SVKM'S NMIMS

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT& ENGINEERING

Academic Year: 2022-2023

Program/s: B.TECH. Year: II Semester: IV

Stream/s ; Computer Science & Business Systems

Subject: Database Management Systems Time: 03hrs (_____to_____)

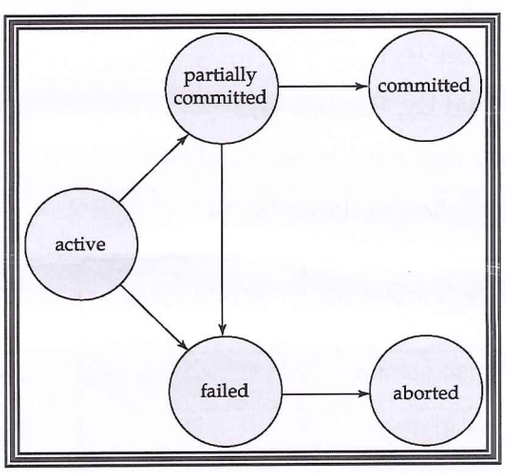
Date:____/____/ 23____. No. of Pages:_____

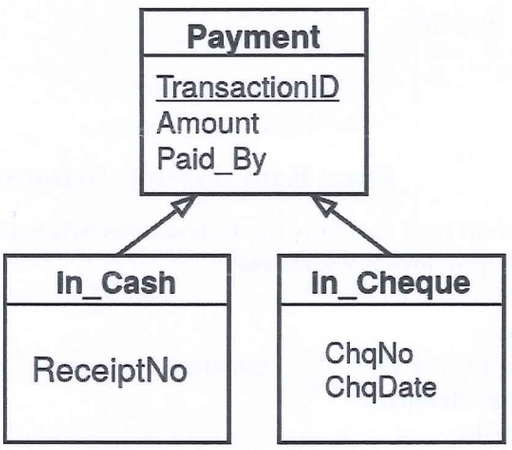
Marks:100

Final Examination-Synoptic

Instructions: Candidates should read carefully the instructions printed on the question paper and on the cover of the Answer Book, which is provided for their use.

1) Question No. 1 is compulsory.
2) Out of remaining questions, attempt any __4___ questions.
3) **In all ___5_ questions to be attempted.**
4) All questions carry equal marks.
5) **Answer to each new question to be started on a fresh page.**
6) **Figures in brackets on the right hand side indicate full marks.**
7) **Assume Suitable data if necessary.**

Q1		Answer briefly:	[20]
CO-2 ; BL-2	a.	Discuss the application where Database management system is useful over the file systems. Any five applications	[5]
CO-4 ; BL-2	b.	Discuss the state transition diagram of transaction management. 	[5]

CO-2 ; BL-3	c.	Describe schema and instances of database. Consider the following Entity Relationship Diagram and decompose into relation schema  Ans. The data which is stored in the database at a particular moment of time is called an instance of the database. The overall design of a database is called schema. Payment(TransactionID, Amount, Paid By) In Cash(TransactionID, ReceiptNo) In Cheque(TransactionID, ChqNo, ChqDate) OR Payment(TransactionID, Amount, Paid By) In Cash(TransactionID, Amount, Paid By, ReceiptNo) In Cheque(TransactionID, Amount, Paid By, ChqNo, ChqDate)	[5]
CO-2 ; BL-4	d.	Given an instance of the Students relation as shown below: **StudentID** | **StudentName** | **StudentEmail** | **StudentAge** | **CPI** 2345 | Shan | shan@math | X | 9.4 1287 | Swati | swati@ee | 18 | 9.5 7853 | Shan | shan@cse | 20 | 9.4	[5]

9876	Swati	swati@mech	19	9.3
8765	Rahul	rahul@civil	20	8.7

i. For (StudentName, StudentAge) to be a key for this instance, the value X should not be equal to what value?

