Q1 a) Primary key and candidate key

Primary key is a unique and non null key which identify a record uniquely in table.

From many candidate keys, the database designer select one candidate key for his database called as primary key

A single column primary key is called simple key a multicolumn primary key is called composite primary key, Not null value allowed, value can't be modified.

candidate key :- It is an attribute whose values uniquely identify each record of a table. A table can have more than one candidate key. candidate key satisfy some conditions which are as follows:

- Different records cannot be identical
- Subset of the candidate key cannot be a key.

Q1b)

Q1b) file system

- Software that manages the data file in a computer system
- Helps to store a collection of raw data files into the hard disk
- Tasks such as storing, retrieving and searching are done manually. so it is difficult to manage data.
- Has data inconsistency
- There is more redundant data
- provides more security to the data
- Handling is easy
- Backup & recovery process is no efficient because it is not possible to recover the lost data.
- Appropriate to handle data of a small-scale organization

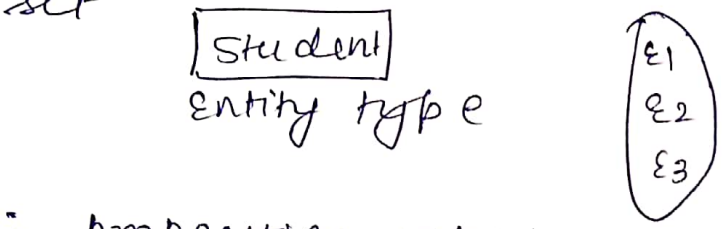
DBMS

- Software to create and manage databases.
- Helps to easily store retrieve & manipulate data in a database.
- operations such as updating searching, selecting data is easier.
- There is low data redunancy.
- comparitively less data security.
- Handling is complex.
- Has a sophisticated backup & recovery
- Suitable for medium to large organization.

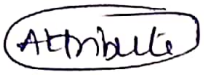
Q1 c) E-R model is used to model the logical view of the system from data perspective which consist of these component. Pg (3)

Entity, Entity type, Entity set

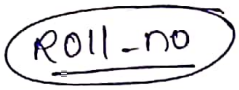
Entity may be an object with a physical existence, an entity is an object of Entity type and set of all entities is called as entity set



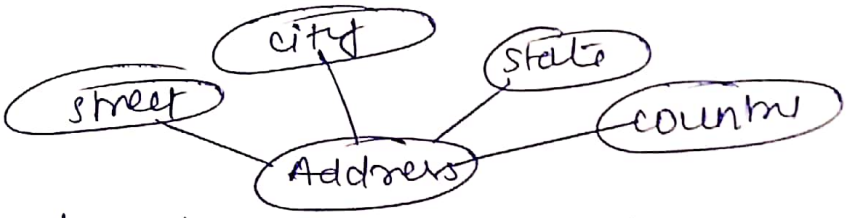
Attributes : properties which define entity type



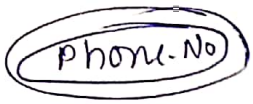
Key - attribute - uniquely identify each entity



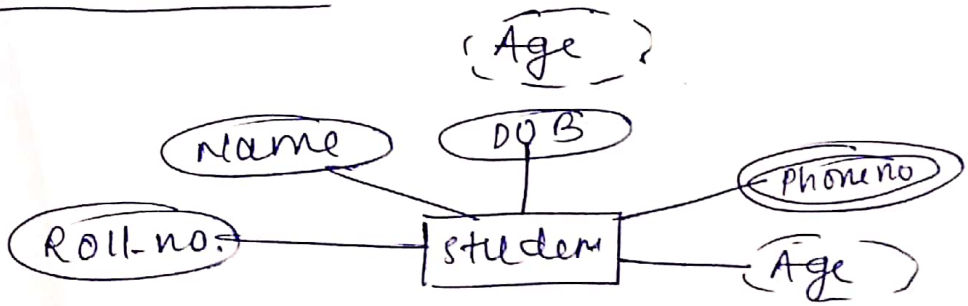
composite attribute - composed of many other attribute



