(11)

Monsoon Semester Examination, Session 2022-23

Examination & Semester: **M.Tech (Computer Science & Engineering) I Semester**
Subject: **Advanced DBMS (CSC502)** Time: **3 Hours**
Instructions: Max. Marks: **100**

(a) Answer the questions serially and to the point.

Section A (Marks: 50) – All Questions are COMPULSARY

Q. No.	Question	Marks
1. (a)	Consider a relation *r(A, B, C, D, E, F, G)* satisfies the following functional dependencies: F = {A → B, BC → DE, AEF → G}. Find the closure of {AC} under this set. Is functional dependency ACF → DG implied by F? Show the complete steps.	**5**
(b)	Consider the following relations: *Student(snum: integer, sname: string, major: string, level: string, age: integer)* *Class(name: string, meets_at: string, room: string, fid: integer)* *Enrolled(snum: integer, cname: string)* *Faculty(fid: integer, fname: string, deptid: integer)* The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Write the following queries in SQL. No duplicates should be printed in any of the answers. (i) Find the age of the oldest student who is either Electrical major or enrolled in a course taught by CK. (ii) Find the names of all classes that either meet in room CR1 or have five or more students enrolled.	**5**
(c)	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, do the following: (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. Show each decomposition step (if any) clearly. (i) C → D, C → A, B → C (ii) ABC → D, D → A	**5**
(d)	Consider the following two transactions: *T*1: read(*A*); read(*B*); if *A* = 0 then *B* := *B* + 1; write(*B*); *T*2: read(*B*); read(*A*); if *B* = 0 then *A* := *A* + 1; write(*A*); Let the consistency requirement be *A* = 0 ∨ *B* = 0, with *A* = *B* = 0 the initial values. (i) Show that every serial execution involving these two transactions preserves the consistency of the database. (ii) Show a concurrent execution of *T*1 and *T*2 that produces a nonserializable schedule.	**5**
2. (a)	Suppose that a B+ tree index on branch-city is available on relation *branch (branch-name, branch-city, assets)*, and that no other index is available. List different ways to handle the following selections? (i) $\sigma_{\neg(branch-city<"Dhanbad")}(branch)$ (ii) $\sigma_{\neg(branch-city="Dhanbad")}(branch)$ (iii) $\sigma_{\neg(branch-city<"Dhanbad" \ V \ assests<5000)}(branch)$	**5**
(b)	Assume that only one tuple fits in a block and memory holds at most 3 page frames. Show the runs created on each pass of the sort-merge algorithm, when applied to sort the following tuples on the first attribute: (kangaroo, 17), (wallaby, 21), (emu, 1), (wombat, 13), (platypus, 3), (lion, 8), (warthog, 4), (zebra, 11), (meerkat, 6), (hyena, 9), (hornbill, 2), (baboon, 12).	**5**
(c)	Consider the relations $r_1(A,B,C)$, $r_2(C,D,E)$, and $r_3(E,F)$, with primary keys *A*, *C*, and *E*, respectively. Assume that r_1 has 1000 tuples, r_2 has 1500 tuples, and r_3 has 750 tuples. Estimate the size of $r_1 \bowtie 1 \ r_2 1 \bowtie r_3$, and give an efficient strategy for computing the join.	**5**
3. (a)	A car-rental company maintains a vehicle database for all vehicles in its current fleet. For all vehicles, it includes the vehicle identification number, license number, manufacturer, model, date of purchase, and color. Special data are included for certain types of vehicles: • Trucks: cargo capacity • Sports cars: horsepower, renter age requirement • Vans: number of passengers • Off-road vehicles: ground clearance, drivetrain (four- or two-wheel drive) Construct a schema definition (using Object-based concept) for this database. Must use inheritance where appropriate.	**5**
(b)	Write down an advantage and disadvantage of a distributed lock manager used for the concurrency control in the distributed database. List down all variants of this approach. Explain the protocol which is a generalization of the majority protocol.	**5**
(c)	What is tunable consistency in Apache Cassandra? Explain the read and write consistency.	**5**

Section B (Marks: 30) – Attempt Any TWO

4. (a) Consider the following relations containing airline flight information. **6**

Flights (flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time)
Aircraft (aid: integer, aname: string, cruisingrange: integer)
Certified (eid: integer, aid: integer)
Employees (eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well, every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly. Write the following queries in relational algebra. For such queries, informally explain why they cannot be expressed.