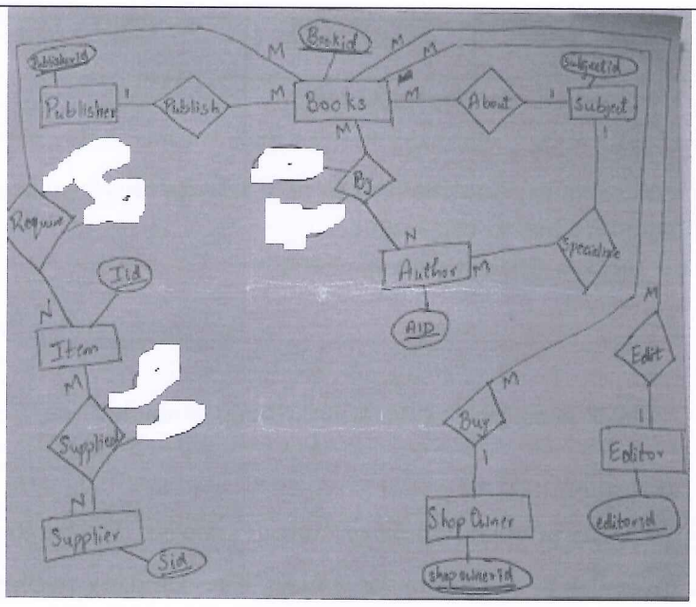
ii. Explain check constraint with an example.

Solution:

i. 20

ii. Check constraint explanation with an example.

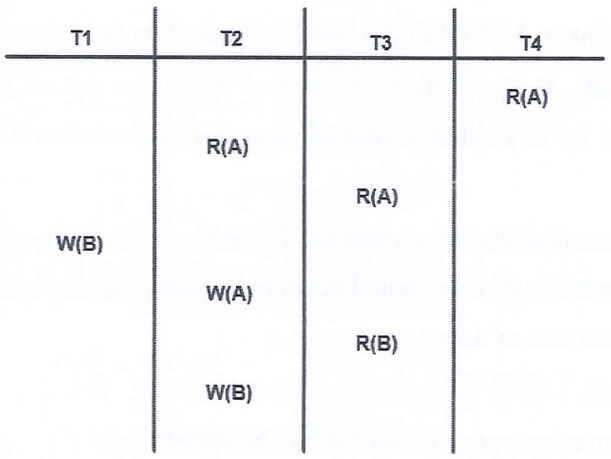
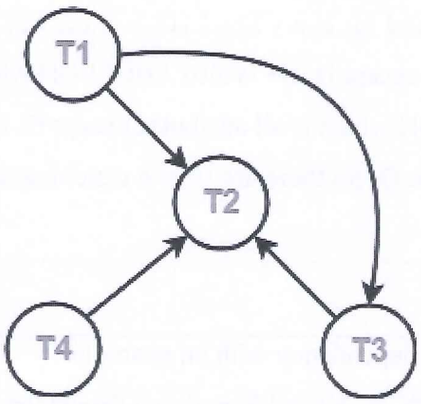
Q2			[20]
CO-2; BL-3	a	A publishing company produces books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editor who, not necessarily being specialist in a particular area, each take sole responsibility for editing one or more book publications. Every book requires some items for publication. These items are supplied by suppliers. One supplier can supply many items. Shop owner buys book from the publisher. Shop owner can buy many books but one book can be bought by one shop owner only. Books are uniquely identified by Bookid. i. Design an Entity Relation for above problem statement. ii. Map the ER diagram into relational schema indicating appropriate primary keys and foreign keys.	[10]

		Ans.  Publisher(Publisherid) Books(Bookid, Publisherid, shopownerid, editorid, subjectid) Author(AID, Bookid, subjectid) Subject(subjectid) Shopowner(shopownerid) Editor(editorid) Item(Iid, Bookid) Supplier(supplierid, Iid)	
CO-1; BL-3	b	List the features and advantages of MongoDB. Ans. Atleast 5 points are expected for features (5 marks) with explanation as well as for advantages (5 marks). **Features of MongoDB –** • **Schema-less Database:** It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection can hold multiple documents and these documents may consist of the different numbers of fields, content, and size. It is not necessary that the one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases. • **Document Oriented:** In MongoDB, all the data stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields(key-value pair) instead of rows and columns which make the data much more	[10]

		flexible in comparison to RDBMS. And each document contains its unique object id. • **Indexing:** In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient. • **Scalability:** MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database. • **Replication:** MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server. • **Aggregation:** It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods • **High Performance:** The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc. **Advantages of MongoDB :** • It is a schema-less NoSQL database. You need not to design the schema of the database when you are working with MongoDB. • It does not support join operation. • It provides great flexibility to the fields in the documents. • It contains heterogeneous data. • It provides high performance, availability, scalability. • It supports Geospatial efficiently. • It is a document oriented database and the data is stored in BSON documents. • It also supports multiple document ACID transition(string from MongoDB 4.0).	