Multivalued attribute :- consisting more than one value

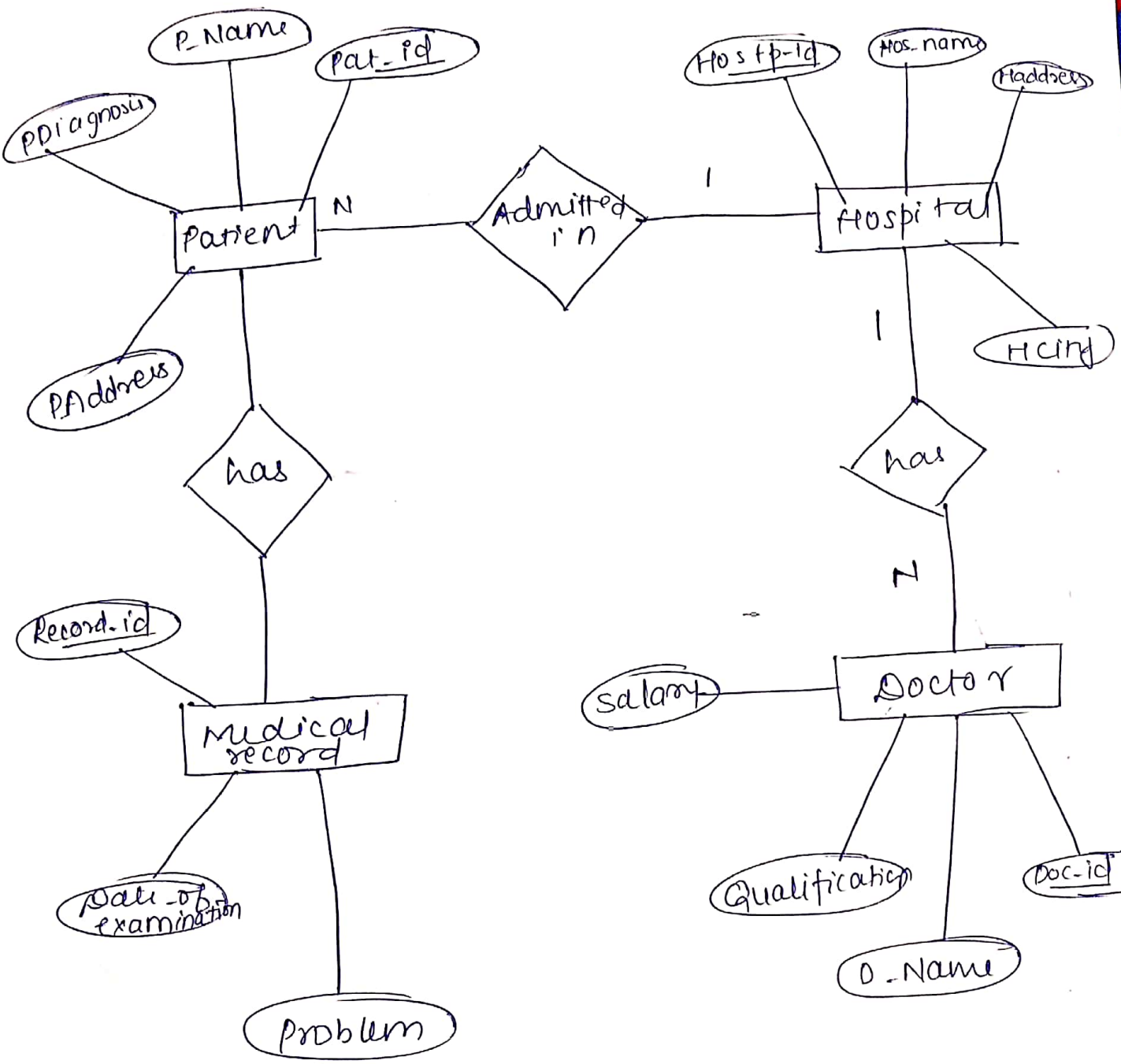


Derived attribute - derived from other attribut



Q1 c)

Hospital Management System



Q) 4 DDL commands

DDL is a standardized language with commands to define the storage groups, different structures and object in a database

1. Create syntax CREATE TABLE [table name] ([column definitions]) [table parameters]

It is used to create a new table or database.

2. Drop - used to delete table

syntax - DROP object type object name;

drop commands to delete objects in a database. drop commands cannot be undone, so once an object is deleted, it cannot be recovered.

