****

**UNIVERSITY COLLEGE OF ENGINEERING**

**(BIT CAMPUS), TIRUCHIRAPPALLI-620 024**

**Second Internal Assessment Test**

**Department of CSE / IT**

**Subject Code / Name:** CS8492 / Database Management Systems

**Date Duration : 26.02.2019** & 1.30Hrs **Marks: 50**

**Degree/Branch:** B.TECH/IT Sec’B’ **Year / Sem:** II / IV

**ANSWER KEY**

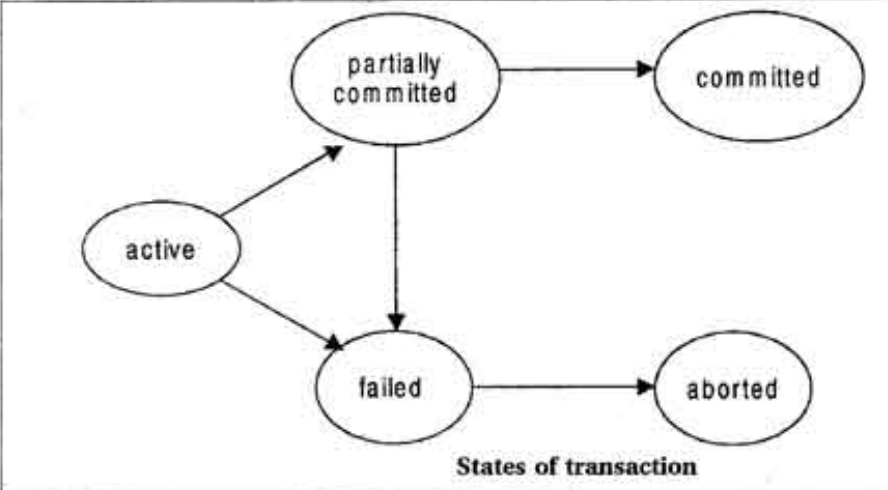
**PART-A**

**Answer *All* Questions 7 x 2 = 14**

1. **What are the ACID properties?**

(Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee database transactions are processed reliably. In the context of databases, a single logical operation on the data is called a transaction. For example, a transfer of funds from one bank account to another, even though that might involve multiple  
changes (such as debiting one account and crediting another), is a singletransaction.

1. **What are the states of transaction?**



1. **Define schedule and its types.**

A **schedule** can have many transactions in it, each comprising of a number of instructions/tasks. Serial **Schedule** − It is a **schedule** in which transactions are aligned in such a way that one transaction is executed first.

1. Serial Schedule

2. Non-serial Schedule

3. Serializable schedule

1. **Define concurrency control and what are the problems of concurrency control?**

Process of managing simultaneous execution of transactions in a shared database, to ensure the Serializability of transactions, is known as concurrency control.

1. Lost updates
2. Dirty read
3. Unrepeatable read
4. **Define the different modes of locks**

The modes of lock are:  
🙣 Shared Lock  
🙣 Exclusive Lock

1. **Define the phases of two phase locking protocol.**

***Growing phase:* a transaction may obtain locks but not release any lock.  
*Shrinking phase:* a transaction may release locks but may not obtain any new locks**

1. **How to construct the serializability precedence graph? and write the rules**
2. Create a node Ti → Tj if Ti executes write (Q) before Tj executes read (Q).
3. Create a node Ti → Tj if Ti executes read (Q) before Tj executes write (Q).
4. Create a node Ti → Tj if Ti executes write (Q) before Tj executes write (Q).

**PART-B**

**Answer any three Questions 3 x 10 = 30**

1. **What is concurrency control protocol? Explain the different types of locking protocol techniques in details**

In the concurrency control, the multiple transactions can be executed simultaneously.

It may affect the transaction result. It is highly important to maintain the order of execution of those transactions.

**1. Shared lock:**

* **It is also known as a Read-only lock. In a shared lock, the data item can only read by the transaction.**
* **It can be shared between the transactions because when the transaction holds a lock, then it can't update the data on the data item.**

**2. Exclusive lock:**

* **In the exclusive lock, the data item can be both reads as well as written by the transaction.**
* **This lock is exclusive, and in this lock, multiple transactions do not modify the same data simultaneously.**

**There are four types of lock protocols available:**

* 1. **Simplistic lock protocol**
  2. **Pre-claiming Lock Protocol**
  3. **Two-phase locking (2PL)**
  4. **Strict Two-phase locking (Strict-2PL)**

1. **Write short notes on**
2. **What is Timestamp method? Explain the concept in details**

* The Timestamp Ordering Protocol is used to order the transactions based on their Timestamps. The order of transaction is nothing but the ascending order of the transaction creation.
* The priority of the older transaction is higher that's why it executes first. To determine the timestamp of the transaction, this protocol uses system time or logical counter.
* The lock-based protocol is used to manage the order between conflicting pairs among transactions at the execution time. But Timestamp based protocols start working as soon as a transaction is created.
* Let's assume there are two transactions T1 and T2. Suppose the transaction T1 has entered the system at 007 times and transaction T2 has entered the system at 009 times. T1 has the higher priority, so it executes first as it is entered the system first.
* The timestamp ordering protocol also maintains the timestamp of last 'read' and 'write' operation on a data.

