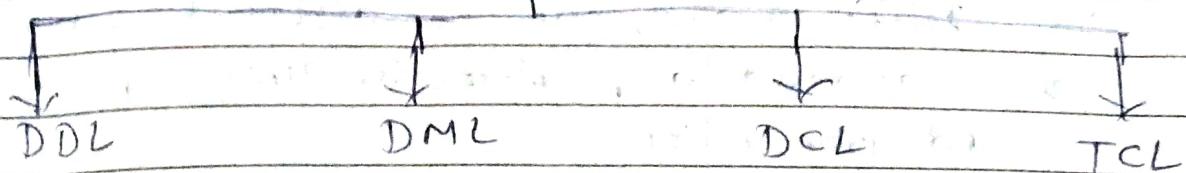


SQL



→ Create	→ Insert	→ Grant	→ Commit
→ Drop	→ Update	→ Revoke	→ Roll back
→ Alter	→ Delete	GR	→ save point
→ Truncate	→ Select		CRS
→ Rename	PDU		
KARDT			

Fragmentation and Replication

a) Distributed Database :-

- Not limited to 1 system.
- Spread over different sides.
- Required when a particular database needs to be accessed by users globally.
- Needs to be managed such that it should look like a single database.

• Types -

• Homogenous -

- All diff. sites store database identically. The O.S., data struct. and DBMS used are all the same at all sides.
- Easy to manage.

D	D	M	M	Y	Y	Y
---	---	---	---	---	---	---

• Heterogeneous Database :-

- Different sites can use different schemas.
- Can lead to problems in query processing and transactions.
- One site can be unaware of other sites.
- Different computers may use diff. O.S, Data Structure and DBMS.

b) Distributed Data Storage

- 2 ways in which data can be stored on different sites.

i) Replication —

- In replication, systems maintains copies of data.
- the entire relationship is stored redundantly at 2 or more sites.

If entire D.B is present at all sites it is fully redundant D.B.

ii) Fragmentation —

- Relations are fragmented (Divided into smaller parts). and each fragment is stored into different sites where they're required.

- Fragments are such that they can be used to reconstruct the original section/rel.

Fragmentation can be done in two ways —

D) Horizontal fragmentation —

- Splitting by rows
- Groups of tuples
- Each tuple is assigned to at least one fragment.

E.g -

CREATE TABLE customer-delhi AS

SELECT * FROM CUSTOMER where city = 'Delhi';
SELECT * FROM customer-delhi;

2) Vertical fragmentation —

- Splitting by columns.
- The schema of rel is divided into smaller schemas.
 - Each fragment must contain a common candidate key so as to ensure lossless join.

E.g -

CREATE TABLE account-personal AS

SELECT acnumbert, custid FROM account;
SELECT * FROM account-personal;

2. Types of Replication Schemas :-

1) Full Replication (for branch table)

- Replicate entire relation / table on two different servers to ensure data availability.

Eg —

On Server 1 :-

```
CREATE TABLE branch-copy AS  
SELECT * FROM branch;  
SELECT * FROM branch-copy;
```

On Server 2:-

```
CREATE TABLE branch-copy2 AS  
SELECT * FROM branch;  
SELECT * FROM branch-copy2;
```

2) PARTIAL Replication :-

- Replicate the particular part / schema of a relation fragment into another server to improve the availability of customers.

On Main Server —

```
CREATE TABLE customers-delhi AS  
SELECT * FROM customer WHERE city = 'Delhi';  
SELECT * FROM customers-delhi;
```

