Birla Institute of Technology & Science, Pilani

Work Integrated Learning Programmes Division
M. Tech (Software Engineering) at Wipro Technologies (WASE)
II Semester 2015 - 2016
Comprehensive Examination (Regular)_ANSWER KEY

Course Number	:	SEWP ZC322
Course Title	:	Data Base Management Systems
Type of Exam	:	Open Book
Weightage	:	60 %
Duration	:	3 Hours
Date of Exam	:	31st July, 2016

Session : FN(9 to 12 Noon)

No. of Pages :4
No. of Questions : 6

Note:

1. Please read and follow all the instructions given on the cover page of the answerscript.
2. Start each answer from a fresh page. All parts of a question should be answered consecutively.

Q1. Draw an ER diagram that captures the following information. (10 Mark)

The Prescriptions-R-X chain of pharmacies has offered to give you a free lifetime supply of medicine if you design its database. Given the rising cost of health care, you agree. Here's the information that you gather:

Patients are identified by an SSN, and their names, addresses, and ages must be recorded.

Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.

Each pharmaceutical company is identified by name and has a phone number.

For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.

Each pharmacy has a name, address, and phone number.

Every patient has a primary physician. Every doctor has at least one patient.

Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.

Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.

Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.

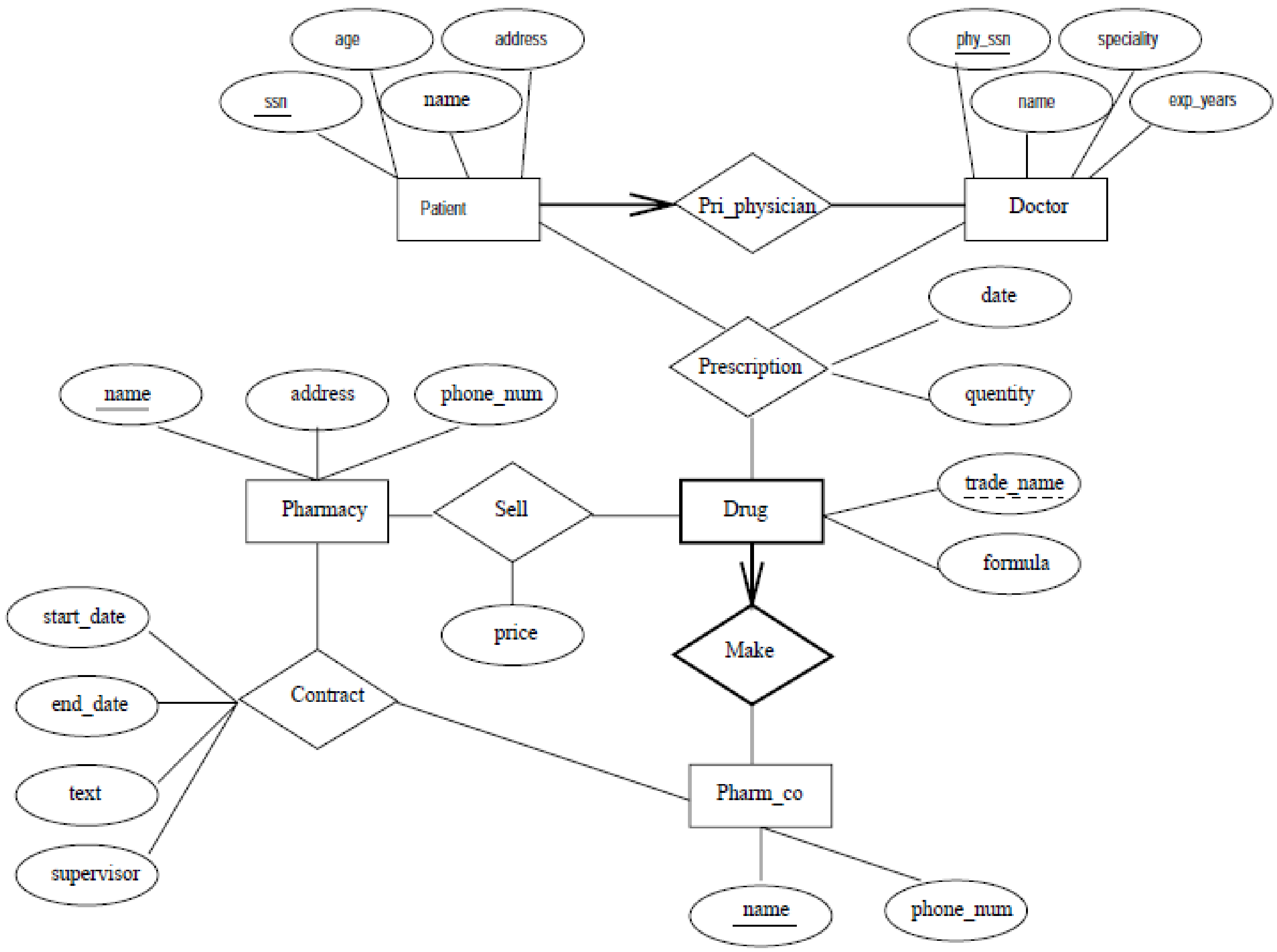
Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.