3. Alter - used for to a make modifications to database objects. such as indexes locations or stogroups

Syntax ALTER object type object name parameters;

4) Rename is used to modify the name of a database table.

Q2 b)

Relational Algebra	Relational calculus
• It is a procedural language	• It is a declarative (non-procedural language
• Relational algebra means how to obtain the result	• It means what result have to obtain
• In relational algebra the order is specified in which the operations have to be performed	• In relational calculus, the order is not specified
• Relational algebra is independent of the domain	• Relational calculus can be domain dependent because of domain relational calculus.
• It is nearer to a programming language	• It is not nearer to programming language but to natural language
• It is one of the languages in which queries can be expressed but the queries should also be expressed in relational calculus to be relationally completed.	• For a database language to be relationally complete, the query written in it must be be expressible in relational calculus.

The relational algebra include some basic operation

1) Select (σ)
2) project (Π)
3) union ($\cup$)
4) set difference ($-$)
5) Cartesian product ($\times$)
6. Rename (ρ)

Relational calculus has 2 variation

1) Tuple Relational calculus
2) Domain Relational calculus

$\{t \mid P(t)\}$ t: set of tuples, P: is the condition which is true for the given set of tuples.

Eg of Relational calculus

$\{R \mid \exists T \in Author(T.article = 'database' AND R.name = T.name)\}$

$\{T.name \mid Author(T) AND T.article = 'database'\}$

Eg of Relational algebra

$\sigma_{topic = "Database}(tutorials)$

$\sigma_{topic = "Database" and author = "guru99"}(Tutorials)$

Q2 c)

1. Cursor

A cursor can be basically referred to as a pointer to the context area - context area is a memory area that is created by oracle when SQL statement is processed The cursor is there responsible for holding the rows that have been returned by a SQL statement. Thus the PL/SQL controls the context area by the help of cursor. An active set is basically the set of rows that the cursor holds. There are of 2 types Explicit & Implicit cursor.

Advantages

- They are faster
- Better concurrency control can be achieved
- They are helpful in performing the row by row wise validation each row

Disadvantages.

i) They use more resources each time

ii) More no. of network round trips can degrade the performance and reduce speed.

Trigger

A trigger is basically a program which gets automatically executed in response to some events such as modification in the database. Some of the events for their execution are DDL statement, DML statement or any database operation. Triggers are thus stored within the database and come into action then specific condition match

Advantage

- helpful in keeping the track of all the changes within the database
- Also help in maintaining the integrity constraints

Disadvantage

- They are very difficult to write which makes the debugging also difficult
- To much use of the triggers or writing complex code can slow down the performance.

6 types of triggers

~~UNSAT~~ BEFORE INSERT, AFTER INSERT, BEFORE UPDATE AFTER UPDATE, BEFORE DELETE, And AFTER DELETE.

Index has the main function of retrieving data from table much quicker. Indexes are created by users on columns that may be accessed frequently. This enables the user to get information quickly from the table and can be created on a single column or even a group.

- The index is a type of data structure It is used to locate and access the data quickly

Structure

search key	Data Reference

Q3 a) Functional dependency refers to the relation of one attribute of the database to another. With the help of functional dependency, the quality of the data in the database can be maintained.

The symbol for representing functional dependency is → (arrow).

Like $X \dashrightarrow Y$

The left side of the above FD diagram is called determinant, and the right side is the dependent.

Eg SIN ---> Name, Address, Birthdate

Q3 c)(i)

Multivalued dependency is having the presence of one or more rows in a table. It implies the presence of one or more other rows in that same table. A multivalued dependency prevents fourth normal form. A multivalued dependency involves at least three attributes of a table. It is represented with a symbol →→ in DBMS.

