Unit No 04

* Purpose of Normalization

- To remove the duplicate data
- To remove the DB anomalies from the relation table.
- helps to reduce redundancy & complexity by examining new data types used in table
- To protect the data and to make the DB more flexible by eliminating redundancy and inconsistent dependency.

- Normalisation is the process of efficiently organising data in a database.
- It is step by step decomposition of complex record into simple record.
- This process also called canonical synthesis.
- This is ↓ by ↓ decompⁿ & DB desiger may not normalize relation to the highest possible normal form
- The relations may be left in a lower normal form like 2NF, which may cause some penalties like data anomalies.

Defenⁿ :- Normalisation is a process of designing a consistent database by minimizing redundancy & ensuring data integrity through decomposition which is lossless.

→ Row level Duplicacy

| SID | Sname | Age |
|---|---|---|
| 1 | RAM | 20 |
| 2 | Varun | 25 |
| 1 | RAM | 20 |

By using primary key

→ column leven Duplicasy

| SID | sname | cid | Cname | F_Id | F_name |
|---|---|---|---|---|---|
| 1 | RAM | C1 | DBMS | F1 | John |
| 2 | RAVI | C2 | JAVA | F2 | Bob |
| 3 | NITIN | C1 | DBMS | F1 | John |
| 4 | Amit | C1 | DBMS | F1 | John |

Normalization

coloumn ki value exactly same then what's the

~~Prop~~ Problem :-

1) Insertion Anomaly
2) Deletion Anomaly
3) Updation Anomaly

(university table)

| SID | Sname | Cid | Cname | FID | Fname | Salary | Hr |
|---|---|---|---|---|---|---|---|
| 1 | RAM | C1 | DBMS | F1 | John | 30000 | 12 |
| 2 | RAVI | C2 | JAVA | F2 | Bob | 40000 | 10 |
| 3 | Nitin | C1 | DBMS | F1 | John | 30000 | 12 |
| 4 | Amit | C1 | DBMS | F1 | John | 30000 | 13 |
| 5 | Vaish ✓ | | | | | | |
| | | C10 | MBBS | | | | |
| | | can't | Insert | F3 | Rash | | |
| | | | | can't insert | | | |

- We can't insert data without inserting primary key (SID) in table / above table

Insertion Anomaly ; An Insertn Anomaly comes / Arises when certain attributes can't be added into the database without the presence of other attributes.

In above table, it is not possible to add a row in university table for an student who doesn't exist in table.

Delete Anomaly :- A deletion anomaly occurs when you delete a record that may contain attributes that shouldn't be deleted.

Update Anomaly :- Updating some records in faculty table may exist in course table
For course, the no. of alloted hrs to a cource is repeated several times Hence, if no. of hrs has to be change
Any omissions will lead to inconsistencies.

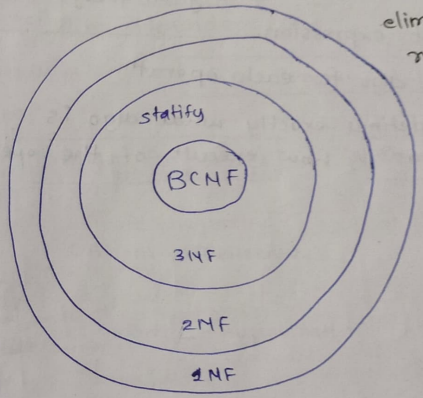
* BCNF

* Boyce - Codd normal form

→ Developed in 1974 by Boyce & Edgar

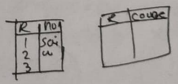
→ Advance version of 3NF
Stronger form of 3NF

→ FD $X \rightarrow Y$, X is the super key of table.
L.H.S of each FD should be super key / candidate key




eliminates all the redundancy that can be discover based on FDs

1NF → Table should not contain any multivalued Attribute.



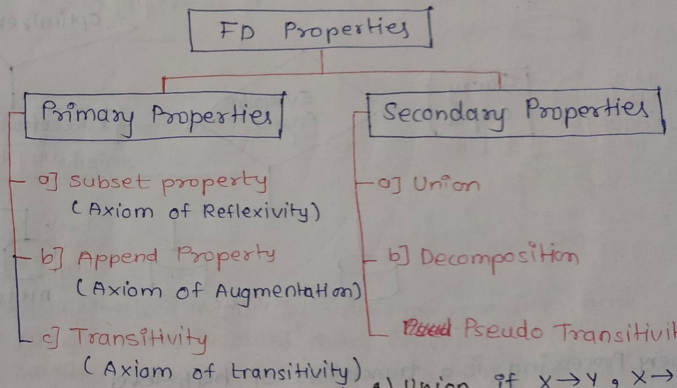
## Query Optimization :-

- We don't except the user to write their queries so that they can be processed efficiently
- Rather, we expect the sys. to construct a query evaluation plan that minimizes the cost of query evaluation.
- Alternate ways of evaluating a given query.
  1) Equivalent expression
  2) Different algo for each operath.
- An evaluath plans defines exactly what algo is used for each oprath & how executh of the operath is co-ordinated.

Q1 State and explain Armstrong's axioms and it property.