(i) Find the names of pilots certified for some Boeing aircraft.
(ii) Identify the aircraft that can be piloted by every pilot whose salary is more than ₹100,000.
(iii) Is there a sequence of flights from Ranchi to Kolkata? Each flight in the sequence is required to depart from the city that is the destination of the previous flight; the first flight must leave Ranchi, the last flight must reach Kolkata, and there is no restriction on the number of intermediate flights. Your query must determine whether a sequence of flights from Ranchi to Kolkata exists for any input Flights relation instance.

(b) List and draw the various architectures to build a large-scale transaction processing systems. Explain the architecture which is widely used in web servers. **5**

(c) The details of hard disk drive is as follows: **4**
Total Records - 10000; Record Size - 250 bytes per record; Block Size - 1000 bytes per block.
Find out the total search time to search a record from the hard disk drive using indexing (index table entry: 20 bytes, where key is of 10 bytes and pointer of 10 bytes) and the data is:
(i) Ordered.
(ii) Unordered.

5. (a) Consider a relation that is fragmented horizontally by dept-number: **6**

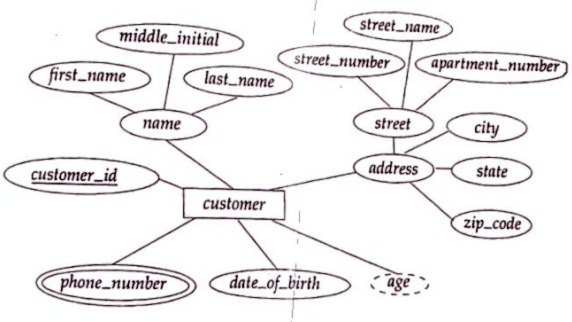
employee (name, address, salary, dept-number)

Assume that each fragment has two replicas: one stored at the Dhanbad site and one stored locally at the plant site. Describe a good processing strategy for the following queries entered at the A site.
(i) Find all employees at the B plant.
(ii) Find the average salary of all employees.
(iii) Find the highest-paid employee at each of the following sites: C, D, E, F.
(iv) Find the lowest-paid employee in the company

(b) To build a highly available distributed system, you must know what kinds of failure can occur. List possible types of failure in a distributed system. Name and explain the protocol that is used to avoid the blocking problem of 2PC and how it handles the above listed possible failures. **5**

(c) Is the expression $ri \ltimes rj$ necessarily equal to $rj \ltimes ri$? Demonstrate with a suitable example. Under what conditions does $ri \ltimes rj = rj \ltimes ri$ hold? **4**

6. (a) Consider the following Entity Diagram, which contains composite, multivalued and derived types of attributes: **6**



(i) Give a schema definition (using Object-based concept) corresponding to the E-R diagram. Use an array to represent the multivalued attribute, and appropriate SQL constructs to represent the other attribute types.
(ii) Give constructors for each of the structured types defined above.

(b) Consider a multidatabase system in which it is guaranteed that at most one global transaction is active at any time, and every local site ensures local serializability. **5**
(i) Suggest ways in which the multidatabase system can ensure that there is at most one active global transaction at any time.
(ii) Show by example that it is possible for a nonserializable global schedule to result despite the assumptions.