$a \rightarrow\rightarrow b$ a is multi-valued dependent on b

Conditions for MVD

Any attribute say a multiple define another attribute b; if any legal relation r(R), for all pairs of tuples t1 & t2 in R such that

$t_1[a] = t_2[a]$

Then there exists t3 & t4 in R such that

$t_1[a] = t_2[a] = t_3[a] = t_4[a]$

$t_1[b] = t_3[b]$; $t_2[b] = t_4[b]$

$t_1 = t_4$; $t_2 = t_3$

	a: Name	b: Project	c: Hobby
t_1	Geeks	MS	Reading
t_2	Geeks	Oracle	Music
t_3	Geeks	MS	Music
t_4	Geeks	oracle	Reading

$t_1[a] = t_2[a] = t_3[a] = t_4[a]$ = Geeks

cond^n 1 satisfied

$t_1[b] = t_3[b]$ = MS

$t_2[b] = t_4[b]$ = oracle } cond^n 2 satisfied.

$t_1 = t_4$ = Reading

& $t_2 = t_3$ Music } cond^n 3 satisfied.

hence $a \rightarrow\rightarrow b$

name →→ project

Q3 c) (i) Lossless join decompositions

Loss-less join decomposition is a process in which a relation is decomposed into two or more relations. This property guarantees that the extra or less tuple generation problem does not occur and no information is lost from the original relation during the decomposition. It is also known as non-additive join decomposition

When the sub relations combine then the new relation must be the same as the original relation was before decomposition

The decomposition is lossless when it satifies the following statement :—

1) If the union of the sub relation R_1 & R_2 then it must contain all the attributes that are available in original relation R.

2) Intersection of R_1 & R_2 should not be null.

3) The ~~sub~~ ~~rela~~ common attribute must be a super key of sub relations either R_1 or R_2.

Q3 d)

Sid	cid	S-name	C-name	Grade	Faculty	F-Phone
1	IS318	Adams	Database	A	Howard	60192
1	IS301	Adams	Program	B	Langely	45869
2	IS318	Jones	Database	A	Howard	60192
3	IS318	Smith	Database	B	Howard	60192
4	IS301	Baker	Program	A	Langely	45869
4	IS3018	Baker	Database	B	Howard	60192

1NF

(Repeating & Multivalued

SID and CID → Grade

2NF

Partial dependence

Sid	Cid	Grade
1	IS318	A
1	IS301	B
2	IS318	A
3	IS318	B
4	IS301	A
4	IS318	B

Sid → sname

Sid	S-name
1	Adams
2	Jones
3	Smith
4	Bakey

Cid → Cname
Cid → Faculty

cid	C-name	Faculty	F-phone
IS318	Database	Smith	60192
IS301	Program	Johnson	4586 9
IS318	Database	Smith	60192
IS318	Database	Smith	60192
IS301	Program	Johnson	4586 9
IS318	Database	Smith	60192

3NF (Transitive dependency)

These two tables stays the same

S-id	C-id	Grade
1	1S318	A

1	1S301	B
2	1S318	A
3	1S318	B
4	1S301	A
4	1S318	B

S-id	S-name
1	Adams
2	Jones
3	Smith
4	Baker

However, there is a transitive dependency

Faculty → F-phone

C.id	C-name	FID
15318	Data base	1
15301	Program	2
15318	Database	1
15318	Database Program	1
15301	Program	2
15318	Database	1

F-ID	Faculty	F-phone
1	HOD SCI	60192
2	Langely	45869

Q4 a)

A transaction is a single logical unit of work that accesses and possibly modifies the contents of a database. Transactions access data using read & write operations.

In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called ACID properties.

A (Atomicity)
C (consistency)
I (Isolation)
D (Durability)

Atomicity: By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e, transaction is considered as one unit and either runs to completion or is not executed at all. It involves 2 operation Abort and commit.

eg

Before x: 500		y: 200
	Transaction T	
T1		T2
Read(x)		Read(Y)
x := x − 100		Y := Y + 100
write(x)		write(Y)
After x: 400		Y: 300

If the transaction fails after completion of T1 but before completion of T2, then the amount has been deducted from x but not added to Y. This results in a inconsistent database state.

→ Consistency: This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database

Eg: Refering to the eg. above the total amount before & after the transaction must be maintained.

Total before T occurs = 500 + 200 = 700

Total after T occurs = 400 + 300 = 700.

the data base is consistent. Inconsistency occurs in case T1 completes but T2 fails.

→ Isolation: This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. changes occuring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been commited. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed.

→ Durability: - This property ensures that once the transaction has completed execution, the updates and modification to the data base are stored in & written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction thus are never lost.