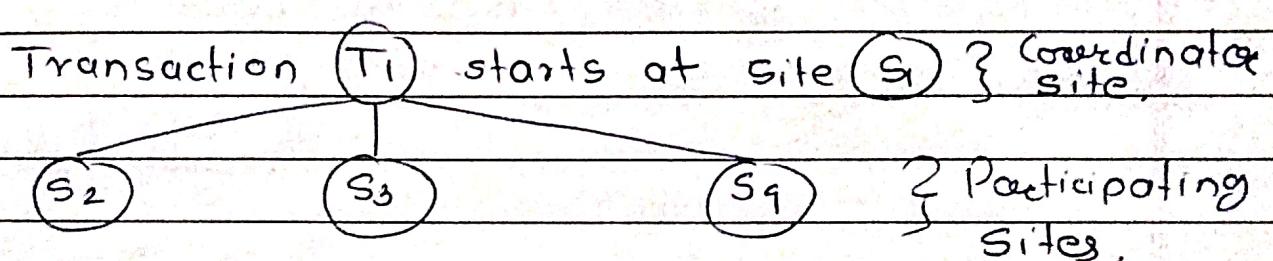
On Replica Server

```
CREATE TABLE customers-delhi-copy AS  
SELECT * FROM customer WHERE city = 'Delhi';  
SELECT * FROM customers-delhi-copy;
```

Two Phase Commit Protocol :-

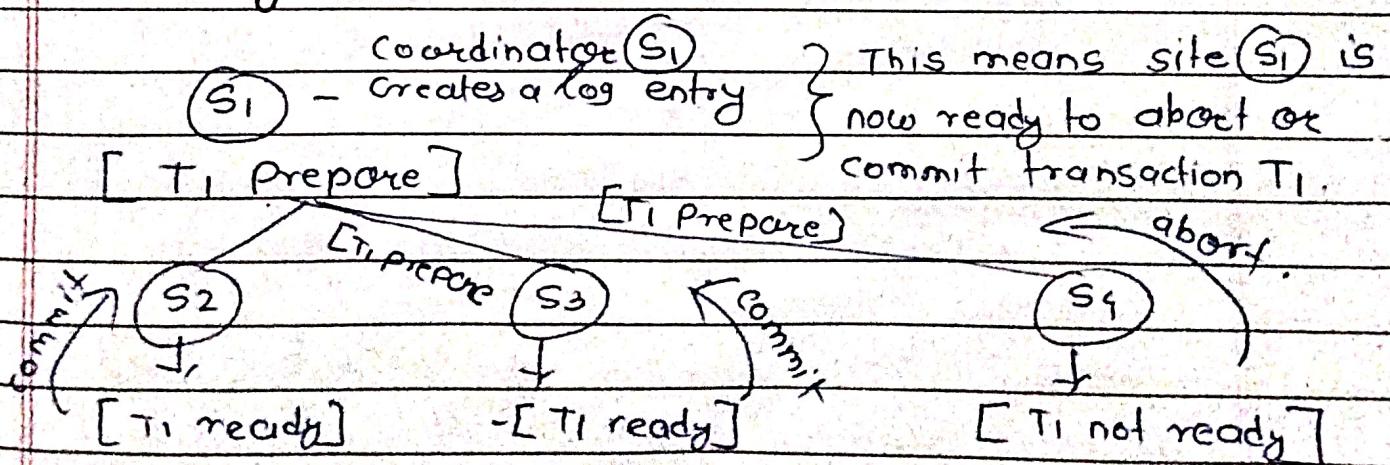
- Recovery System in distributed database.
- There are two different phases —
 - a) Voting. : Participating sites vote \rightarrow commit. \rightarrow Abort.
 - b) Decision. :- Coordinator site decides whether the transaction has to be committed or needs to be aborted.

E.g -



Based on the votes of S_2 , S_3 and S_4 the S_1 site will decide whether to commit the transaction T_1 or not commit it.

i) Voting:-



- So the participating sites will create a response which will be sent to coordinator that whether they are ready to commit or abort the transaction,

ii) Decision:-

[i) [Ready, T] message
from all Participating Sites.

