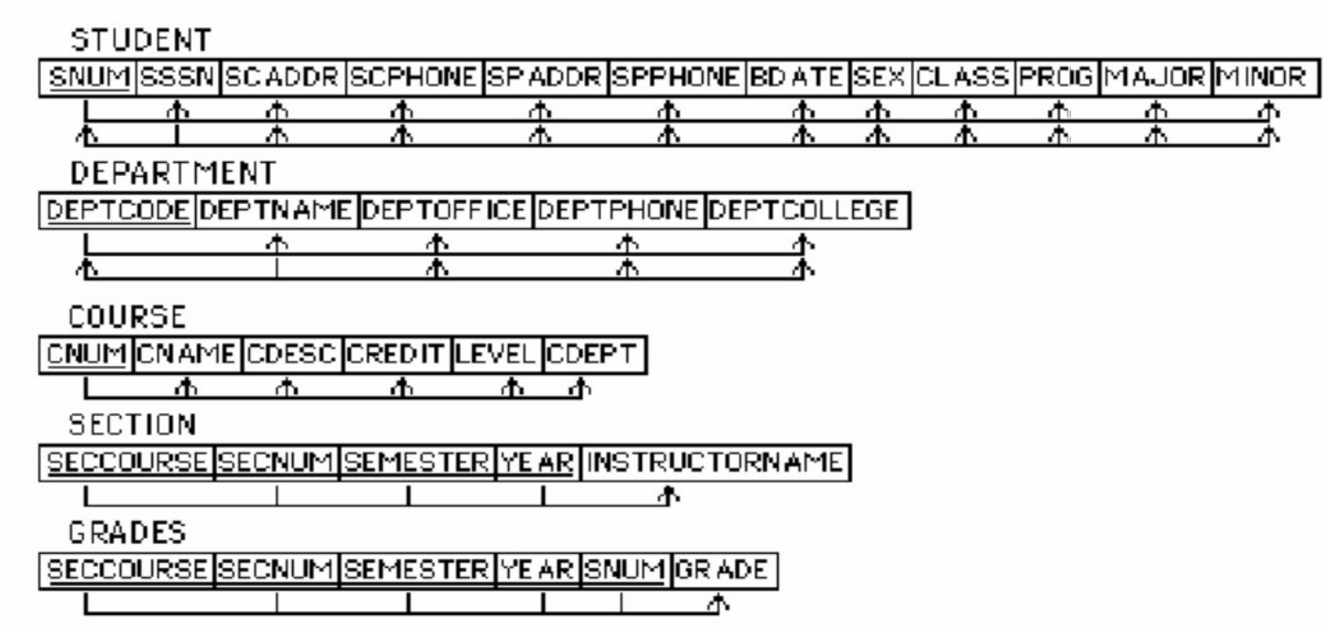
8. (14.19) Suppose we have the following requirements for a university database that is used to keep track of students' transcripts:
    a. The university keeps track of each student's name (SNAME), student number (SNUM), social security number (SSSN), current address (SCADDR) and phone (SCPHONE), permanent address (SPADDR) and phone (SPPHONE), birthdate (BDATE), sex (SEX), class (CLASS) (freshman, sophomore, ..., graduate), major department (MAJORDEPTCODE), minor department (MINORDEPTCODE) (if any), and degree program (PROG) (B.A., B.S., ..., Ph.D.). Both ssn and student number have unique values for each student.
    b. Each department is described by a name (DEPTNAME), department code (DEPTCODE), office number (DEPTOFFICE), office phone (DEPTPHONE), and college (DEPTCOLLEGE). Both name and code have unique values for each department.
    c. Each course has a course name (CNAME), description (CDESC), code number (CNUM), number of semester hours (CREDIT), level (LEVEL), and offering department (CDEPT). The value of code number is unique for each course.
    d. Each section has an instructor (INSTUCTORNAME), semester (SEMESTER), year (YEAR), course (SECCOURSE), and section number (SECNUM). Section numbers distinguish different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ...; up to the number of sections taught during each semester.
    e. A grade record refers to a student (Ssn), refers to a particular section, and grade (GRADE).

    Design a relational database schema for this database application. First show all the functional dependencies that should hold among the attributes. Then, design relation schemas for the database that are each in 3NF or BCNF. Specify the key attributes of each relation. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.