Ans:- ER Diagram

Evaluation of marks are based upon showing ER Diagram, Cardinality ratio,Keys. No Partial Marks should be offer

Entities correctly identified:	[3 Mark]
Attributes correctly identified:	[2 Mark]
Primary keys correctly identified:	[1 Mark]
Relationships and cardinality correctly identified:	[4 Mark]

ER/ERR Schema (variations are possible to this answer)



Q.2 A Consider the following relational schema and translate the following SQL-query into an expression of the relational algebra. **(2.5 * 2=5 Marks)**

- **Student** (snum, sname, major, level, age)
- **Class** (name, meets at, room, fid)

- **Enrolled** (snum, cname)
- **Faculty** (fid, fname, deptid)

Q2 A)	**Q2A)**
1. SELECT S.sname FROM Student S WHERE S.snum NOT IN (SELECT E.snum FROM Enrolled E)	**2.** SELECT C.name FROM Class C WHERE C.room = ’R128’ OR C.name IN (SELECT E.cname FROM Enrolled E GROUP BY E.cname HAVING COUNT(*) >= 5)

Answer:-

Q.2 Relational Algebra for the given SQL

1)

$$\pi_{\texttt{S.sname}}(\rho_{\texttt{S}}(\texttt{Student})\ \overline{\bowtie}$$
$$\pi_{\texttt{S.snum,S.sname,S.major,S.level,S.age}}\ \sigma_{\texttt{E.snum=S.snum}}\ (\rho_{\texttt{E}}(\texttt{Enrolled}) \times \rho_{\texttt{S}}(\texttt{Student})))$$

2)

$$\pi_{\texttt{C.name}}\ \sigma_{\texttt{C.room='R128'}}\ \rho_{\texttt{C}}(\texttt{Class})$$
$$\cup\ \pi_{\texttt{C.cname}}\ \sigma_{\texttt{COUNT(*)>=5}}\ \gamma_{\texttt{E.cname,COUNT(*),C.name,C.meets\_at,C.room,C.fid}}$$
$$\sigma_{\texttt{E.cname=C.cname}}(\rho_{\texttt{E}}(\texttt{Enrolled}) \times \rho_{\texttt{C}}(\texttt{Class})).$$

Q2B) (5 Marks)
EmpMaster (eid: number, ename: varchar, sal: number, age: number, deptid: number)
Dept(deptid: number, budgetAmt: number, floor: number, mgreid: number)
Salaries range from Rs.10,000 to Rs.100,000, ages vary from 20 to 80, each department has about five employees on average, there are 10 floors, and budgets vary from RS.10,000 to Rs 100,000. You can assume uniform distribution of values.