Q4b)

A schedule is serialized if it is equivalent to a serial schedule. A concurrent schedule must ensure it is the same as if executed serially means one after another It refers to the sequence of actions such as read, write, abort, commit are performed in serial manner

Serializibility helps in identifying which non-serial schedule gives consistent result as given by serial schedule.

There are two types of serializability

- View serializability
- Conflict serializability.

Conflict serializability : It orders any conflicting operations in the same way as some serial execution. A pair of operations is said to conflict if they operate on the same data item and one of them is a write operation

That means :-

- Read i(x) - Read j(x) → non-conflict read-read operation.
- Read i(x) - write j(x) → conflict read-write operation
- Write i(x) - read j(x) → conflict write-read operation.
- Write i(x) - write j(x) → conflict write-write operation.

Conflicting operations: Two operations are said to be conflicting if all condition satisfy
- They belong to different transaction
- The operate on same data item
- At least one of them is write operation.

Conflict Equivalent:- Two schedules are said to be conflict equivalent when one can be ~~transferred to~~ transformed to another by swapping non-~~equi~~ conflicting operations.

View Serializability

- A schedule will view serializable if it is view equivalent to a serial schedule.
- If a schedule is conflict serializable then it will be view serializable.
- The view serializable which doesnot conflict serializable contains blind writes.

View Equivalent: The two schedules S1 & S2 are said to be view equivalent if they satisfy the following conditions.

1. **Initial Read** :- An initial read must be the same. Suppose 2 schedules S1 & S2 .:In S1, Ti is reading a data item A, then in S2 transaction Ti should also read A

T_1	T_2
Read (A)	write(A)

T_1	T_2
	write(A)
read(A)	

2. **Update 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

T_1	T_2	T_3
write(A)	write (A)	Read (A)

T_1	T_2	T_3
	write(A)	
write(A)		Read(A)

3. **Final write**: A final write must be the same between both the schedules. In schedule S1, if a transaction Ti updates A at last the in S2, final writes operation should also be done by Ti

R-read
W-write

T_1	T_2	T_3
write(A)	R(A)	W(A)

T_1	T_2	T_3
	R(A)	
W(A)		W(A)

Q4 d)

(1) checkpoints :—

The checkpoint is used to declare a point before which the DBMS was in the consistent state, and all transactions are committed.

During transaction execution. After execution, transaction log files will be created.

upon reaching the checkpoint, the log file is destroyed by saving its update to the database. Then a new log is created with upcoming execution operations of the transaction and it will be updated until the next checkpoint and the process continues.

Why do we need checkpoints?

Whenever transaction logs are created in a real-time environment, it eats up's lots of storage space. Also keeping track of every update & its maintenance may increase the physical space of the system.

The methodology utilized for removing all previous transaction logs and storing them in permanent storage is called checkpoints.

24d(ii) Deadlock handling :-

Deadlock is a situation where a process or set of processes is blocked, waiting for some other resource that is held by some other resource (waiting process).

The 4 condn to occur deadlock
1) Mutual exclusive
2) hold & wait
3) No-~~preemption~~ pre-emption
4) circular ~~hold~~ wait

Methods for handling deadlocks

3 approaches to deal with deadlocks

1) Deadlock prevention
2) Deadlock avoidance
3) Deadlock detection

① **Deadlock prevention** - The strategy of deadlock prevention is to design the system in such a way that the possibility of deadlock is excluded. Indirect method prevent the occurrence of one or three necessary condn of deadlock. ie mutual exclusion, no-preemption Direct method prevent the occurence of circular wait.

But this prevention does not yield good result becoz :-

- long waiting time required
- inefficient use of allocated resource.

No-pre-emption - techniques for 'no pre-emption are'

if a process that is holding some resource, request another resource that cannot be immediately allocated to it, the all resource currently being held are released.

Circular wait one way to ensure that this condition never hold

2) Deadlock avoidance - This approach allows the 3 necessary conditions of deadlock but makes judicious choice to assure that deadlock point is never reached. It allows more concurrency than avoidance detection a decision is made dynamically whether the current resource allocation request will lead to deadlock. Two techniques to avoid deadlock

1. Process initiation ~~deadlock~~ denial
2. Resource allocation denial

3. Deadlock detection - Deadlock detection is used by employing an algorithm that tracks the circular waiting and killing one or more processes so that deadlock is removed. The system do state is examined periodically to determine if a set of processes is deadlocked. This technique does not limit resource access or restrict process action It never delays the process initiation and facilitates online handling.