		• It does not require any SQL injection. • It is easily integrated with Big Data Hadoop	
Q3			[20]
CO-3; BL-3	a	Consider the following relational schema Sailors (<u>sid</u>, sname, rating, age) Boats (<u>bid</u>, bname, color) Reserves (<u>sid</u>*, <u>bid</u>*, day) Formulate the SQL queries for the following 1. Find the IDs of sailors who have reserved either a red or a green boat using set operation. **(select *R.sid*** **from *boats* as *B*, *reserves* as *R*** **where *B.bid* = *R.bid* and B.color = 'red')** **union** **(select *R.sid*** **from *boats* as *B*, *reserves* as *R*** **where *B.bid* = *R.bid* and B.color = 'green')** 2. Find the count of sailors in the database. **select count**(*) **from** *sailors* 3. Find the average age of sailors having rating of 10 **select avg**(age) **from** *sailors* **where** *rating* = 10 4. For each rating, find the average age of the sailors **select** *rating*, **avg**(*age*) **from** *sailors* **group by** *rating* 5. For each rating, find the age of the youngest sailor with age at least 18. SELECT S.rating, MIN (S.age) FROM Sailors S WHERE S.age >= 18 GROUP BY S.rating	[10]

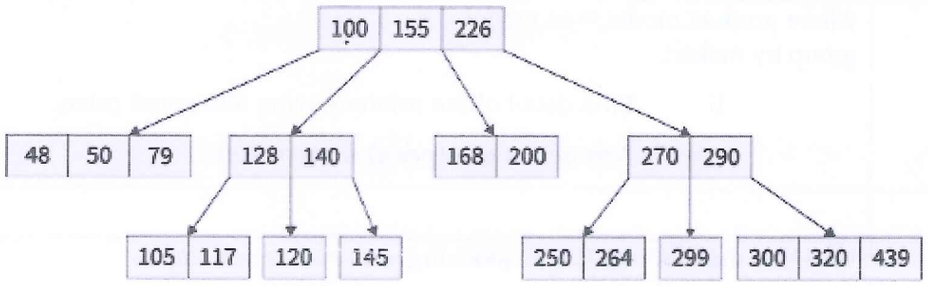
CO-2; BL-4	b	Discuss the importance of normalization and also explain Boyce - Codd normal form with an example. Relation R has eight attributes ABCDEFGH. Fields of R contain only atomic values. F = {CH -> G, A -> BC, B -> CFH, E -> A, F -> EG} is a set of functional dependencies (FDs) hold for R. Find all candidate keys does the relation R have? Ans. To correct duplicate data and database anomalies. [3mark] To avoid creating and updating any unwanted data connections and dependencies. To prevent unwanted deletions of data. To optimize storage space. To reduce the delay when new types of data need to be introduced. To facilitate the access and view of data to users and product tools. BCNF is the advance version of 3NF. It is stricter than 3NF. [3mark] A table is in BCNF if every functional dependency X → Y, X is the super key of the table. For BCNF, the table should be in 3NF, and for every FD, LHS is super key A+ is ABCEFGH which is all attributes except D. B+ is also ABCEFGH which is all attributes except D. E+ is also ABCEFGH which is all attributes except D. F+ is also ABCEFGH which is all attributes except D. So there are total 4 candidate keys AD, BD, ED and FD.	[10]
Q4			[20]
CO-2; BL-4	a	Discuss partial and transitive functional dependency with an example. Given a relation R(P, Q, R, S, T, U, V, W, X, Y) and Functional Dependency set FD = { PQ → R, P → ST, Q → U, U → VW, and S → XY}, determine whether the given R is in 3NF? If not convert it into 3 NF.	[10]
	Ans.	Partial and transitive dependency explanation (4 mark) R1(P, S, T), R2(Q, U), R3(U, V, W), R4(S, X, Y), and R5(P, Q, R). (3+3mark)	