(c) Compare MongoDB and Apache Cassandra. **4**

Section C (Marks: 20) – Attempt Any TWO

7. Let us consider the following schema: **10**

Concerts(City, Venue, Year, Month, Date, Song, Album)

In this relation, City (e.g., "Mumbai") and Venue (e.g., "Taj") record where the concert took place and Year, Month, and Date keep track of when the concert took place. The idea is that these five attributes uniquely specify a concert. The attribute Song records the name of a song performed at a concert. You add the attribute Album to record which album the song belongs to. However, after using the database for a few months, you realize that your band (and the real world) have some characteristics that you should model in your database.

(a) Convert each of the next four sentences about Concerts into a functional or a multi-valued dependency. [You can use the first letter of each attribute as an abbreviation for the attribute. Consider each of these four sentences independently. If you cannot write down a functional or a multi-valued dependency, say so, and explain why you cannot, if possible. Do not assume any other constraints, even if they seem reasonable to you.]
   (i) Each song appears in at most one album.
   (ii) A city does not have two venues with the same name.
   (iii) In an effort to please its fans, the band plays at most one song from any album in a given concert.
   (iv) The manager books the band in any city at most once every year.

(b) Assume that all the functional and/or multi-valued dependencies you specified in the part (a) hold in Concerts, as do any dependencies that follow from them. However, no other dependencies hold in Concerts.
   (i) Use Armstrong's axioms to derive the Functional Dependency (FD) *(City, Year, Album) → Song*.
   (ii) What are the keys for Concerts?

(c) You realize that you must decompose Concerts into multiple relations. Here is a decomposition into two relations: *Concerts1(City, Venue, Year, Month, Date)*, and *Concerts2(City, Year, Song, Album)*
   (i) For each relation Concerts1 and Concerts2, state what normal forms it satisfies.
   (ii) For each of the four FDs from the part (a), specify which relation (Concerts1, Concerts2, both, or neither) you can use to verify the FD.

8. Consider the two-phase commit (2PC) protocol with write-ahead undo + redo logging [only the log file needs to be flushed to disk at the time of transaction commit]. Assume we have a system where the only failures involve hosts halting with their disks and logs intact, followed (potentially) by reboots, with no network message loss. **10**

(a) Suppose there is a coordinator C and two participants P1 and P2. Suppose we have the following sequence of events in a 2PC:

C sends Prepare Transaction T1 to P1, P2
P1 sends Prepared to C
P2 sends No to C

   (i) Write an appropriate next message following 2PC, ignoring potential redundant retransmissions.
   (ii) If P1 starts committing T1 immediately after sending a "Prepared" message to C and before waiting for C's response, is the protocol still valid? Briefly explain why or why not.
   (iii) If P2 starts aborting T1 immediately after sending a "No" message to C and before waiting for C's response, is the protocol still valid? Briefly explain why or why not.

(b) Suppose we have the following sequence of events in a 2PC:

C sends Prepare Transaction T2 to P1,P2
P1 sends Prepared to C
P2 sends Prepared to C
P2 crashes
P2 comes back online
C sends Commit Transaction T2 to P1, P2
P1 commits, sends DONE to C

   (i) Assuming standard 2PC operation in the face of the above scenario, what we can expect P2 to do next?
   (ii) P2 must make use of its log on recovery to either commit or abort. Why can P2 expect the log to be complete with respect to T2 operations in this scenario?

9. Write a MongoDB statement for each of the following: **10**

(i) Create a collection by the name "Student" in a database named "myDB".
(ii) Insert a document in the above created collection. The document should include your_name, your_discipline, your_branch, your_semester, your_dob. Assume your branch as CEE.
(iii) Update the above inserted document that changes your branch from CEE to CSE.
(iv) Update documents in the above created collection that changes your_branch from CEE to CSE.
(v) Update documents in the above created collection that changes your_branch to CEE.
(vi) Find the document wherein the student belongs to CEE branch.
(vii) Display the name of all students and their date of birth.
(viii) Display the name of all students belongs to CSE branch.
(ix) Display the name and discipline of all non-CSE students.
(x) Display the name, discipline and date of birth of all non-CSE first semester students.