→ FD Properties [Armstrong's Axioms / closures of FD)

- Given that Relation R[X, Y, Z, W]; represent a table R with set of indivisible attributes X, Y, Z and W.
- It is possible to derive many properties of functional dependencies.
- Axioms are nothing but rules of interface which Provieds a simple technique for reasoning about functional dependencies




1) Primary Properties.

a) subset property — if y is subset of x, then $X \rightarrow Y$

which can be reffered as X is functionally dependent on Y

b] Append Property. if $X \rightarrow Y$ then $XZ \rightarrow YZ$

It is possible to appen z on both side of FD provieds that th same table

c] Transitivity if $X \rightarrow Y$ & $Y \rightarrow Z$ then $X \rightarrow Z$

It is PT Utransi if all X,Y,Z are part of same table

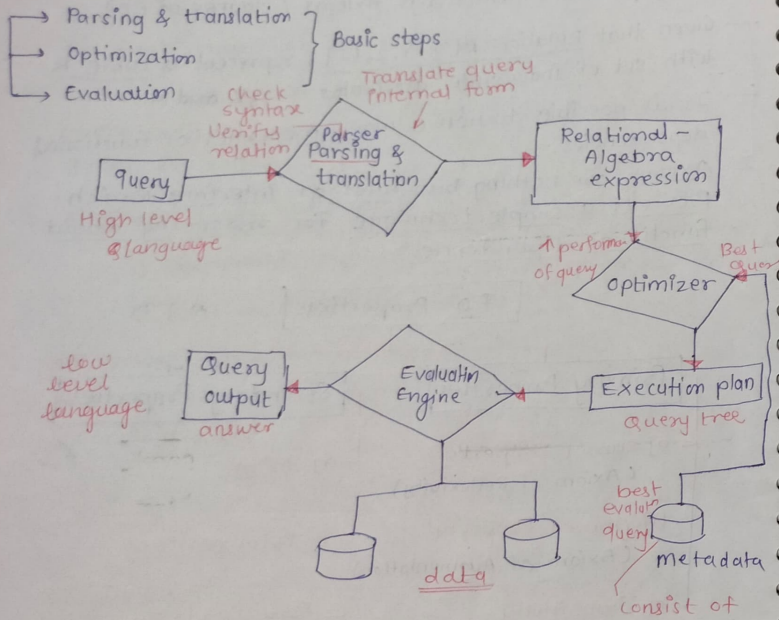
a) Union if $X \rightarrow Y$, $X \rightarrow Z$ then $X \rightarrow YZ$

b) Decomposition if $X \rightarrow YZ$ then $X \rightarrow Y$ & $X \rightarrow Z$

c) Pseudo $X \rightarrow Y$ & $YZ \rightarrow W$

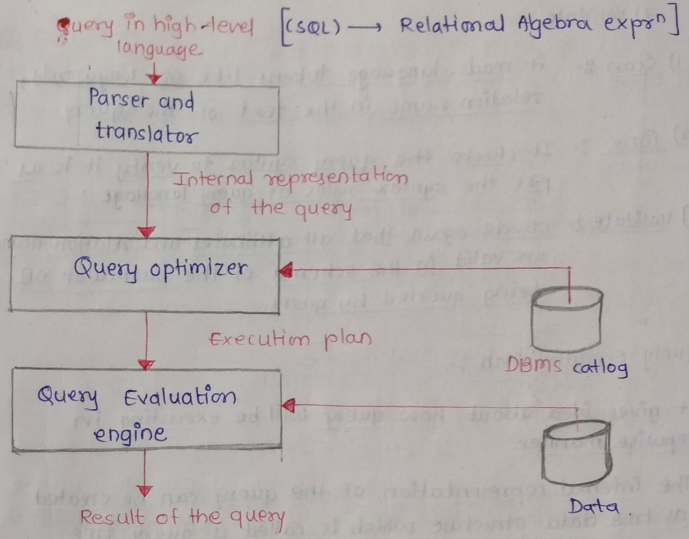
For given relation R[X,Y,Z,W

# Query Processing :-




- Query Processing is a translation of high-level high level query into low level expression
- It is sp step wise process that can be used at the physical level of the file system
- query optimization and actual execution of the query to get the the result.
- It requires basic concepts of relational algebra and file structure.

## * Query Processing




- The activity involved in parsing, validating, execution and optimising a query is called Query Processing.
- It inccludes transaction of high-level queries into low level expression that can be used at the physical level of the file system.
- Parser :- checks the syntax of Query, verifies attribube & relⁿ.
- Relational QP refers to the range of activities which inclu in extracting data from a database using a db query.

Query Optimisation.

selecting the most effeient query evaluation plan from among the many statergies usually possible for a given query.

2] A Query Processing involves below steps

1) Scan
2) Parse
3) Validate

1) Scan :- it reads language tokens like SQL keywords, relation name in the text of the query

2) Parse :- It checks the query syntax to verify it is as per the syntax rules of query language.

3) Validate :- checks again that all attributes and relation name are valid in the schema of the particular DB being queried by query.

3) Query Execution Plan :-

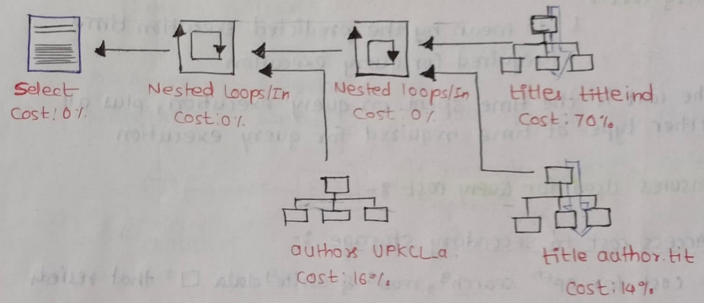
- It gives idea about how query will be executing in stepwise manner.
- The internal representation of the query can be created as tree data structure which is called a query tree.
- The DBMS must find all alternative execution strategies for retrieving the result of the query from the database
- Ex. Dept info can be accessed in follow way
  [I] by Dept table , [II] Emp-Dept_Join_table
- Query can have many possible execution strategies.
- Process of selecting a suitable strategy for processing a query is known as query optimization.
- Query optimization is achievable for simple queries but it becomes very complex for difficult queries.
- Each DBMS has a number of general database access algorithm that such as SELECTION , PROJECTION or JOIN or combination of these operation.

Q What are the measures of cost ?

→

Measure of a Query Cost.