Q5 a)

Multiple Granularity means hierarchically breaking up the database into blocks that can be locked and can be tracked needs what needs to lock and in what fashion. Such a hierarchy can be represented graphically as a tree.

~~For eg - consider the~~

The multiple granularity protocol enhances concurrency and reduces lock overhead.

It maintains the track of what to lock & how to lock

It makes easy to decide either to lock a data item or to unlock it.

Concurrency control is provided in a database to:

(i) enforce isolation among transactions
(ii) preserve database consistency through consistency preserving execution of transactions.
(iii) resolve read-write and write-read conflicts.

various concurrency control techniques are:

1. Two lock based protocol :-

Granularity is the level and type of information that lock protects is called locking granularity.

2 types of locks → shared locks → if Ti has obtained shared locks on data item A then Ti can read A but cannot modify A, denoted by 'S'

→ Exclusive locks → It is read-write lock that means if Ti having the lock on A then it can do both read & write, denoted by X

Two-phase locking protocol :- Locking is an operation which secures: permission to read, OR permission to write a data item. Two phase locking is a process used to gain ownership of shared resources without creating the possibility of deadlock.

It prevents deadlock from occuring in distributed systems by releasing all the resources it has acquired, if it is not possible to acquire all the resources required without waiting for another process to finish using a lock.

A transaction in the two phase locking protocol can assume one of 2 phase:

- Growing phase : In this phase a transaction can only acquire locks but cannot release any lock.

(ii) shrinking phase: In this phase a transaction can only release locks but cannot acquire any.

2. Time-stamp ordering protocol:
A timestamp is a tag that can be attached to any transaction or any data item, which denotes a specific time on which the transaction or the data item had been used in any way.
A timestamp can be implemented in 2 ways. One is to directly assign the current value of the clock to the transaction or data item. The other is to attach the value of a logical counter that keeps increment as new timestamps are required. The time stamp can be of 2 types –

(i) W-timestamp(x): This means the latest time when the data item x has been written into.

(ii) R-timestamp(x): This means the latest time when the data item x has been read from.

3. Multiversion concurrency control: Multiversion schemes keep old versions of data item to increase concurrency. Multiversion 2 phase locking: Each successful write results in the creation of a new version of the data item written. Timestamps are used to label the versions. When a read(x) operation is issued, select an appropriate version of x based on the timestamp of the transaction.

(A) Validation concurrency control : The optimistic approach is based on the assumption that the majority of the database operations do not conflict. The optimistic approach requires neither locking nor time stamping techniques. Instead, a transaction is executed without restrictions until it is committed. Using an optimistic approach, each transaction moves through 2 or 3 phase i.e, read, validation and write.

(i) During read phase, the transaction reads the database, executes the needed computations and makes the updates to a private copy of the database values. All update operations of the transactions are recorded in a temporary update file, which is not accessed by the remaining transactions.

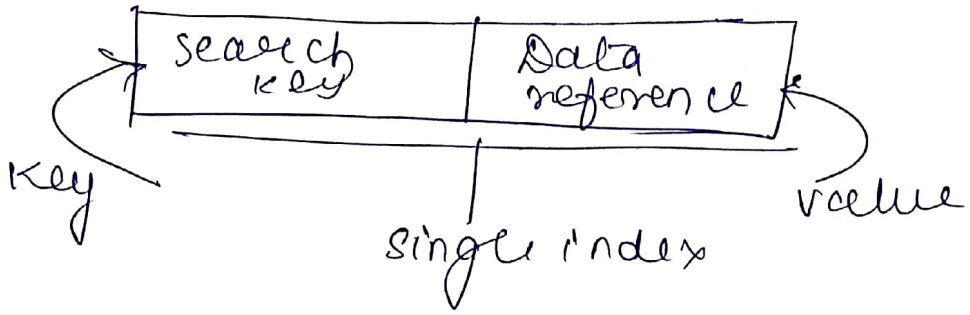
(ii) During the validation phase, the transaction is validated to ensure that the changes made will not affect the integrity and consistency of the database. If the validation test is positive, the transaction goes to a write phase, if it is -ve, the transaction is restarted and the change are discarded.