Query: Print ename, age, and sal for all employees.
Query: Find the deptids of departments that are on the 10th floor and have a budget of less than Rs25,000

Question 1 :- For each of the following queries, which index would you choose to speed up the query? (2.5 Marks)
Question 2:- If your database system does not consider index-only plans (i.e.,data records are always retrieved even if enough information is available in the index entry), how would your answer change? Explain briefly.
(2.5 Marks)

Answer:-
Q2 B Option 1:- (2.5 Marks)
We should create an unclustered hash index on ename, age, sal fields of EmpMaster since then we could do on index only scan. If our system does not include index only plans then we should not create index for this query. Since this query requires us to access all the Emp records, an index won’t help us any, and so should we access the records using a filescan.
Q2B Option 2:- (2.5 Marks)
We should create a clustered dense B+tree index on floor, budget fields of Dept, since the records would be ordered on these fields then. So when executing this query, the first record with floor=10 must be retrieved, and then the other records with floor =10 can be read in order of budget. This plan, which is the best for this query, is not an index-only plan

Q3 A: Modern disk drives store more sectors on the outer tracks than the inner tracks.
Since the rotation speed is constant, the sequential data transfer rate is also higher on the outer tracks. The seek time and rotational delay are unchanged. Given this information, explain good strategies for placing files with the following kinds of access patterns:
a. Frequent, random accesses to a small file (e.g., catalog relations).
b. Sequential scans of a large file (e.g., selection from a relation with no index).
c. Random accesses to a large file via an index (e.g., selection from a relation via the index).
d. Sequential scans of a small file.

(1.5*4=6 Marks)

Ans:
a. Place the file in the middle tracks. Sequential speed is not an issue due to the small size of the file, and the seek time is minimized by placing files in the center. (1.5 Marks)
b. Place the file in the outer tracks. Sequential speed is most important and outer tracks maximize it. (1.5 Marks)
c. Place the file and index on the inner tracks. The DBMS will alternately access pages of the index and of the file, and so the two should reside in close proximity to reduce seek times. By placing the file and the index on the inner tracks we also save valuable space on the faster (outer) tracks for other files that are accessed sequentially. (1.5 Marks)
d. Place small files in the inner half of the disk. A scan of a small file is effectively random I/O because the cost is dominated by the cost of the initial seek to the beginning of the file. (1.5 Marks)

Q3B.
What are checkpoints and why are they important? List the actions taken by the recovery manager during checkpoints? (4 Marks)

Ans. Checkpoint is a type of entry in the log. A checkpoint record is written into the log periodically at that point when the system writes out to the database on disk all DBMS buffers that have been modified. As a consequence of this, all transactions that have their [commit, T] entries in the log before a checkpoint entry do not need to have their WRITE operations *redone* in case of a system crash, since all their updates will be recorded in the database on disk during checkpointing. Actions taken by the recovery manager during Checkpoint The recovery manager of a DBMS must decide at what intervals to take a checkpoint. The interval may be measured in time—say, every *m* minutes—or in the number *t* of committed transactions since the last checkpoint, where the values of *m* or *t* are system parameters. (2 Marks)

Taking a checkpoint consists of the following actions: 1. Suspend execution of transactions temporarily. 2. Force-write all main memory buffers that have been modified to disk..
3. Write a [checkpoint] record to the log, and force-write the log to disk. 4. Resume executing transactions. As a consequence of Step 2, a checkpoint record in the log may also include additional information, such as a list of active transaction ids, and the locations (addresses) of the first and most recent (last) records in the log for each active transaction. This can facilitate undoing transaction operations in the event that a transaction must be rolled back.(2 Marks)

Q4 (10 Marks)

Consider an empty B_+-tree with *n=4*, insert the following record keys in the specified order:
80, 25, 69, 14, 58, 3, 47, 91, 36, 81.

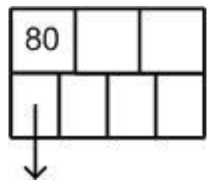
Ans:-

First step is to make the following table: Table 2 Marks, Tree 8 Marks

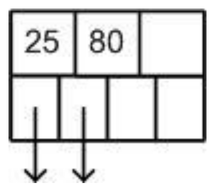
B_+-tree with n=4		

	Root	Interior node	Leaf node
Minimum number of pointers	2	$\lceil (n)/2 \rceil = 2$	$\lfloor (n)/2 \rfloor = 2$
Number of pointers to keep in the first node after splitting	$\lceil (n+1)/2 \rceil = 3$	$\lceil (n+1)/2 \rceil = 3$	$\lceil (n)/2 \rceil = 2$