1) Query cost
2] Measures used for query cost
3] other measures used for Query cost




Sample Query cost in SQL server

* other measure used for Query cost
  - Disk access
  - CPU cycles
  - Transist time in n/w

1) CPU cost is difficult to calculate
2) CPU speed normally increasing at Faster than Disk speed
3) CPU cost is relatively lower than Disk cost
4) Primarly considered with distributed / II sys.
5] Disk accen cost :-
  - No. of seeks ⟹ Diff. for Random I/o / sequen
  - No. of blocks read
  - No. of blocks write

  cost(W) > cost(R)

No. of seek (N) ; cost = N x Avg - seek time
block read ; cost = N x Avg block read cost
block write ; cost = N x Avg block read cost

No. of blocks transfer from disk & no. of seeks.

$t_T$ = time to transfer 1 block

$t_S$ = time for 1 seek

cost for b block transfer plus S seeks.

$$\text{Cost query} = b * t_i + S * t_S$$

* **Query cost** :-

L do mean by the predicted execution time required for query execution

- The cost is the time spent on query execution, plus all other types of time required for query execution

+ **Measures Used for Query cost** :-

1) Access cost to secondary storage :-
   L cost for opⁿ search^g, read^g & write^g data □^s that resieds 2^ry storage or disk, where as contiguous allocation used to store block on same disk or it is scattered on the disk which affect the access cost.

2) Storage cost :- cost for storing intermediate files which generated byan execut^n strategy for the query

3) Computation cost :- cost for memory opt^n during query execut^n

4) Memory usage cost :- cost → memory buffers required during query execut^n

5) Communicat^n cost :- cost to send query results 1 DB site to other site.

6) Number of different resources : → used → pointer, disk accen.

7) Data transfer rate

8) Cost of scanning disk segment containing tuples

9) Cost models for different index access methods

# Transaction Managment

Q) What is transaction

— It is a program unit whose execution may change the content of db.

→ DB before transact^n → DB after transact^n { consistent state }

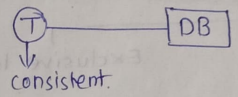
→ Ex :-

| A | B |
|---|---|
| (300) | (400) |
| Read (A) | Read (B) |
| A = A - 100 | B = B + 100 |
| Write (A) | Write (B) |

before transact^n = A + B = 700

After transact^n = B + A = 700

its in consisten state.




if transaction statisfy the ACID Properites, your transaction is maintaing DB consistency.

## ACID Properties

A successful transaction changes the db from 1 consistent state to other.

A – Atomicity
C – Consistency
I – Isolation
D – Durability

1) Atomicity :-

ACID
- Atomicity → The entire transaction takes place at once or doesn't happen at all [Transaction manager]
- Consistency → The database must be consistent before and after the transaction. Appl^n programmer [Concurrency Contr manage
- Isolation → Multiple Transaction occur independentl without interference [Recovery Manager]
- Durability → The changes of a successful transact occurs even if the system failure occur

★ Two phase locking Protocol

1) lock-based Protocol

↳ A lock gra guarantees exclusive use of data items to a current transactn

- Access data items (lock acquires)
- After completion of transactn (Release lock).

Types of locks
- shared lock (lock-S)
- Exclusive lock (lock-X)

Types of Locks

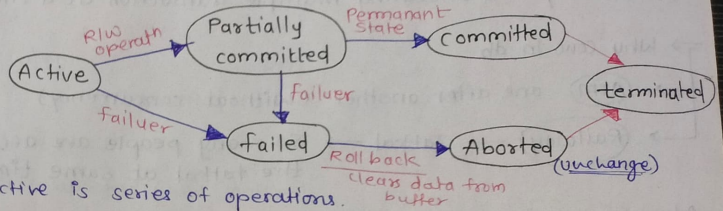
Shared lock (lock-S) — Read data item values.

Exclusive lock (lock-X) — both R&W data items

| $T_1$ | $T_2$ | |
|---|---|---|
| Lock-X(B) | | Exclusive lock (R/W) |
| R(B) | | |
| B-CO | | |
| W(B) | | |
| Unlock(B) | | |
| | Lock-S(B) | shared lock (R) |
| | R(B) | |
| | Unlock(B) | |

Any no. of transactn can hold shared lock blut exclusive lock can be hold only by one transaction at a time

states of transaction :-




- Active is series of operations.
  - ↳ Transactn ⟶ buffer (local mem) {Partially committed}
  - ↳ If transactn satisfy ACID propesty the transaction committed ⟶ values store in DB & then terminated.

1] Active

2] Partially committed

3] failed

4] Committed

5] Aborted

6] Terminated.

* Concurrency Control :- It is a process of managing simultaneous execution of transaction in shared db

Concurrency ⟶ Executing multiple transaction at a time

⇒ Why conc in db
- (ATM) one after another (without concurrency)
- (Railway) ⟶ Tatkal ⟶ so many people are accessing the tatkal at same time

Advantages :-
(a) Waiting time ↓
(b) Response time ↓
(c) Resource Utilization ↑
(d) efficiency ↑

⟶ simultaneous execution of transaction over a shared db can create several data integrity & consistency problems.

⟶ 3 main flaws. & a) lost updates.
b) Uncommited data.
c) inconsistent Retrivals.

⇒ conflicts (WR, RW, WW) serializability of transaction
Serial executions.

a) Reading Uncommitted data (WR conflict, "dirty read")

A = 5
(A+1)
A = 6

| T1 | T2 |
|---|---|
| R(A) 5 | |
| W(A) | |
| | R(A) 6 |
| R(B) | C |
| W(B) | |
| Abort | |

[Just stored in Buffer]
T2 [Read A as 6 without commiting] T1 ⟶ dirty read

Failed unchanged Rollback.