From the above description, we can presume that the following functional dependencies hold on the attributes:
FD1: {SSSN} -> {SNAME, SNUM, SCADDR, SCPHONE, SPADDR, SPPHONE, BDATE, SEX, CLASS, MAJOR, MINOR, PROG}
FD2: {SNUM} -> {SNAME, SSSN, SCADDR, SCPHONE, SPADDR, SPPHONE, BDATE, SEX, CLASS, MAJOR, MINOR, PROG}
FD3: {DEPTNAME} -> {DEPTCODE, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE}
FD4: {DEPTCODE} -> {DEPTNAME, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE}
FD5: {CNUM} -> {CNAME, CDESC, CREDIT, LEVEL, CDEPT}
FD6: {SECCOURSE, SEMESTER, YEAR, SECNUM} -> {INSTRUCTORNAME}
FD7: {SECCOURSE, SEMESTER, YEAR, SECNUM, SSSN} -> {GRADE}
These are the basic FDs that we can define from the given requirements; using inference rules IR1 to IR3, we can deduce many others. FD1 and FD2 refer to student attributes; we can define a relation STUDENT and choose either SSSN or SNUM as its primary key.

Similarly, FD3 and FD4 refer to department attributes, with either DEPTNAME or DEPTCODE as primary key. FD5 defines COURSE attributes, and FD6 SECTION attributes.

Finally, FD7 defines GRADES attributes. We can create one relation for each of STUDENT, DEPARTMENT, COURSE, SECTION, and GRADES as shown below, where the primary keys are underlined. The COURSE, SECTION, and GRADES relations are in 3NF and BCNF if no other dependencies exist. The STUDENT and DEPARTMENT relations are in 3NF and BCNF according to the general definition given in Sections 14.4 and 14.5, but not according to the definitions of Section 14.3 since both relations have secondary keys.


STUDENT
SNUM SSSN SCADDR SCPHONE SPADDR SPPHONE BDATE SEX CLASS PROG MAJOR MINOR
DEPARTMENT
DEPTCODE DEPTNAME DEPTOFFICE DEPTPHONE DEPTCOLLEGE
COURSE
CNUM CNAME CDESC CREDIT LEVEL CDEPT
SECTION
SECCOURSE SECNUM SEMESTER YEAR INSTRUCTORNAME
GRADES
SECCOURSE SECNUM SEMESTER YEAR SNUM GRADE

The foreign keys will be as follows:
STUDENT.MAJOR -> DEPARTMENT.DEPTCODE
STUDENT.MINOR -> DEPARTMENT.DEPTCODE
COURSE.CDEPT -> DEPARTMENT.DEPTCODE
SECTION.SECCOURSE -> COURSE.CNUM
GRADES.(SECCOURSE, SEMESTER, YEAR, SECNUM) ->
SECTION.(SECCOURSE, SEMESTER, YEAR, SECNUM)
GRADES.SNUM -> STUDENT.SNUM

Note: We arbitrarily chose SNUM over SSSN for primary key of STUDENT, and DEPTCODE over DEPTNAME for primary key of DEPARTMENT.

9. (20.14) Change transaction T 2 in Figure 21.2b to read:

```
read_item(X);
X:= X+M;
if X > 90 then exit
else write_item(X);
```

Discuss the final result of the different schedules in Figure 21.3 (a) and (b), where M = 2 and N = 2, with respect to the following questions. Does adding the above condition change the final outcome? Does the outcome obey the implied consistency rule (that the capacity of X is 90)?

The above condition does not change the final outcome unless the initial value of X > 88.The outcome, however, does obey the implied consistency rule that X < 90, since the value of X is not updated if it becomes greater than 90.

10. (22.21) Suppose that the system crashes before the [read_item,T3,A] entry is written to the log in Figure 23.1 (b); will that make any difference in the recovery process?

There will be no difference in the recovery process, because read_item operations are needed only for determining if cascading rollback of additional transactions is necessary.

11. (22.24) Suppose that we use the deferred update protocol for the example in Figure 23.6. Show how the log would be different in the case of deferred update by removing the unnecessary log entries; then describe the recovery process, using your modified log. Assume that only redo operations are applied, and specify which operations in the log are redone and which are ignored.

In the case of deferred update, the write operations of uncommitted transactions are not recorded in the database until the transactions commit. Hence, the write operations of T2 and T3 would not have been applied to the database and so T4 would have read the previous (committed) values of items A and B, thus leading to a recoverable schedule.