(iii) During the write phase, the changes are permanently applied to the database.

Q5 d)

Indexing is a way to optimize the performance of a database by minimizing the no. of disk access required when a query is processed.

structure



1. Sequential file organization or ordered index file → In this, the indices are based on a stored ordering of the values.

2. Dense Index — For every search key value in the data file, there is an index record.

3. Sparse index - The index record appears only for a few items in data file

4. Hash file organization - The value being distributed uniformly across a range of buckets.

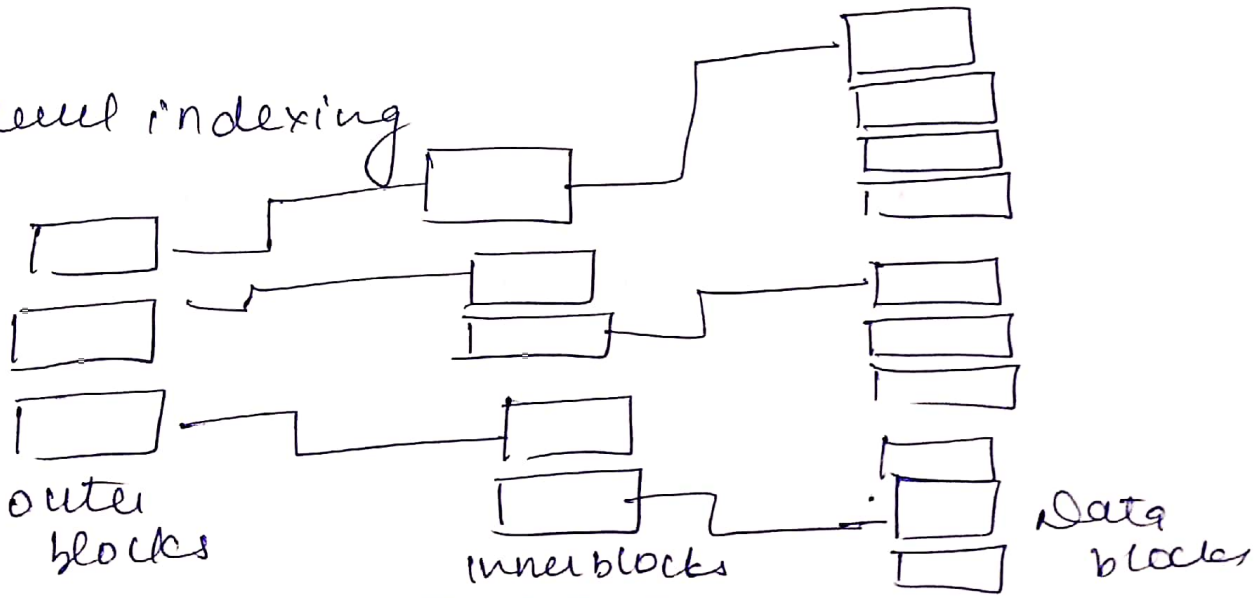
- Clustered Indexing
when one or more than 2 records are stored in a same file these type of storing known as cluster indexing.

Primary Indexing is a type of clustered Indexing wherein the data is sorted according to the search key.

eg students studying in each semester are grouped 1, 2, 3 semester etc.

- Non clustered Indexing tells us where the data lies it gives us a list of virtual pointers or references on the location where the data is actually stored
It requires more time as compared to the clustered index because some amount of extra work is done in order to extract work is done in order to extract the data by further following the pointer

- Multilevel indexing



(i) Primary index – primary index is defined on an ordered data file. The data file is ordered on a key field.
The key field is generally the primary key of the relation.

(ii) Secondary Index – secondary index may be generated from a field which is a candidate key has a unique value in every record, or a non-key with duplicate values.

(iii) Multilevel Indexing – With the growth of the size of the database, indices also grow. As the index is stored in the main memory. The multilevel indexing segregates the main block into various smaller blocks so that the same can stored in a single block.

Pg (34)