Now
T1 [A = 5] T2 [A = 6]

DB is not consistent

2) Unrepeatable Read (RW conflict) :

| A = 10 | T1 | T2 |
|---|---|---|
| A | | |
| | R(A) | |
| A+5 | | R(A) 10 |
| | W(A) | |

3) lost updated (WW conflict) :

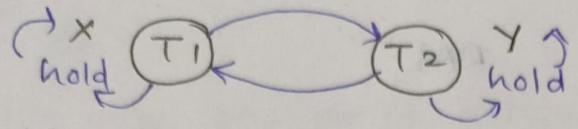
| A = 10 | T1 | T2 |
|---|---|---|
| A = 10+1 | R(A) | |
| 11 | W(A) | W(A) blind write as 60 |
| A = 11 | C | C A = 1 & 60 |

Purpose of Concurrency cntrl
- To Enforce Isolation
- To preserve db consistency
- To resolve conflicts.

Techniques :-
1) locking
2) Time stamp based
3) Optimistic
4) Multiversion

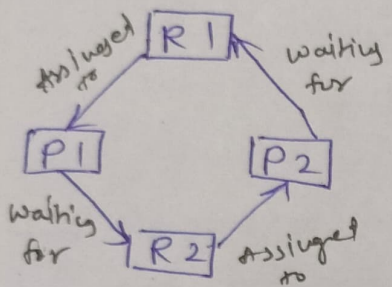
Deadlock :- It is a situation which 2 or more transactions are waiting for one another to give up locks.

Successfull when T1 complete X & Y operation




After complitn of x transaction T1 waiting for y transaction & T2 waiting for x tras'

T1 is waiting for T2
T2 is waiting for T1
} this situation is Deadlock



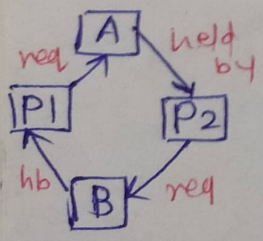

| T1 | T2 |
|---|---|
| lock-x(x) | |
| R(x) | |
| W(x) | |
| | Lock-x(y) |
| | R(y) |
| | W(y) |
| Lock-x(y) } waiting for T2 to release lock on Y | |
| R(y) | |
| W(y) | |
| | Lock-x(x) } waiting for T1 to release lock on x |
| | R(x) |
| | W(x) |

This is Deadlock.

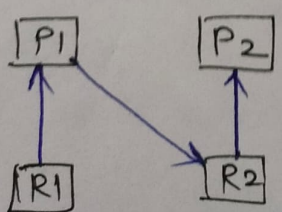
* Necessary Condith.

1) Hold and wait
2) Mutual Exclusion
3) No preemption
4) Circular wait

Hold & wait

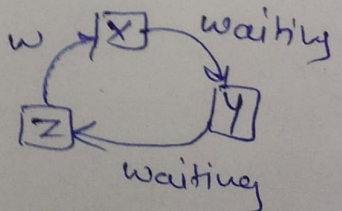



Mutual Exclusn
Process is reqting which is already lock by other Process




No preemptn
work completed then release the lock

Circular wait



Two phase locking (2PL)

⇒ both locks & unlocks.

lock point - The point at which the growing phase ends.

2PL

Growing Phase: locks are obtained but Not released

Shrinking Phase: locks are released but new locks are required.

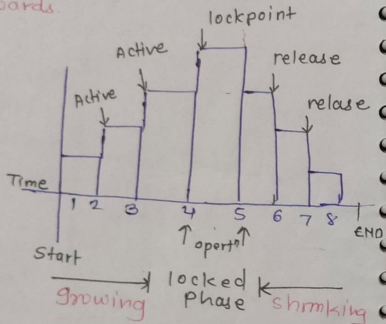
lockpoint ← onwards

final lock to growing Phase

⇒ Holding locks

⇒ locking to early

⇒ Penalty to other transactn




| | T1 | T2 |
|---|---|---|
| | $l_1(A)$ | $l_2(A)$ |
| | $r(A)$ | $r_2(A)$ |
| A=A+100 | | A=2*A |
| | $W_1(A)$ | $W(A)$ |
| | $l_1(B)$ | $l_2(B)$ |
| | $U_1(A)$ | $U_2(A)$ |
| | $r(B)$ | $r_2(B)$ |
| =B+100 | | B=2*B |
| | $W_1(B)$ | $W_2(B)$ |
| | $U_1(B)$ | $U_2(B)$ |

lock operations
ecedes all unlock
erations

Variation of 2PL Protocol

- Conservation (static) 2PL
- Strick 2PL
- Rigrous 2PL

shared / Exclusive can't be released until commite.

1) After commite only it release the lock.

2) until commit her exclusive lock can't be released

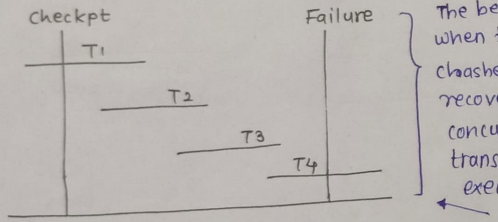
- Disadvantages of 2PL
  - More Management - Need / require extra backend managment
  - Potential shipping Delays
  - Increased Risk

* checkpoint

- A mechanism where all the previous logs are removed from the system & stored permanently in a storage disk.
- Checkpoint declares a point before which the DBMS was in consistent state, and all the transaction were committed

List the operations performed by the system when CP is to be taken. What does recovery system do if there is crash.

→ How to use CP
- → Write begin_checkPoint record into log.
- → Collect checkpoint data in the stable storage
- → Write end_checkpoint record into log.