By using the procedure RDU_M (deferred update with concurrent execution in a multiuser environment), the following result is obtained:

The list of committed transactions T since the last checkpoint contains only transaction T4. The list of active transactions T' contains transactions T2 and T3.Only the WRITE operations of the committed transactions are to be redone. Hence, REDO is applied to:

[write_item,T4,B,15]

[write_item,T4,A,20]

The transactions that are active and did not commit i.e., transactions T2 and T3 are canceled and must be resubmitted. Their operations do not have to be undone since they were never applied to the database.

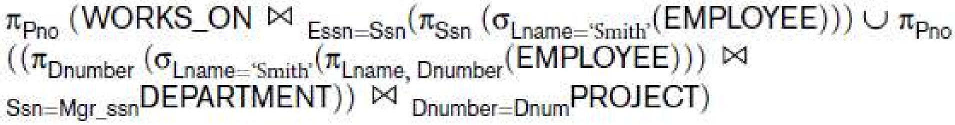
12. Describe the meaning of a Table Scan
 iterate over all table rows

13. Why are the table scans considered to be a problem
 They increase I/O time during the iteration over the rows of

14. How are the table scans avoided?
 Index scans can be used instead of table scans. Because index access significantly reduces the number of I/O read operations, it often outperforms a table scan.

15. Describe the relationship
    a. Schema and Relation
    b. Relation and Attribute
    c. Relation and Tuples

16. Translate the following set of relational operations into a single SQL query:



$$\pi_{Pno}(\text{WORKS\_ON} \bowtie_{Essn=Ssn}(\pi_{Ssn}(\sigma_{Lname=\text{‘Smith’}}(\text{EMPLOYEE}))) \cup \pi_{Pno}((\pi_{Dnumber}(\sigma_{Lname=\text{‘Smith’}}(\pi_{Lname,\ Dnumber}(\text{EMPLOYEE}))) \bowtie_{Ssn=Mgr\_ssn}\text{DEPARTMENT})) \bowtie_{Dnumber=Dnum}\text{PROJECT})$$

17. Which of following languages is used to define the application’s queries?
    a. DDL
    b. DQL
    c. DML
    d. Metadata

B

18. What does DDL define?
    Data Definition Language (DDL)
    DDL is any set of commands that modify the schema and not the actual data
19. What mechanism is used to authorize user?
    Password
20. How do we eliminate duplicates from a select's result set?
    SELECT DISTINCT

21. Which of the following describe the act of making a transaction's changes to the database state permanent?
    a. The SAVE operation
    b. The ROLLBACK operation
    c. The COMPLETE operation
    d. The COMMIT operation

    D

22. What best describes the referential integrity?
    a. Primary to Foreign Key
    b. Primary to Candidate Key
    c. Foreign to Primary Key

    C

23. Which of the clauses is only useful when used in SELECT statements containing aggregation function? e.g. min(), max(), sum(), etc.?
    a. Group By
    b. Having
    c. Order By
    d. In

    A

24. Which of the following is not true of views?
    a. Serves to decouple clients from the internal schema
    b. Can be used to cache the results from complex aggregated queries
    c. Implements CRUD operations across joined tables
    d. Can be used to provide clients access to otherwise protected tables

    C

25. Describe a record blocking factor.
    blocking factor: The number of records in a block. Note: The blocking factor is calculated by dividing the block length by the length of each record contained in the block.

26. What are the two strategies employed by the query processor when performing searches on two attributes from a single table.
    a. Consider the case where one attribute is indexed and the other is not
    b. Consider the case when both attributes are separately indexed

    Use the example of searching for a student with last-name and phone number

27. Name and describe the four levels of Transaction Isolation provided by SQL clients

    a. **Read Uncommitted –** Read Uncommitted is the lowest isolation level. In this level, one transaction may read not yet committed changes made by other transaction, thereby allowing dirty reads. In this level, transactions are not isolated from each other.
    b. **Read Committed –** This isolation level guarantees that any data read is committed at the moment it is read. Thus it does not allows dirty read. The transaction hold a read or write lock on the current row, and thus prevent other rows from reading, updating or deleting it.
    c. **Repeatable Read –** This is the most restrictive isolation level. The transaction holds read locks on all rows it references and write locks on all rows it inserts, updates, or deletes. Since other transaction cannot read, update or delete these rows, consequently it avoids non repeatable read.
    d. **Serializable –** This is the Highest isolation level. A *serializable* execution is guaranteed to be serializable. Serializable execution is defined to be an execution of operations in which concurrently executing transactions appears to be serially executing.