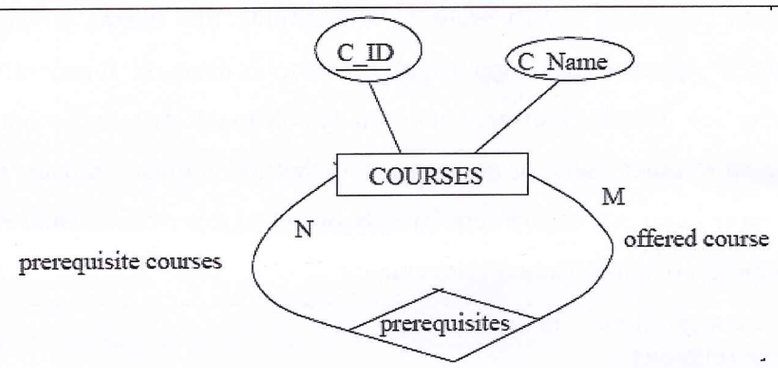
CO-4; BL-4	b	Discuss view serializability with an example Check whether the given schedule S is conflict serializable or not.  	[10]
	Ans.	View serializability is a concept that is used to compute whether schedules are View-Serializable or not. A schedule is said to be View-Serializable if it is view equivalent to a Serial Schedule. [5+5 mark]  • Clearly, there exists no cycle in the precedence graph. • Therefore, the given schedule S is conflict serializable Two schedules S1 and S2 are said to be view equivalent if they satisfy the following conditions: 1. Initial Read 2. Updated Read 3. Final Write	

		In schedule S1, if Ti is reading A which is updated by Tj then in S2 also, Ti should read A which is updated by Tj There may be some schedules that are not Conflict-Serializable but still gives a consistent result because the concept of Conflict-Serializability becomes limited when the Precedence Graph of a schedule contains a loop/cycle. In such a case we cannot predict whether a schedule would be consistent or inconsistent. As per the concept of Conflict-Serializability, We can say that a schedule is Conflict-Serializable (means serial and consistent) if its corresponding precedence graph does not have any loop/cycle. But, what if a schedule's precedence graph contains a cycle/loop and is giving consistent result/accurate result as a conflict serializable schedule is giving? So, to address such cases we brought the concept of View-Serializability because we did not want to confine the concept serializability only to Conflict-Serializability.	
Q5			[20]
CO-4; BL-4	a	i. Discuss cascading rollback with an example. Ans. i. If in a schedule, failure of one transaction causes several other dependent transactions to rollback or abort, then such a schedule is called as a Cascading Schedule or Cascading Rollback or Cascading Abort ii. Compare the recoverable and irrecoverable schedule with an example. .Ans.Recoverable Schedules: Schedules in which transactions commit only after all transactions whose changes they read commit are called recoverable schedules. In other words, if some transaction Tj is reading value updated or written by some other transaction Ti, then the commit of Tj must occur after the commit of Ti. Irrecoverable Schedule: The table below shows a schedule with two transactions, T1 reads and writes A and that value is read and written by T2. T2 commits. But later on, T1 fails. So we have to rollback T1. Since T2 has read the value written by T1, it should also be rollbacked.	[10]

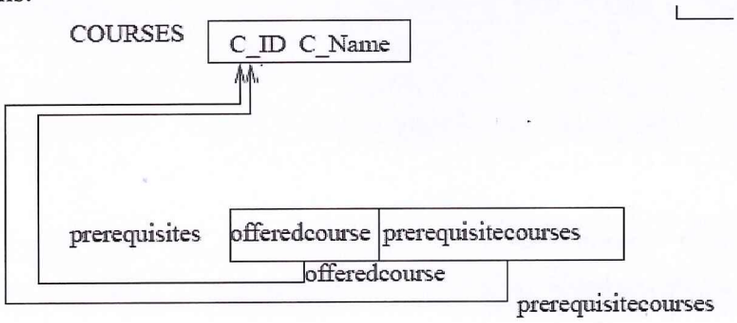
		But we have already committed that. So this schedule is irrecoverable schedule. When Tj is reading the value updated by Ti and Tj is committed before committing of Ti, the schedule will be irrecoverable.	
CO-2; BL-4	b	Suppose that we have the following relational database schemas Product(maker, model) PC(model, speed, ram, hd, cd, price) Laptop(model, speed, ram, hd, screen, price) Printer(model, color, type, price) with the assumption that model number are unique over all manufacturer and product types. This means that the above four relations have model as one of their keys. Provide Relational algebra-based expressions for the following question: a) Which manufacturers produce printer and laptop? b) List the price of all the PC, laptop, and printer. c) Find all manufactures that produce printers but do not produce PC. Ans. NOTES: R: Product(maker, model) S: PC(model, speed, ram, hd, cd, price) T: Laptop(model, speed, ram, hd, screen, price) U: Printer(model, color, type, price) **(∝ : Join)** a) $\Pi_{maker}(R \propto U) \cap \Pi_{maker}(R \propto T)$ b) $\Pi_{model, price}(S) \cup \Pi_{model, price}(T) \cup \Pi_{model, price}(U)$ c) $\Pi_{maker}(R \propto U) - \Pi_{maker}(R \propto S)$ Write sql statement for the following: i. Find for each manufacturer the maximum price of a PC. Ans. select maker, max(price) from product, pc	[10]