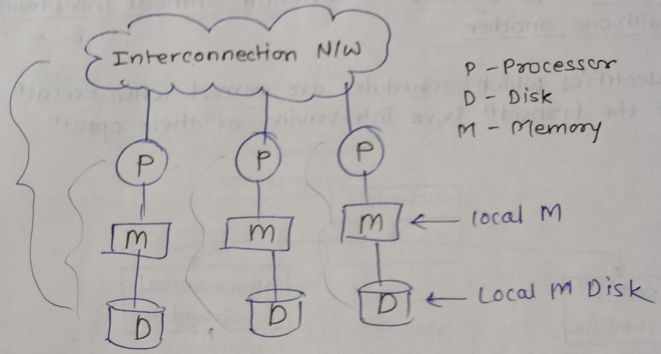



The behavior when the sys crashes & recovers when concurrent transaction are executed

- Recovery sys R the logs ← (Back) from the end to the Task CP from T4 to T1
- It will keep track of two lists Undo and Redo
- Whenever there is a log with instruction <Tn, start> & <Tn, commit> or only <Tn, commit> then it will put that transactn in Redo list
  T2 & T3 contain <Tn, start> and <Tn, Commit> whereas T1 will have only <Tn, Commit>. Here, T1, T2, T3 are in the redo list.
- Whenever a log record with no instructn of commit or abort is found, instruction that transactn is put to Undo list, Here, T4 has <Tn, start> but no <Tn, commit> as it is ongoing transaction. T4 will be put in the undo list

Serializable schedule

Shared nothing




- Each P has its own LM & LD
- A P at 1 node may ~~}~~ with another P using ↑ speed ~~}~~ n/w
- Any terminal can act as a Node which functions as Server for data that is stored on local disk
- Moreover, the interconnection n/w for shared nothing systems are usually designed to be scalable, so that we can ↑ transmission capacity as more nodes are added to the n/w

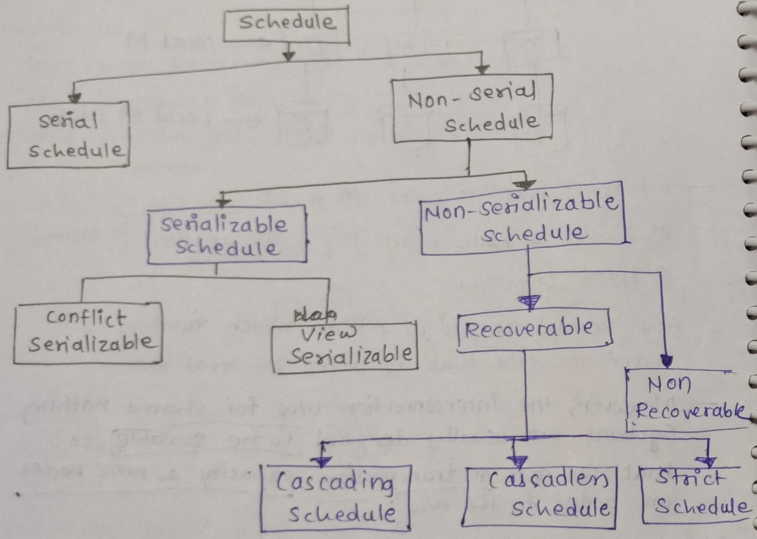
Advantages :-
- single IC n/w
- We can achieve ↑ degree of parallelism
- shared nothing architecture system are more scalable
- Easily support a large number of procesors

Disadvantages :
- cost of ~~}~~ and of non lo
  ⓐ level is ↑ than other 2 Architecture.

What is Serializable schedule ? Explain types with ex.

→ Is used to find non-serial schedule that allow the transactn to execute concurrently without interfering with one another.

– Identifies which schedules are correct when execution of the transctn have interleaving of their operatn




– Serializable schedule helps in improving both resource utilization & CPU throughput

– If it can be transformed into a serial schedule by swapping non-conflicting operatn

If it is view equal to a serial schedule (no overlapping transaction)

1) They belong to different transactn
2) ~~u~~ operate on same data item
3) At least one of them is a write opn

Draw & explain all parallel database architecture.

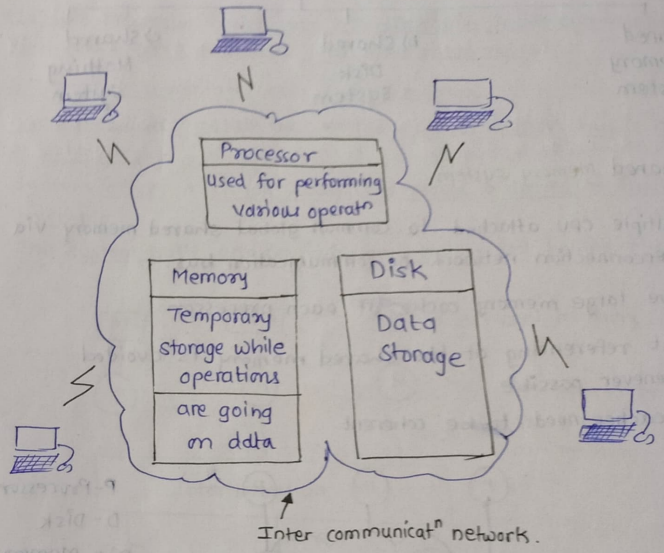



Fig: Parallel database system.

– A II DBMS is a DBMS that runs across multiple processor / CPUs
– Is mainly designed to execute query operatn in II whereever possi
– The II DBMS link a no. of smaller machine to achieve the same throughput as expected from a single large machine.
– In II DB machine envirnoment instead of 1 backend machine, there are multiple backend machines connected either to a single host or to multiple host.

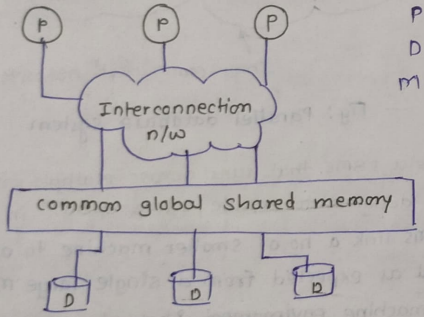
– A II DB sys seeks to
  → improve performance
  → Increase availability
  → Increse reliability

Types of Parallel Database

a) Shared Memory System
b) Shared Disk system
c) Shared Nothing System

## a) Shared Memory system

- Multiple CPU attached to common global shared memory via interconnection network or communication bus.
- have large memory cache at each processor
- that referencing of the shared memory is avoided whenever possible.
- chr caches needs to be coherent.