28. How does 2 Phase Locking guarantee the schedules when all transactions are following 2 phase locking?
29. What activities identify the layout of the table files and index files on the target file system?
30. What describes the general solution to normalizing relations that are not in second and third normal form?
    By removing any transitive dependencies
31. Is this true that functional dependencies are symmetric?
32. A relation in First Normal Form must be in _______________ form.
    Second Normal
33. Fundamentally what do indexing and other optimization techniques minimize?
    I/O time

34. What is the disadvantage of Unspanned Records?
    They must be in one block only
35. What is one of the characteristics of Unordered Data Files?
    Good insertion efficiency
36. What is the main purpose of using an index?
    For accessing blocks without traversing the entire set.
37. Under what circumstances a secondary index is required?
    For faster lookup
38. Is the following schedule serializable? R1(Y); R1(X); W1(X);R2(X);W1(Y);C1;W2(X);
39. Which of the locks allow shared access to an object?
    a. Write Locks
    b. Read Locks
    c. Read-Write Locks
    d. Binary Locks

    C

40. Create an EER based on the description.
    The problem:
    Buying and selling artwork is a very high business, and anything that is a big business wants its data to be handled neatly and cleanly, because a fake painting sold for millions is an embarrassment. In that respect, the Marcus Art Gallery has hired you to create a database to track their pictures.

    The Description:
    - Every painting has a unique catalog number, a location, and a text description
    - Every artist has a name, a date of birth, and a unique library reference number
    - Each artist can pain many paintings, but no painting is created by multiple artists. Every painting has been painted by an artist, and that relationship has a date painted
    - Three subclasses of artist exist
        - The first is the genuine master painter, which includes information about their country of origin and the century they painted.
        - The second subclass is the copyist someone under license creates reproduction of original painting, which includes information about their current business address, their telephone number, and the name of their agent
        - The third class of artist is the forger who is a known faker of art and the gallery needs to be aware of their activities, which include if the forger is active, in prison, and which country the faker currently resides

    - Two independent individuals are a dealer and a buyer
    - A dealer is a person with a name, a contact number, and a certain level of reputation

    - Their name and contact number make up a unique identifying number
    - A dealer sells the painting to the gallery, and that relationship has a price attribute and a date. Many different dealers sell many different paintings
- A buyer is a person with a name, a contact number, and a shipping address
    - Their name and contact number make up a unique identifying number
    - A buyer buys the painting from the gallery, and that relationship has a price attribute and a date and a commission. Many different buyers buy many different paintings

41. Try to map the relational schema of Figure 6.14 into an ER schema. This is part of a process known as reverse engineering, where a conceptual schema is created for an existing implemented database. State any assumptions you make.