**Basic Timestamp ordering protocol works as follows:**

1. Check the following condition whenever a transaction Ti issues a **Read (X)** operation:

* If W\_TS(X) >TS(Ti) then the operation is rejected.
* If W\_TS(X) <= TS(Ti) then the operation is executed.
* Timestamps of all the data items are updated.

2. Check the following condition whenever a transaction Ti issues a **Write(X)** operation:

* If TS(Ti) < R\_TS(X) then the operation is rejected.
* If TS(Ti) < W\_TS(X) then the operation is rejected and Ti is rolled back otherwise the operation is executed.

**Where,**

**TS(TI)** denotes the timestamp of the transaction Ti.

**R\_TS(X)** denotes the Read time-stamp of data-item X.

**W\_TS(X)** denotes the Write time-stamp of data-item X.

1. **What is an optimistic technique? Explain the concept in details.**
2. **Read phase:** In this phase, the transaction T is read and executed. It is used to read the value of various data items and stores them in temporary local variables. It can perform all the write operations on temporary variables without an update to the actual database.
3. **Validation phase:** In this phase, the temporary variable value will be validated against the actual data to see if it violates the serializability.
4. **Write phase:** If the validation of the transaction is validated, then the temporary results are written to the database or system otherwise the transaction is rolled back.

Here each phase has the following different timestamps:

**Start(Ti):** It contains the time when Ti started its execution.

**Validation (Ti):** It contains the time when Ti finishes its read phase and starts its validation phase.

**Finish(Ti):** It contains the time when Ti finishes its write phase.

* This protocol is used to determine the time stamp for the transaction for serialization using the time stamp of the validation phase, as it is the actual phase which determines if the transaction will commit or rollback.
* Hence TS(T) = validation(T).
* The serializability is determined during the validation process. It can't be decided in advance.
* While executing the transaction, it ensures a greater degree of concurrency and also less number of conflicts.
* Thus it contains transactions which have less number of rollbacks.

1. **i) What is RAID? Explain the concept in details**

**RAID refers to redundancy array of the independent disk. It is a technology which is used to connect multiple secondary storage devices for increased performance, data redundancy or both. It gives you the ability to survive one or more drive failure depending upon the RAID level used.**

**It consists of an array of disks in which multiple disks are connected to achieve different goals.**

**RAID technology**

**There are 7 levels of RAID schemes. These schemas are as RAID 0, RAID 1, ...., RAID 6.**

**These levels contain the following characteristics:**

* **It contains a set of physical disk drives.**
* **In this technology, the operating system views these separate disks as a single logical disk.**
* **In this technology, data is distributed across the physical drives of the array.**
* **Redundancy disk capacity is used to store parity information.**
* **In case of disk failure, the parity information can be helped to recover the data.**

**Standard RAID levels**

**RAID 0**

**RAID 1**

**RAID 2**

**RAID 3**

**RAID 4**

**RAID 5**

**RAID 6**

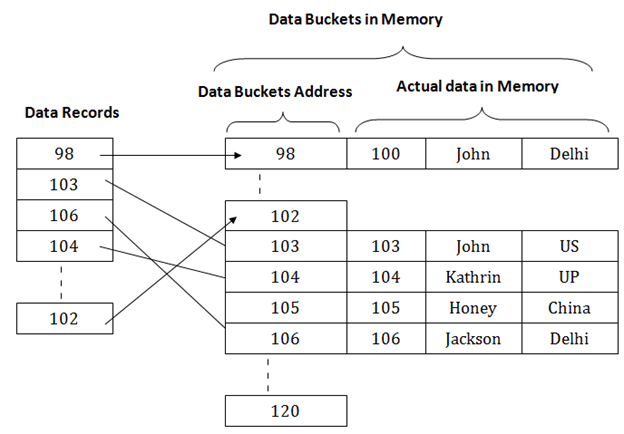
**ii) What is hashing? Explain the types in details**

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anamolies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

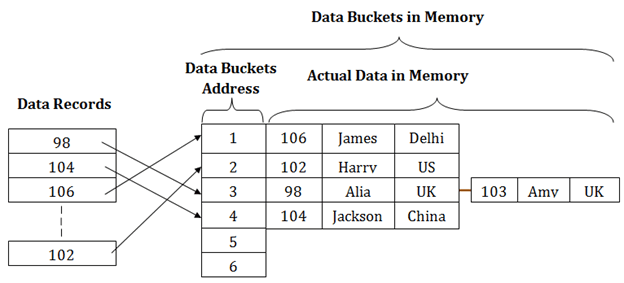
In a huge database structure, it is very inefficient to search all the index values and reach the desired data. Hashing technique is used to calculate the direct location of a data record on the disk without using index structure.