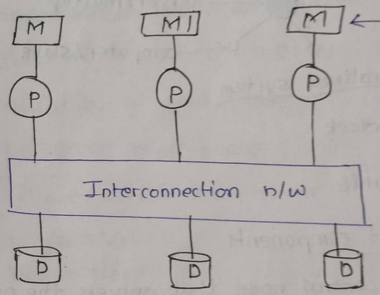

P-Processor
D-Disk
M = Memory

dvantages:
Effective Communicat$^{n}$ in processor
Data → access → any processor
without being moved from 1 lace to other
A P send message to other P uch faster using memory rites.

Disadvantages:
- Bandwidth problem
- Not scalabe beyond 32 or 64 P
- Interconnect$^{n}$ n/w into bottelneck
- More no.of P can ↑ waiting time of Processors.

## b) shared Disk system

→ Multiple Processor can access all disk via inter communication n/w. But, every processor has local memory.
→ S.D has 2 advantages over shared memory
→ Each processor has its own memory; the memory bus is not a bottleneck.
→ System offers a simple way to provide a degree of fault tolerance
→ The system built around this architecture are called cluster




P- Processor
D- Disk
M - memory.

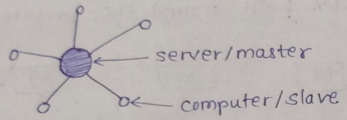
Advantages :-
→ High degree of fault tolerance is achived.
→ CPU has its own local M, so M.bus will not face bottelneck
→ fault tolerance

Disadvantag :-
→ Required rigid data Partitionning.
→ cost of communicat$^{n}$ & of non local disk access is higher than other 2 Architecture.

Explain in brief centralized & client server architecture.

→ Centralized sys. are sys. that use c/s architecture where one or more client nodes are directly connected to a central server.

→ Most commonly used type of sys. in many organizn where client sends a request to a company server & receives the responce.




haracteristics of centralized system

] Presence of a global clock

] One single central unit

] Dependent failure of components.

c/s architecture. The central node that servers the other nodes n the system is the server node & all the other node are he client nodes.

imitation of centralized system -

an't scale up vertically after a certain limit

fter a limit, even if you increase the h/w & s/w capabilities f the server node, the performance will not increase appreciably eading to a cost / benefit ration<1

Bottlenecks can appear when the traffic spikes

* Advantages client/server Architecture

1] Easy to physically secure.
2] Smooth & elegant personal experience
3] Dedicated resource
4] More cost-efficient for small sys.
5] Quick updates are possible.
6] Easy detachment of a node from sys.

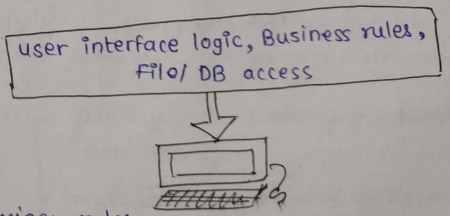
* Disadvantages client/server Architecture

1) Highly dependent on the n/w connectivity
2] No graceful degradation of the sys.
3] Less possibility of data backup.
4) Difficult server maintenace

Types of client-server Architecture.

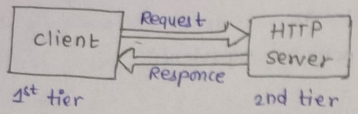
a] single Tier
b] Two - Tier
c] Three - Tier
d] N-Tier

1] Single Tier




- Presentatn bussines rules data access layers in single computing layer.
- Data is stored at single location & hence applications created on single tier architecture are relatively easy to manage & implement data consistency.

) Two Tier



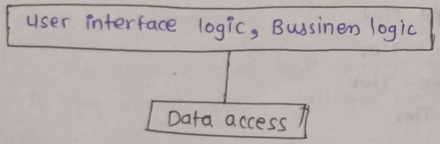

- seperates data & business logic.

It is generally data driven;
with Appln existing entirely on local machine
DB is deployed at specific & secure locatn

Characterized by a web client that displays info content & web server that transfer info to client

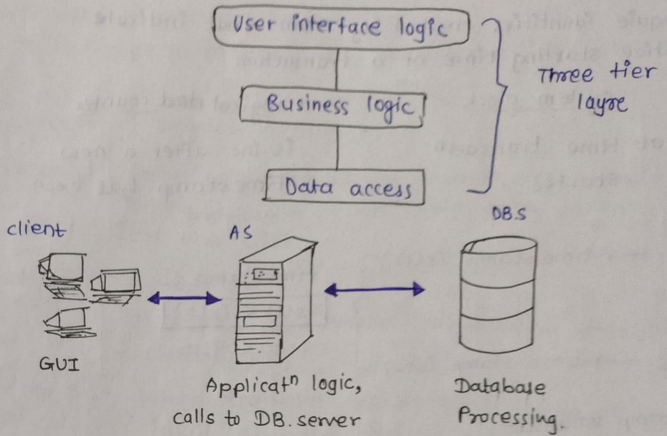
limitations.

1) When no of user ↑, performance begins to deteriorate
2) limited flexibility in moving program functionality from 1 Server to another without manually regenerating procedural code.




Two tier layer.

Three Tier Architecture.




Fig; Three Tier architecture.

1) Browser or GUI
2) Web server or Applicatn
3) Database Server [RDBMS]

- client tier - Contains all things that are visible to the user, such as screen layout & navigatn
- middel tier [ applicatn logic, process managment hides complexity of process Distribution.
- Data storage tier [ This layer in made up of DBMS & data storage.
  manages the storage & retrieval of data as well as security, data integrity of application.

Advantages : [ low cost deployment & maintenace.
offers high degree of flexibility in deployment of sys.