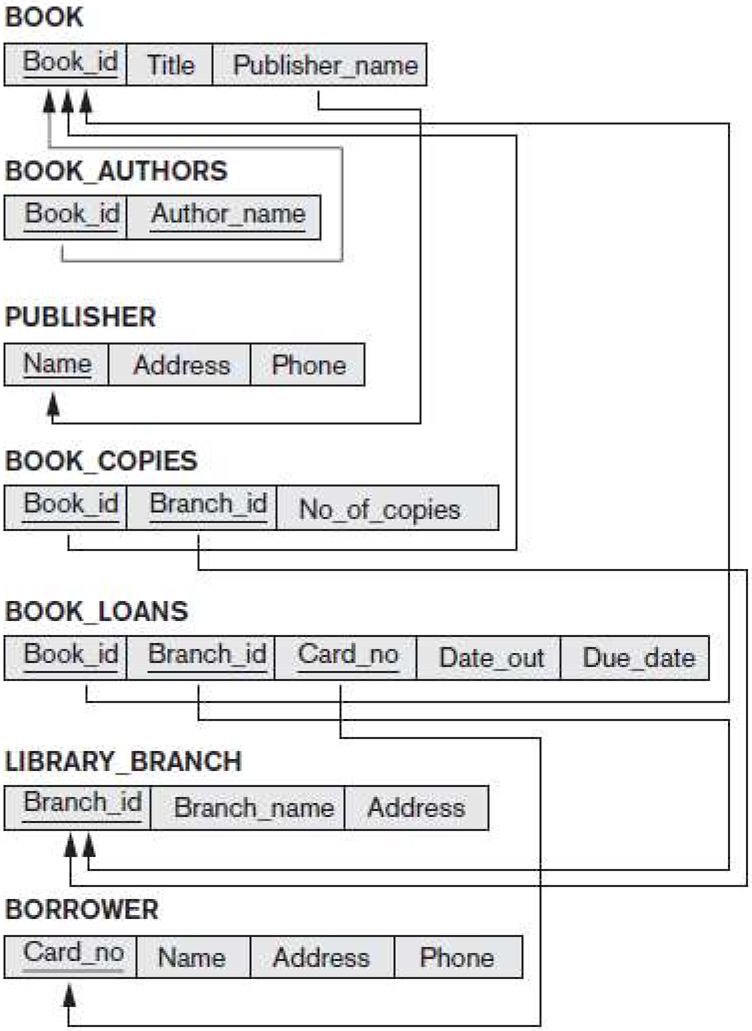
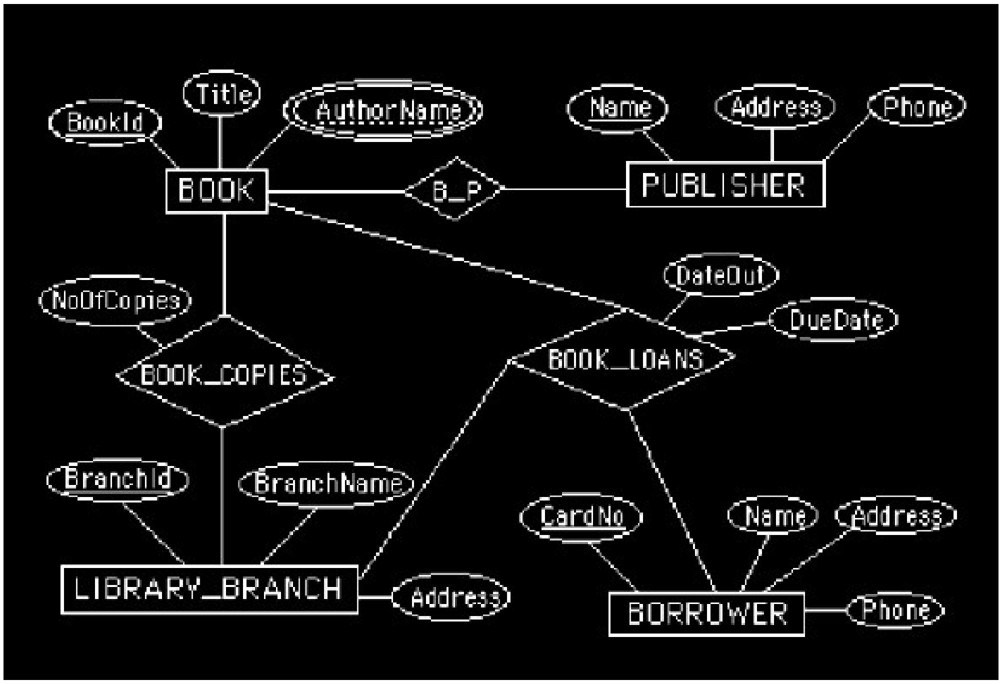
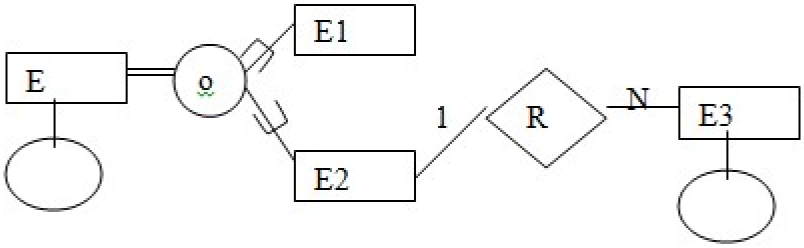


Figure 6.14
A relational database schema for a LIBRARY database.