In this technique, data is stored at the data blocks whose address is generated by using the hashing function. The memory location where these records are stored is known as data bucket or data blocks.

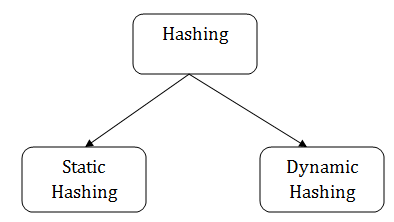
In this, a hash function can choose any of the column value to generate the address. Most of the time, the hash function uses the primary key to generate the address of the data block. A hash function is a simple mathematical function to any complex mathematical function. We can even consider the primary key itself as the address of the data block. That means each row whose address will be the same as a primary key stored in the data block.

The above diagram shows data block addresses same as primary key value. This hash function can also be a simple mathematical function like exponential, mod, cos, sin, etc. Suppose we have mod (5) hash function to determine the address of the data block. In this case, it applies mod (5) hash function on the primary keys and generates 3, 3, 1, 4 and 2 respectively, and records are stored in those data block addresses.



Types of Hashing:

1. **What is serializable schedule? Explain the types of serializability with an example.**

The serializability of schedules is used to find non-serial schedules that allow the transaction to execute concurrently without interfering with one another.

It identifies which schedules are correct when executions of the transaction have interleaving of their operations.

A non-serial schedule will be serializable if its result is equal to the result of its transactions executed serially.

**Conflict Serializable Schedule**

* A schedule is called conflict serializability if after swapping of non-conflicting operations, it can transform into a serial schedule.
* The schedule will be a conflict serializable if it is conflict equivalent to a serial schedule.

**Conflicting Operations**

The two operations become conflicting if all conditions satisfy:

1. Both belong to separate transactions.
2. They have the same data item.
3. They contain at least one write operation.

**View Serializability**

1. A schedule will view serializable if it is view equivalent to a serial schedule.

2.If a schedule is conflict serializable, then it will be view serializable.

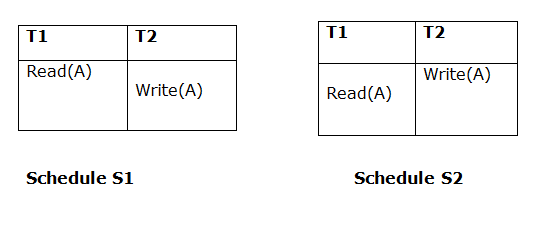
3.The view serializable which does not conflict serializable contains blind writes.

**View Equivalent**

Two schedules S1 and S2 are said to be view equivalent if they satisfy the following conditions:

**1. Initial Read**

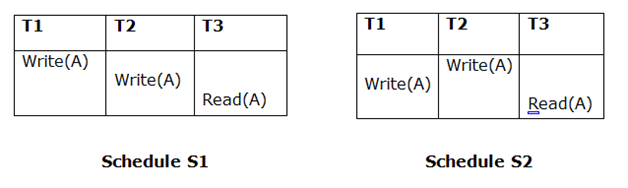
An initial read of both schedules must be the same. Suppose two schedule S1 and S2. In schedule S1, if a transaction T1 is reading the data item A, then in S2, transaction T1 should also read A.

Above two schedules are view equivalent because Initial read operation in S1 is done by T1 and in S2 it is also done by T1.

**2. Updated Read**

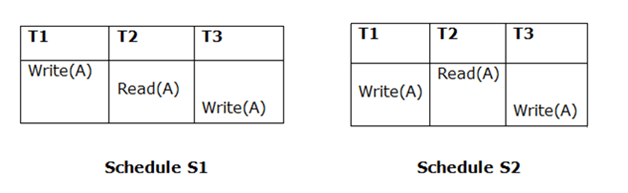
In schedule S1, if Ti is reading A which is updated by Tj then in S2 also, Ti should read A which is updated by Tj.

Above two schedules are not view equal because, in S1, T3 is reading A updated by T2 and in S2, T3 is reading A updated by T1.

3. Final Write

A final write must be the same between both the schedules. In schedule S1, if a transaction T1 updates A at last then in S2, final writes operations should also be done by T1.

Above two schedules is view equal because Final write operation in S1 is done by T3 and in S2, the final write operation is also done by T3.