## ★ Time stamping method

▲ A unqie identifies created by DBMS that indicate relative starting time of a transaction.

system clock — at time transactn started

logical clock counter — is inc after a new timestamp has been assigned.

9:00 $T_1$ ⟶ time stamp Ts(1)
⋮
9:10 $T_2$ ⟶ time stamp Ts(2)

timestamp of
Ts(1) < Ts(2)
9:00 < 9:10

Time stamp values :-

R- Read timestamp(A) ⟹ largest time stamp of any transtn that execute Read successfully.

W- Write timestamp(A) ⟹ largest time stamp of any transtn which that execute write succes

1. Read time-stamp

| 10 | 20 | 30 | 40 |
|---|---|---|---|
| T1 | T2 | T3 | T4 |
| $R_1(A)$ | | | |
| $W_1(A)$ | | | |
| | | $R_3(A)$ | |
| | $R_2(A)$ | | |
| | | | $W_r(A)$ |

RTS(A) = 0
Update $T_1$ = 10 — 0 < 10
Update $T_2$ = 30 — 10 < 30
Update T3 = 20 — 20 < 30 ✗

∴ RTS(A) = 30

Write time-stamp.

WTS(A) = 0
Update $T_1$ = 10 — 0 < 10 ✓
update T4 = 40 — 10 < 40 ✓

WTS(A) = 40

- TS protocol ensures freedom from deadlock as no transactn even wait
- But the schedule may not be even be recoverable.

## ★ Shadow paging

- Used to recover from failure.
- It helps to maintain DB consistency in case of failure

Concept of shadow paging :-

step 1 :- Page is a segment of memory. Page table is an index of pages. Each table entry pt to a page on the disk.

step 2 :- 2 page ▤ are used during the life of a transatn
1) current page table 2) shadow page table.
- shadow page is copy of the current page table

step 3 :- When transactn start both table looks like identical, the current table is updated for each write operation

step 4 : The shadow page is never changed during the life of the transactn

step 5 : When the current transaction is committed, the shadow page entry becomes a copy of the current page table entry and the disk block with the old data is released.

step 6 : The shadow page table is stored in non-volatile memory. If the system crash occurs, then the show shadow page table is copied to the current page table.

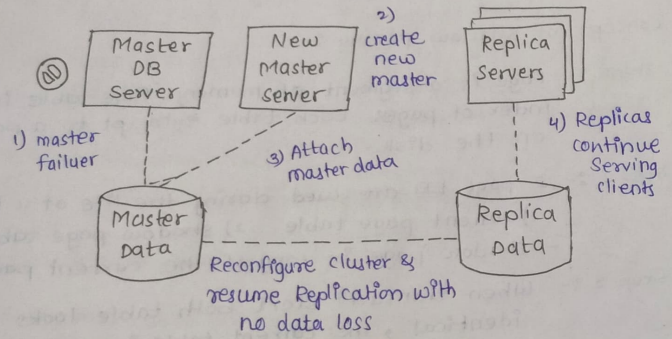
Advantage :-

1] No need for log records
2] No undo/Redo algorithm
3] Recovery is faster.

Disadvantage :-

1] Data is fragmented or scatter
2] Garbage collectn problem.
3] Concurrent transaction are difficult to execute.

* Cloud database




A cloud database is a database that is built, deployed, and accessed in a cloud environment, such as private public & hybrid cloud.

- As cloud services mature and reduce in cost
- Moving your database to the cloud can offer flexible, affordable and scalable DB mgmt.

Advantage
- Scalability
- Reduce Administratn Burden
- Improved Security

Downtime
Average cost

* Disadvantages
  - → Risk of data confidentiality
  - → Depends on Internet connecth
  - → Lack of total control
  - → Difficult to migrate.
  - → Fixed contracts can be a problem

8- Mobile Database

→ A database that can be connected to a mobile computing device over a mobile n/w (or wireless n/w)
→ A database which is actually stored by the mobile device
→ Seperate from main DB
→ Can easily be transported to various places. Even they not connected to the main DB.
→ They can still communicate with the DB to share & exchang data.
→ The main sys DB that stores all data & is linked to the mobile DB
→ Allow user to view info even while on the move. It shares info with main DB
→ mp or lappi uses M.DB
→ A communicatn link allows the transfer of data between M.DB & the main DB.

* Advantages of Mobile DB
  - → It provieds wireless DB access
  - → DB sys. are synchronized using M.DB & multiple use can access the data with seamless delivery process
  - → Requires very little support & maintainance
  - → Syncronised with multiple devices such as mobiles, computer devices, laptop etc.

* Disadvantages
  - less secure
  - Security hazard
  - limited battery tho loss power frequently.

Explain Transformation of rational Expression.

→ Relational Algebra is procedural query language, which takes Relation as input and generate relation as o/p

— 2 relational algebra expression are said to be equivalent if the two expression generate the same set of tuple is irrelevant.

— The order of tuples is irrelevant

— An equivalence rule says that expression of two forms are equivalent replace expression of 1st from by second or vice versa.

* NO SQL

— term NOSQL was first used by carlo strozzi in 1998

— He mentioned this name for his Open Source Database System in which there was no provision of SQL query interface.

— In early 2009, at conference held in USA, NOSQL was comes into picture & actually comes in practices.

— It is not a RDBMS ✓

— Specially designed for large amount of data stored in distributed envimoment

— It is not bounded by table schema restrictions like RDBMS

— It gives options to store some data even if there is no such column is present in table.

— Avoid join operation.

* Need

— In real time, data requirements are changed a lot. Data is easily available with FB, Google, Twitter & other.

— The data that includes user info, social graphs, geographi locatn data & other user-generated content.

— To make use of for specially designed for operating huge amount of data.

* Advantages:

1) Good resource scalability ✓
2) lower operational cost ✓
3) No static schema
4) Faster data Processing ✓
5) No complecated relationships ✓

* Disadvantages

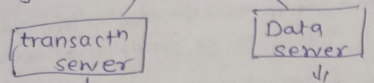
→ Not a defined standard
→ Limited query capabilities.