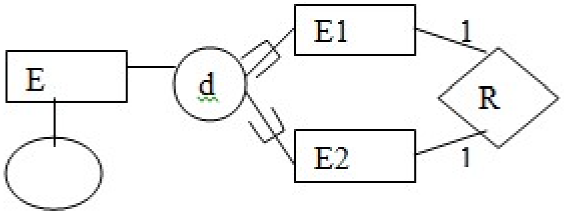


42. Which of the following EER diagram(s) is/are incorrect and why?

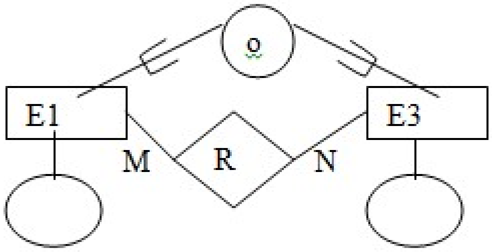
a)



b)



c)



43. Figure 9.8 shows an ER schema for a database that may be used to keep track of transport ships and their locations for maritime authorities. Map this schema into a relational schema, and specify all primary keys and foreign keys.

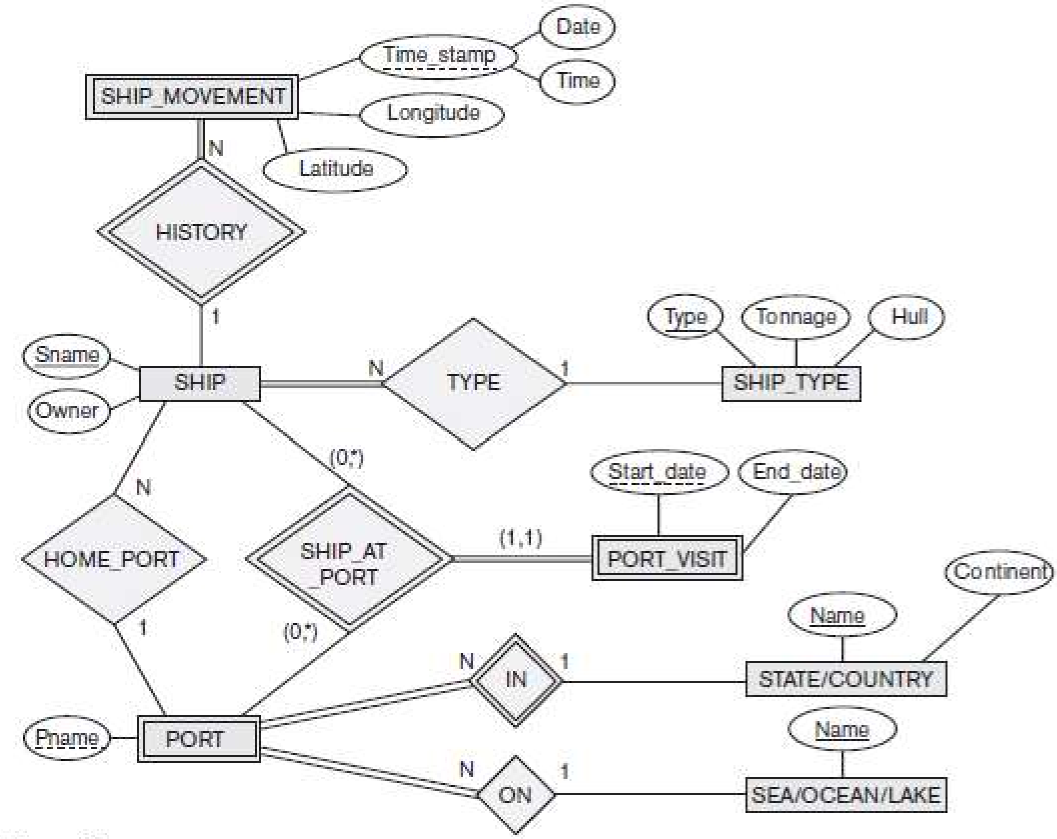


Figure 9.8
An ER schema for a SHIP_TRACKING database.