Then Result :- COMMIT

[ii) If atleast one [not-ready T]

Then Result :- ABORT

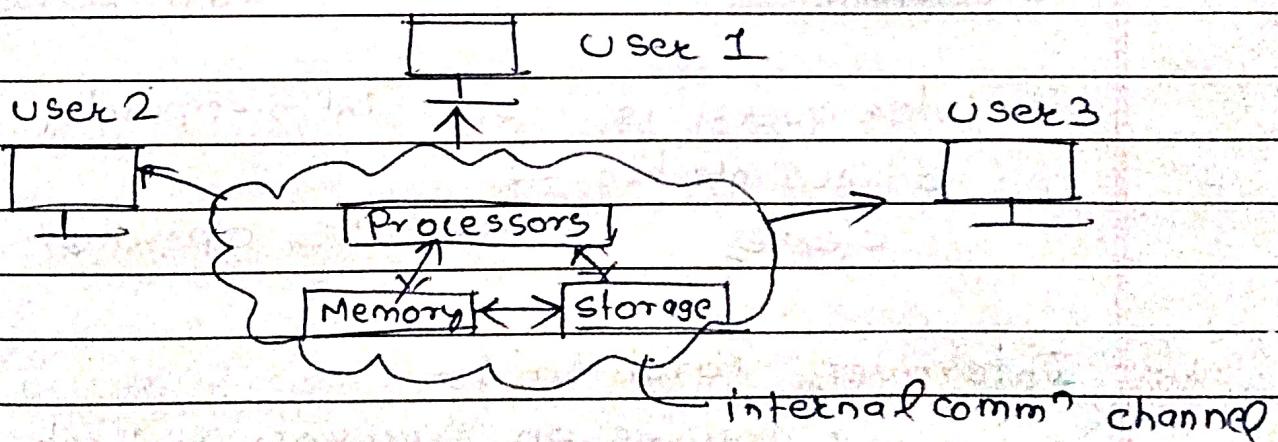
(Hence it aborts the current transaction.)

Parallel Databases.

- Sometimes the client server and centralized system is not efficient to handle huge amount of data with high data transfer rate.
- The need to improve the efficiency gave birth to the concept of parallel databases.
- Parallel db sys. improves performance of data processing using multiple resources in parallel like multiple CPU and disks are used parallelly
- It also performs many parallelization operations like data loading and query processing.

Advantage —

- Provide distributed access of data.



Query Processing :-

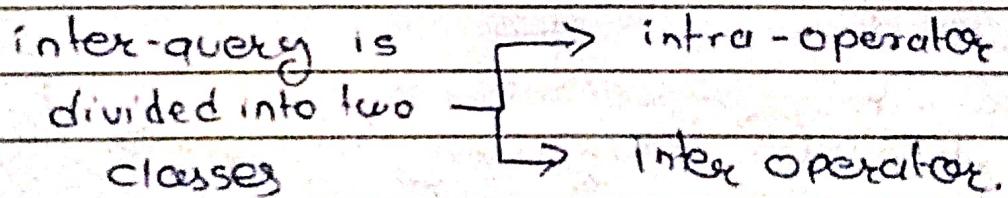
- Query processing is the process through which database Management System parses, verifies and optimizes a given query before creating low-level code that the DB understands.

- Query Processing in DBMS where code is first generated and then executed to perform various operations has two phases: compile time and runtime.

Types of Parallelism in Query Execution in DBMS :-

a) Intra query Parallelism:-

- Breaking down a single query into smaller tasks.
- Optimizes Overall process by utilizing resources effectively.
- Boosts the speed of query execution by distributing processing power across different