* Companies working with NOSQL

→ Google
→ Facebook
→ LinkedIn
→ McGraw-Hill Educati

* Server system Architecture.

- Server system can be categorized.

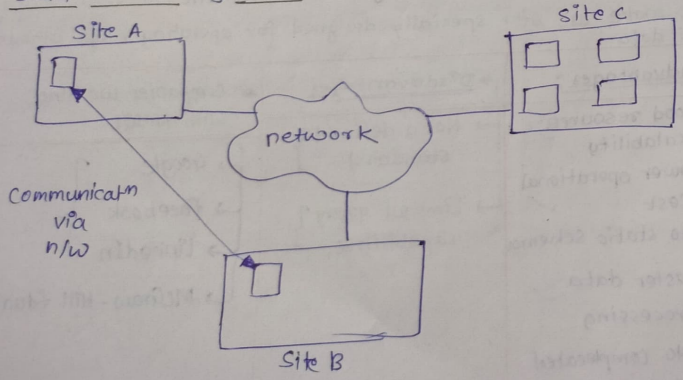


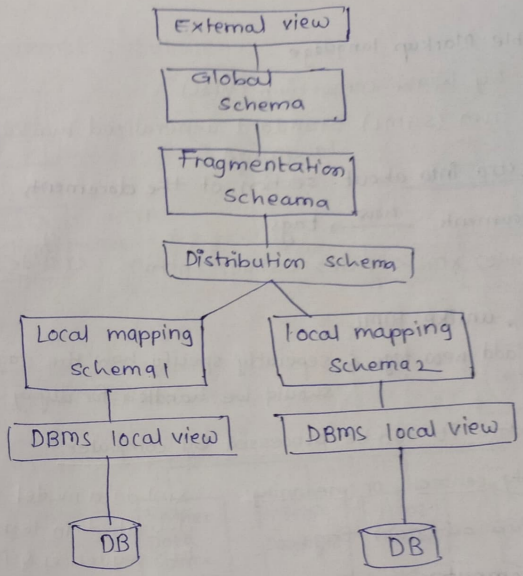

used in RBMs

used in OOP DBSys.

* Distributed system.

- loosely coupled sites that share no physical component
- DB sys. that run on each site are independent of each other
- ⇄ may access data at 1 or more sites.
- Data spread over multiple machines.
- N/W interconnects the machines
- Data shared by user on multiple machine.




Distributed DB

Homogeneous
↓
Same s/w / schema on all sites

Heterogeneous

Diff. s/w / schema on Diff. sites.

# XML

- Extensible Markup language.
- Defined by WWW Consortium (W3C)
- Derived from (SGML) Standard Generalized markup language.
- giving extra info about section of the documents

Documents → have → tags

eg: <title> XML </title> <slide> Intro ... </slide>

- Extensible, unlike HTML
  - user can add new tags & seperatly specify how the tag should be handled for display
- Mark-up data it can be processed by computer.
- Describe only content, or 'meanning'
- We make up our own tags.

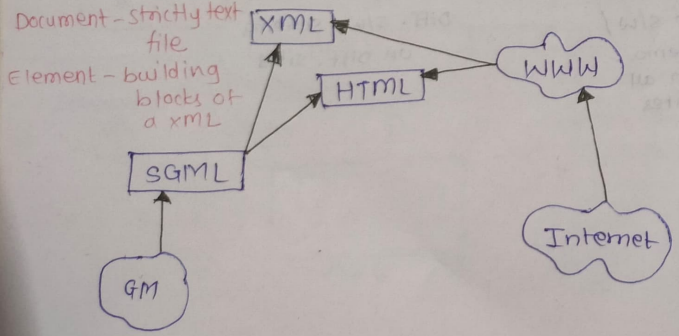
XML data model is described in terms of sequences & items, atomic value & nodes

XML is for computer.

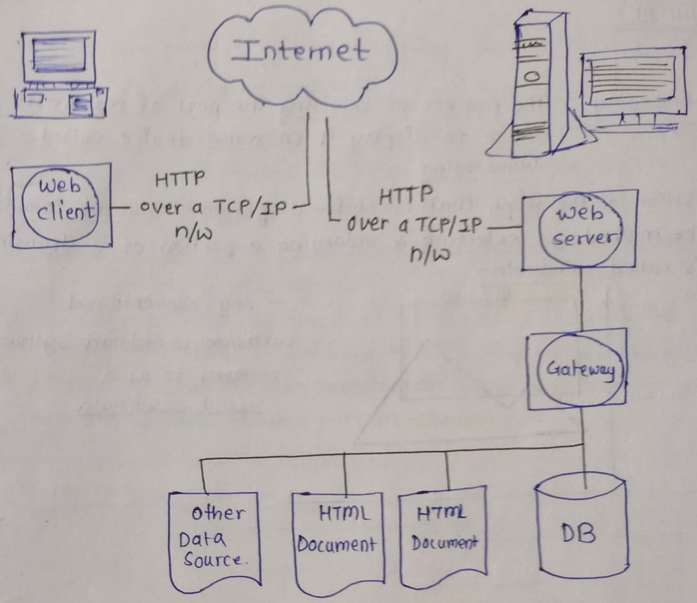
- Describes Data
- Rules ar strict & errors are X allowed.
- Most of the can display XML

Document - strictly text file

Element - building blocks of a XML




## ★ Internet Database




Internet Services.

1) Web server
2) DB - enabled services.
3) FTP
4) Firewalls & proxy servers
5) Doc search.
6) load balancing caching.

- WWW
  - Browser
  - Web server
  - Web pages

* Communicatn tech

1) IP Address
2) HTTP — web s to browser
3) URL — Uniform Resourse Locater.

Internet Related language.

- HTML
- SGML
- XHTML
- JAVA (applet)
- Java script — CSS