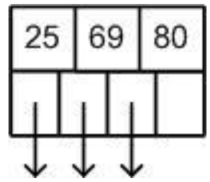
B^+-tree after inserting the record with key 80:



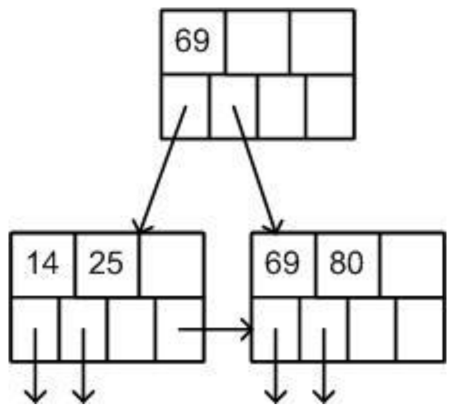
B^+-tree after inserting the record with key 25:



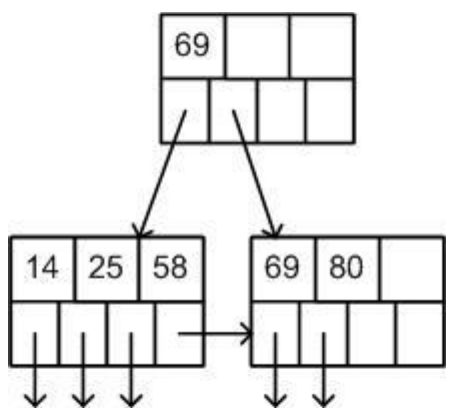
B$^+$-tree after inserting the record with key 69:



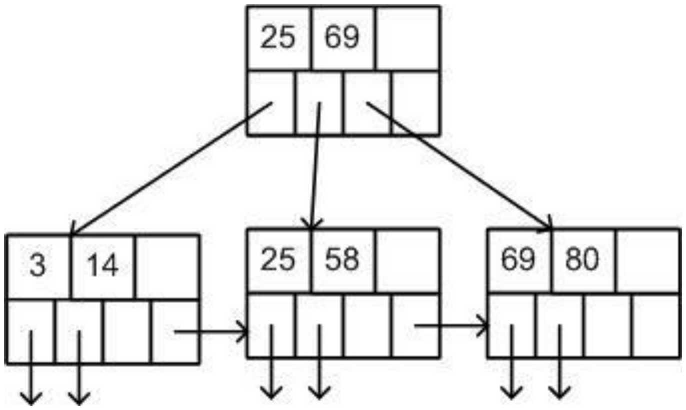
B$^+$-tree after inserting the record with key 14:



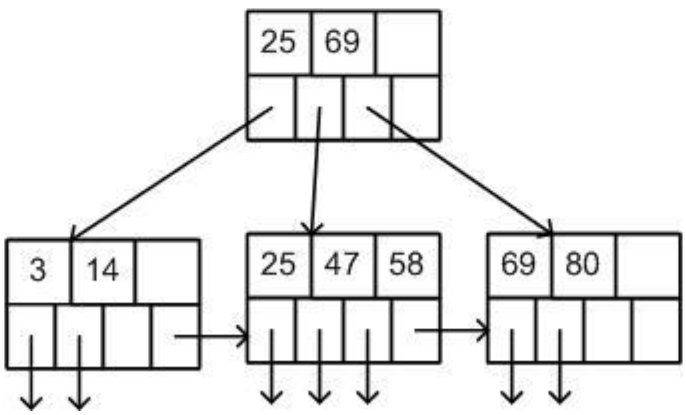
B$^+$-tree after inserting the record with key 58:



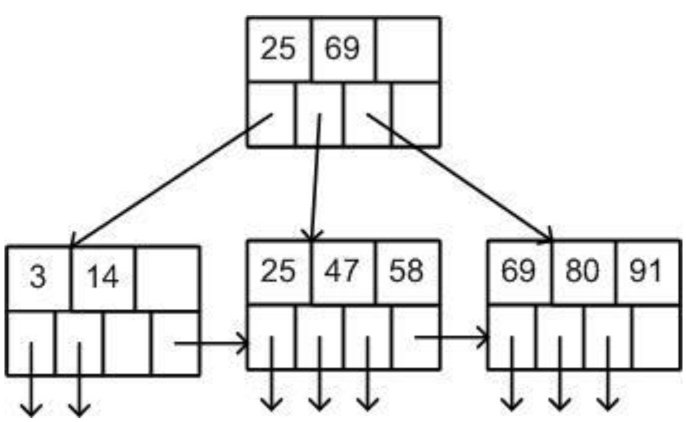
B⁺-tree after inserting the record with key 3:



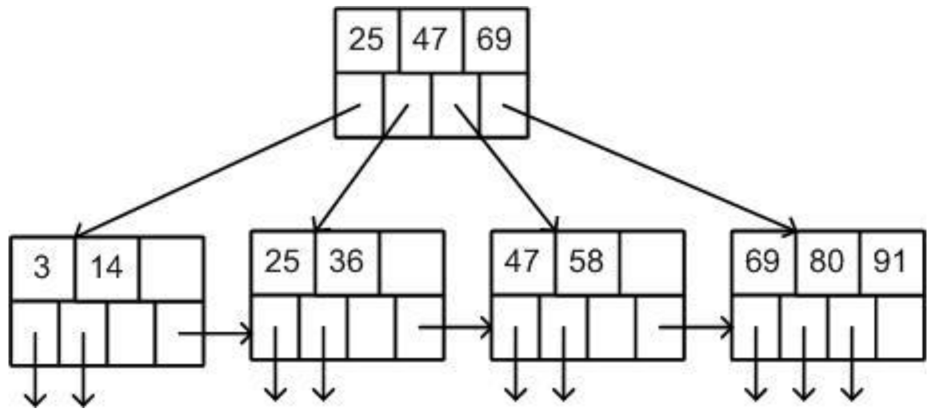
B⁺-tree after inserting the record with key 47:



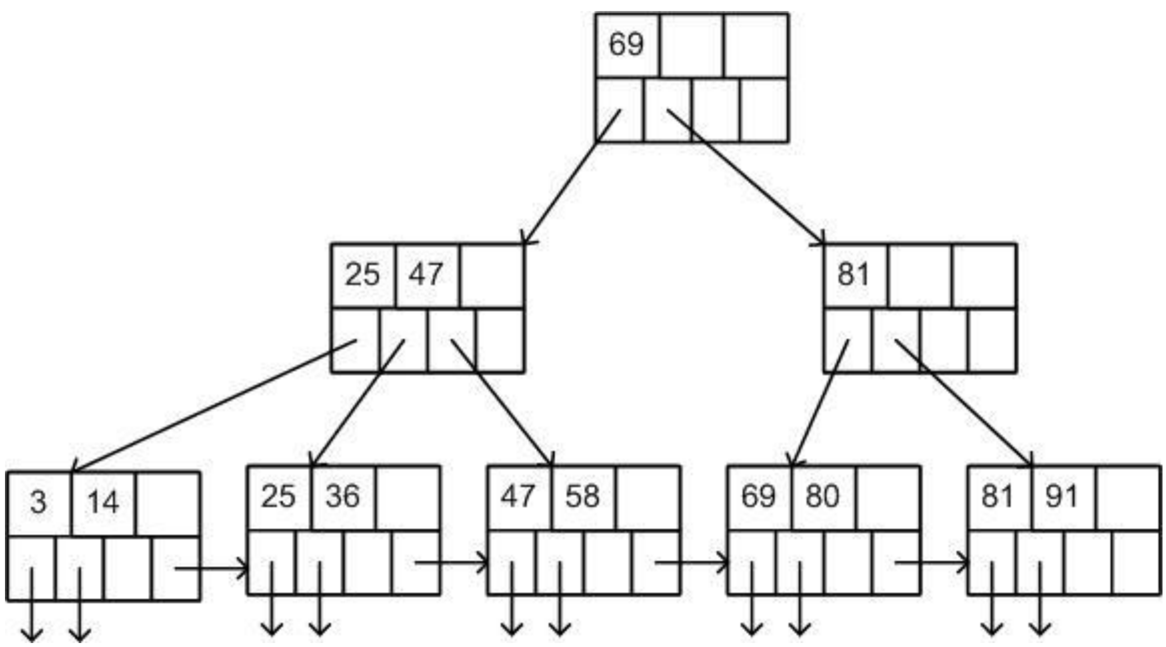
B⁺-tree after inserting the record with key 91:



B$^+$-tree after inserting the record with key 36:



B$^+$-tree after inserting the record with key 81:



Q5A) Consider the following partial Schedule S involving two transactions T1 and T2. Only the read and the write operations have been shown. The read operation on data item P is denoted by read (P) and the write operation on data item P is denoted by write (P). What would the consequence when the transaction T1 fails immediately after time instance 9. **(5 Marks)**

Time	Transaction-id	
	T1	*T2*
1	*read*(*A*)	
2	*write*(*A*)	
3		*read*(*C*)
4		*write*(*C*)
5		*read*(*B*)
6		*write*(*B*)
7		*read*(*A*)
8		*commit*
9	*read*(*B*)	

Q.5B) Consider two transactions: **(5 Marks)**

T1: BEGIN A=A+100, B=B-100 END

T2: BEGIN A=1.06*A, B=1.06*B END

- 1st TXN transfers $100 from B's account to A's
- 2nd TXN credits both accounts with 6% interest.

Assume at first A and B each have $1000. What are the legal outcomes of running T1 and T2???

Answer :-

Q.5A) Schedule S is non-recoverable and cannot ensure transaction atomicity because T2 reads value of 'A' which is written by T1 and T2 is committed before T1. **(5 Marks)**

Q.5B) There is no guarantee that T1 will execute before T2 or vice-versa, if both are submitted together. But, the net effect *must* be equivalent to these two transactions running serially in some order.

Legal outcomes: **A=1166, B=954 or A=1160, B=960** **(5 Marks)**

Q6 :10 Marks (2 for Each answer from set of 6.1 to 6.5)

Suppose you are given a relation R with four attributes ABCD. For each of the following sets of FDs, assuming those are the only dependencies that hold for R.

[10]

(a) Identify the candidate key(s) for R.

(b) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF).

(c) If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.

Perform the above tasks for the following set of functional Dependencies:

6.1. C → D, C → A, B → C

6.2. B → C, D → A

6.3. ABC → D, D → A

6.4. A → B, BC → D, A → C

6.5. AB → C, AB → D, C → A, D → B

Answers:

1. (a) Candidate keys: B

[2]

(b) R is in 2NF but not 3NF.
(c) C → D and C → A both cause violations of BCNF. One way to obtain a (lossless) join preserving decomposition is to decompose R into AC, BC, and CD.

2. (a) Candidate keys: BD
[2]
(b) R is in 1NF but not 2NF.
(c) Both B → C and D → A cause BCNF violations. The decomposition: AD, BC, BD (obtained by first decomposing to AD, BCD) is BCNF and lossless and join-preserving.

3. (a) Candidate keys: ABC, BCD
[2]
(b) R is in 3NF but not BCNF.
(c) ABCD is not in BCNF since D → A and D is not a key. However if we split up R as AD, BCD we cannot preserve the dependency ABC → D. So there is no BCNF decomposition.

4. (a) Candidate keys: A [2]
(b) R is in 2NF but not 3NF (because of the FD: BC → D).
(c) BC → D violates BCNF since BC does not contain a key. So we split up R as in: BCD, ABC.

5. (a) Candidate keys: AB, BC, CD, AD [2]
(b) R is in 3NF but not BCNF (because of the FD: C → A).
(c) C → A and D → B both cause violations. So decompose into: AC, BCD but this does not preserve AB → C and AB → D, and BCD is still not BCNF because D → B. So we need to decompose further into: AC, BD, CD. However, when we attempt to revive the lost functional dependencies by adding ABC and ABD, we that these relations are not in BCNF form. Therefore, there is no BCNF decomposition.