SHIP

SNAME OWNER TYPE PNAME

SHIP_TYPE TYPE TONNAGE HULL

STATE_COUNTRY NAME CONTINENT

SEAOCEANLAKE

NAME

SHIP_MOVEMENT

SSNAME DATE TIME LONGITUDE LATITUTE

PORT

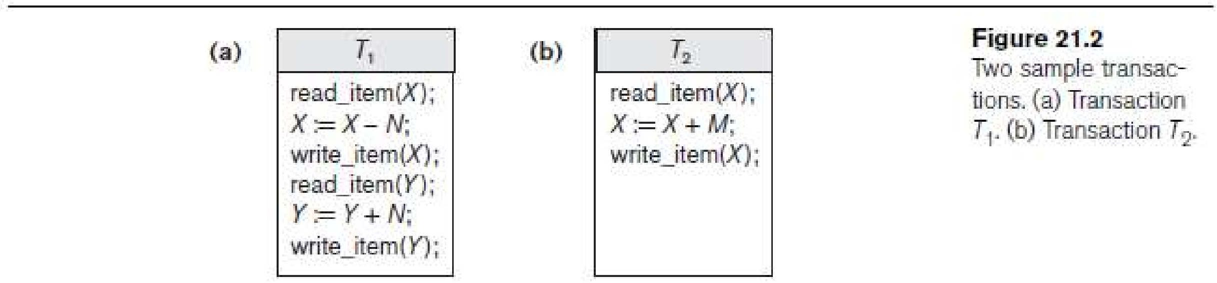
S_C_NAME PNAME S_O_L_NAME

VISIT VSNAME VPNAME STARTDATE ENDDATE

f.k. f.k. f.k. f.k. f.k. f.k. f.k.

21. What act or operation makes transaction's changes to the database state permanent?

9. Fig 21.2



(a) T_1

```
read_item(X);
X := X − N;
write_item(X);
read_item(Y);
Y := Y + N;
write_item(Y);
```

(b) T_2

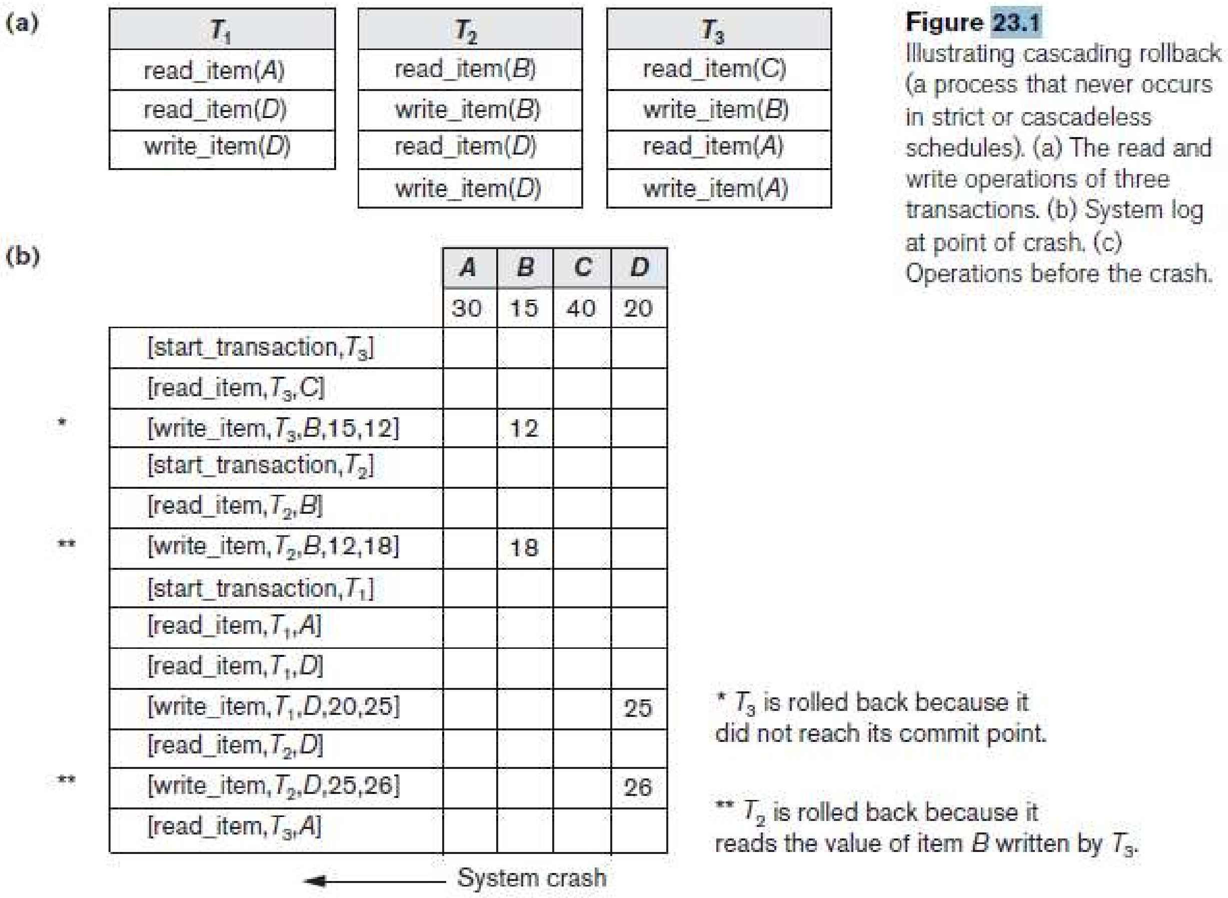
```
read_item(X);
X := X + M;
write_item(X);
```