b) Interquery Parallelism:-

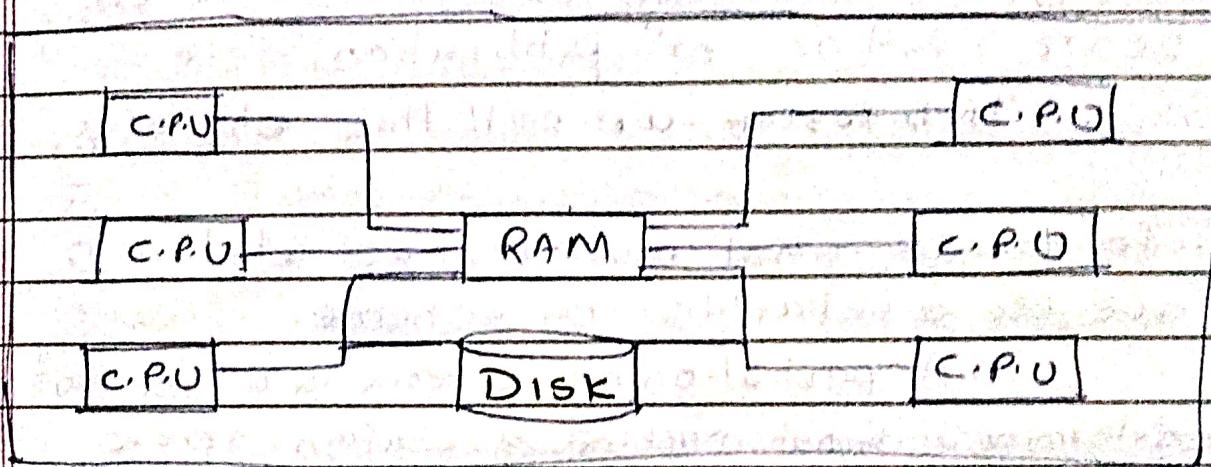
- Focuses on executing multiple queries simultaneously.
- This feature allows multiple queries to be executed simultaneously instead of having to wait for one query to complete before initiating the next one.



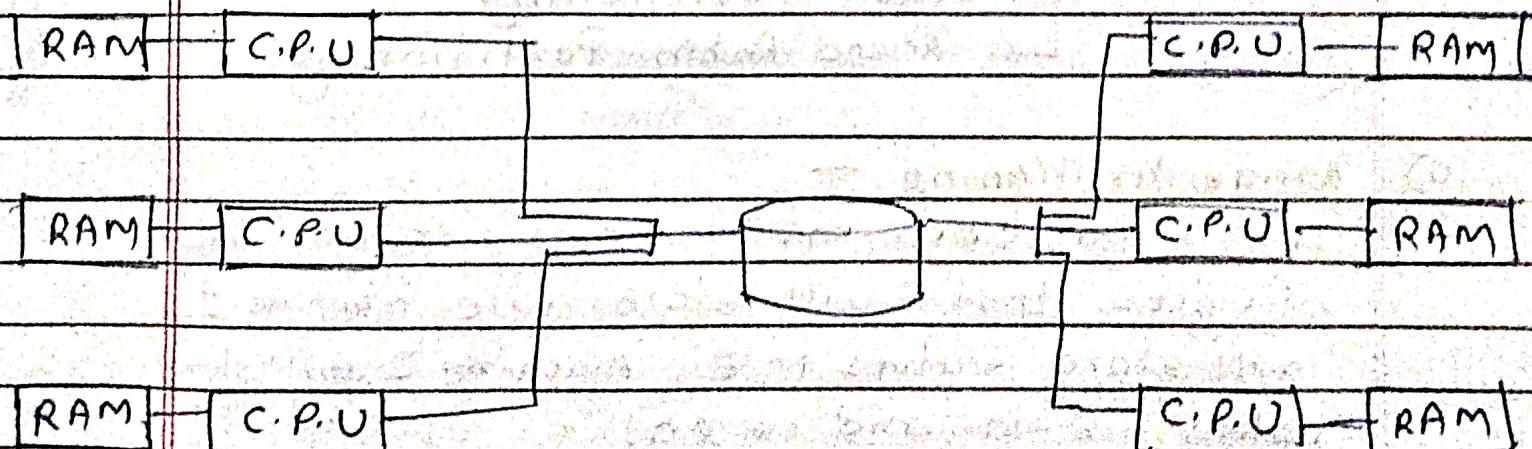
• main goal is to increase overall system throughput and reduce query response time.

Parallel Architectures :-

a) Shared Memory :



b) Shared Disk :



• Partitioning :-

- Imagine trying to find a specific book in a library with millions of books and no organization system.
- Partitioning is like organizing the library into different sections based on genre, author or publication date.
- In industry we call this sharding.