		where product.model = pc.model group by maker; ii. Find detail of the printer having maximum price. Ans. select max(price) from printer	
Q6			[20]
CO-4; BL-4	a	Illustrate the working B-tree indexing with suitable example. Ans. B-tree in DBMS is an m-way tree which self balances itself. Due to their **balanced structure**, such trees are frequently used to manage and organise enormous databases and facilitate searches. In a B-tree, each node can have a maximum of n child nodes. In DBMS, B-tree is an example of **multilevel indexing**. Leaf nodes and internal nodes will both have record references. B Tree is called Balanced stored trees as all the leaf nodes are at same levels. Working of B trees : • When B-tree is used for database indexing, it becomes a little more complex because it has both a key and a value. The value serves as a reference to the particular data record. A payload is the collective term for the key and value. • For index data to particular key and value, the database first constructs a unique random index or a **primary key** for each of the supplied records. The keys and record byte streams are then all stored on a B+ tree. The random index that is generated is used for indexing of the data. • So this indexing helps to decrease the searching time of data. In a B-tree, all the data is stored on the leaf nodes, now for accessing a particular data index, database can make use of binary search on the leaf nodes as the data is stored in the sorted order. • if indexing is not used, the database reads each and every records to locate the requested record and it increases time and cost for searching the records, so B-tree indexing is very efficient. Examples of B-Tree Suppose there are some numbers that need to be stored in a database, so if we store them in a B-tree in DBMS, they will be stored in a sorted order so that the searching time can be logarithmic.	[10]

		 The above data is stored in sorted order according to the values, if we wanna searching for the node containing the value 1, so the following steps will be applied: • First, the parent node with key having data 3 is checked, as 1 is less than 3 so the left children node of 3 is checked. • In left children, there are 2 keys, so it will check from the leftmost key as the data is stored in sorted order. • Leftmost element is having key value as 1 which match the element to be searched, so thats how we the element we wanted to search.	
CO-4; BL-2	b	i. Describe the concept of Specialization and Generalization in extended ER [5mark] Ans. Generalization – Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it. It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common. Specialization – In specialization, an entity is divided into sub-entities based on their characteristics. It is a top-down approach where higher level entity is specialized into two or more lower level entities ii. Give the relational schema of the following ER diagram. [5mark]	[10]



Ans.



Q7			[20]
CO-4; BL-2		Write short note : i. Two phase locking protocol in concurrency control ii. SQL Injection attack Ans. **Concurrency control Protocols** Two-phase-locking protocol •**Basic 2PL**: Transaction is said to follow the *two-phase-locking protocol* if all locking operations precede the *first* unlock operation –Expanding (growing) = first phase –Shrinking = second phase •During the shrinking phase no new locks can be acquired! –Downgrading ok –Upgrading is not •**Conservative 2PL** (static) 2PL: Lock all items needed BEFORE execution begins by predeclaring its read and write set –If any of the items in read or write set is already locked (by other transactions), transaction waits (does not acquire any locks) –Deadlock free but not very realistic	[10]

SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.

CO-3; BL-2

b

Explain the concept of joins in SQL.

Consider two relations:

loan

loan-number	*branch-name*	*amount*
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

borrower

customer-name	*loan-number*
Jones	L-170
Smith	L-230
Hayes	L-155

Give the output of following: select * from

a) loan natural inner join borrower

Ans.

Loan-number	Branch-name	amount	Customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

b) loan left outer join borrower on loan.loan-number = borrower.loan-number

Ans.

Loan Number	Branch-name	amount	Loan Number	Customer Name
L-170	Downtown	3000	L-170	Jones
L-230	Redwood	4000	L-230	Smith
L-260	Mumbai	30000	-	-

c) loan full outer join borrower using (loan-number)

Loan Number	Branch-name	amount	Loan Number	Customer Name
L-170	Downtown	3000	L-170	Jones
L-230	Redwood	4000	L-230	Smith
L-260	Mumbai	30000	-	-
-	-	-	L-155	Hayes

[10]