Figure 21.2
Two sample transactions. (a) Transaction T_1. (b) Transaction T_2.

10. Fig 23.1b

ROLLBACK to T1

Commit



(a)

T_1	T_2	T_3
read_item(A)	read_item(B)	read_item(C)
read_item(D)	write_item(B)	write_item(B)
write_item(D)	read_item(D)	read_item(A)
	write_item(D)	write_item(A)

(b)

		A	B	C	D
		30	15	40	20
	[start_transaction,T_3]				
	[read_item,T_3,C]				
*	[write_item,T_3,B,15,12]		12		
	[start_transaction,T_2]				
	[read_item,T_2,B]				
**	[write_item,T_2,B,12,18]		18		
	[start_transaction,T_1]				
	[read_item,T_1,A]				
	[read_item,T_1,D]				
	[write_item,T_1,D,20,25]				25
	[read_item,T_2,D]				
**	[write_item,T_2,D,25,26]				26
	[read_item,T_3,A]				

* T_3 is rolled back because it did not reach its commit point.

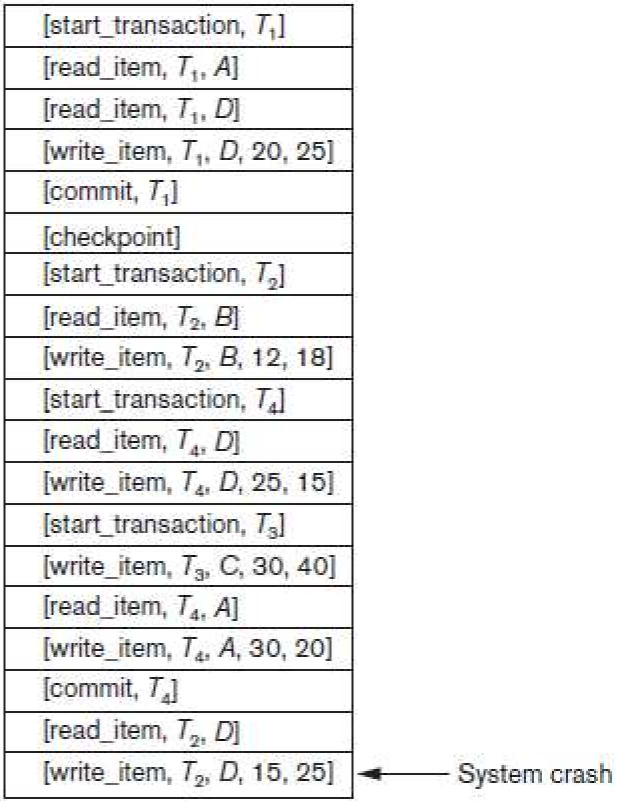
** T_2 is rolled back because it reads the value of item B written by T_3.

Figure 23.1 Illustrating cascading rollback (a process that never occurs in strict or cascadeless schedules). (a) The read and write operations of three transactions. (b) System log at point of crash. (c) Operations before the crash.

11. 23.6

ROLLBACK to T1

Commit



[start_transaction, T_1]	
[read_item, T_1, A]	
[read_item, T_1, D]	
[write_item, T_1, D, 20, 25]	
[commit, T_1]	
[checkpoint]	
[start_transaction, T_2]	
[read_item, T_2, B]	
[write_item, T_2, B, 12, 18]	
[start_transaction, T_4]	
[read_item, T_4, D]	
[write_item, T_4, D, 25, 15]	
[start_transaction, T_3]	
[write_item, T_3, C, 30, 40]	
[read_item, T_4, A]	
[write_item, T_4, A, 30, 20]	
[commit, T_4]	
[read_item, T_2, D]	
[write_item, T_2, D, 15, 25]	← System crash

Figure 23.6
A sample schedule and its corresponding log.

ROLLBACK to T4

Commit