⇒ To decide what machines get what data we use a partitioning scheme.

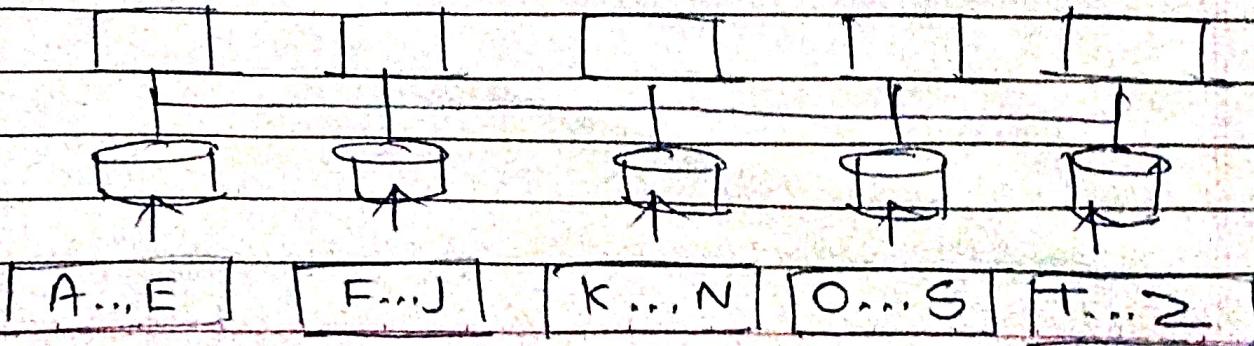
- A partitioning scheme is a rule that determines what machine a certain record will end up on.

• Types -

- Range Partitioning
- Hash Partitioning
- Round Robin Partitioning.

a) Range Partitioning -

- Each machine gets a certain range of values that it will store. (i.e. machine 1 will store values 1-5, machine 2 will store values 6-10 and so on).



b) Hash Partitioning :-

- In hash partitioning scheme, each record is hashed and is sent to a machine matches that hash value.
- This means that all like values will be assigned to the same machine (i.e if value 4 goes to machine 1 then all of the 4s must go to that machine).

c) Round Robin Partitioning :-

- In this scheme we go record by record and assign each record to the next machine.
- for example, the first record will be assigned to the first machine, the second record will be assigned to the second machine and so on.
- This scheme will actually achieve maximum parallelization.

Parallel Sorting :-

- Parallel sorting involves dividing the dataset into smaller chunks and distributing them across multiple processors or cores for simultaneous processing.
- Each processor independently sorts its assigned portion of the data, and then the sorted sublists are merged together to obtain the final sorted result.

There are two steps for parallel sorting—

- 1) Range partition the table.
- 2) Perform local sort on each machine.

Parallel Join Methods:-

1) Parallel Sort-Merge method—

- It first performs a parallel sort to order the data, and then merges the sorted tables in parallel.
 - During the merge, the facility concurrently joins multiple rows from one table with the corresponding rows in the other table.

- You can use the parallel sort-merge join method to execute any join that meets the requirements for a parallel join.

2) Parallel range join method :-

• It uses a join index to determine the ranges of rows between the tables that can be joined in parallel.

The parallel range join method requires you to create a join index on the columns to be joined in the tables that you want to merge.

The join index divides the two tables into a specified number of near-equal parts, or ranges, based on matching values between the join columns.

3) Parallel joins with Group By :-

A powerful feature of the server parallel join facility is its integration with the server parallel Group-By facility.

If the result of the parallel join contains a GROUP BY statement, the partial results of the parallel joins threads are passed to the parallel Group-By facility, which performs the group-by operation in parallel.

Oracle utilities :-

- 1) Data Pump:- for fast data export / import
- 2) SQL*loader:- for bulk loading of data from external files.
- 3) RMAN : for backup and recovery operations
- 4) ASM: for efficient storage management
- 5) Oracle Enterprise Manager — for monitoring and managing the database
- 6) flashback technology - for recovering from data loss or corruption,