Day → 6

45.

Subquery :→

→ Subquery is a query within a query.

→ A subquery must be enclosed in brackets and can be used in SELECT, from, where & Having clause.

Independent Subquery :→

→ In an I.S., the inner & outer query are independent of each other. You can run an inner query & inspect it's result independent of the outer query.

** Imp. Points →:

→ Subquery in SELECT & from clause are rarely used.

→ Subqueries in WHERE and HAVING clause are classified into independent & correlated subquires.

Correlated Subquery :→

→ A C.S. is one in which the inner query that depends upon the outer query for it's execution

→ Specially it used a column from one of tables in the outer query.

→ The inner query is executed iteratively for each selected row of the outer query.

→ In case of independent subquery, the inner query just executes once.

Centralized vs. distributed:

Centralized SQL database →	Distributed NoSQL database
→ Store & process data centrally.	→ Data is stored across many machines.
→ If the central db server goes down, no users request can be processed	→ If any DB server goes down, user request can be processed from remaining servers
→ Response time are slower	→ Since user requests are distributed across many servers, response time is much faster than centralized users.
→ To meet the growing load, you can upgrade the server (CPU, RAM, hard disk, etc).	→ To meet growing load, more servers can be rapidly added. This is horizontal scalability
→ This is called vertical scalability	→ Scalability is unlimited
→ Scalability is limited by hardware constraint.	

*** Remaining

In Day → 7 & Day → 8

025 Normalization topics

Transactions:→

↳ *** Remaining

Day → 9 →

Big Data:→

→ variety (reviews, comments, ratings & votes, likes, photos, videos etc).

→ Volume (typically 100's of Terabytes of user contributed data).

→ Velocity (speed at which data arrives & changes).

Ex
→ A large no. of people commenting & rating on tourist locations

value	
Byte	→ 1
KB	→ 1000
MB	→ 1000^2
GB	→ 1000^3
TB	→ 1000^4
PB	→ 1000^5
EB	→ 1000^6
ZB	→ 1000^7
YB	→ 1000^8.

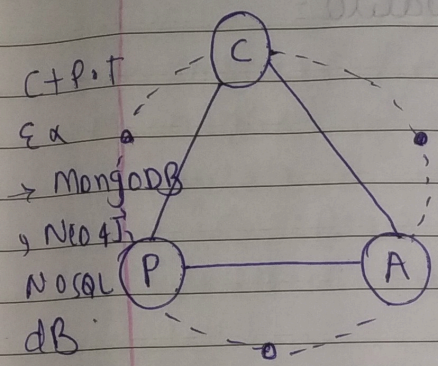
NOSQL databases:→

→ Big data is typically stored in NOSQL (Not Only SQL) databases.

→ Major differences b/w SQL & NOSQL databases:→

SQL (Relational database)	NOSQL databases
→ Relational representation (stored in tables as rows & columns)	→ NOSQL represent as Non-relational representation
→ All rows have a fixed set of columns	→ Data doesn't have a fixed structure (can be spread among across images, audio, videos, comments, ratings, like etc)
→ Used for relatively smaller databases (100 GB)	→ Used for large databases (e.g. fb Page updates, live stock market

→ Some give importance to consistency & while other give importance to availability.



Consistency + Availability (Oracle) (SQL databases)

Availability + P.T. NoSQL database e.g. Cassandra

Eventual consistency

** remaining

Desired features of databases →:

→ All database should ideally provide the following features

→ Consistency →

An end user must be able to see the latest data at all times.

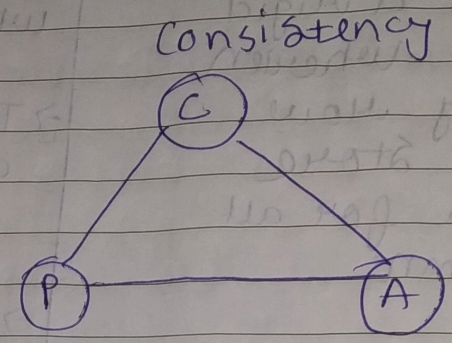
Availability →

Every database request must be responded by the server

Partition Tolerance →

When two systems can't talk to each other in a network, it's called network Partition. Our DB system should continue to function even if there is a N.P.

→ All three are abbreviated as CAP.



Eric Brewer's CAP Theorm:⇒

→ It states that it's impossible for a distributed system to guarantee all three (C, A & P.T.).

→ NOSQL databases are P. Tolerant. In case of N.P. there is a tradeoff b/w consistency & availability.

→ Some to co while impo a

C+P.T Ex → MongoDB, Neo4J, NOSQL DB (P)

Event

** rem

Rigid vs. flexible schemas:

→ Relational (SQL database) follows a rigid schema wherein every row must store values for all column.

→ This also results in insufficient storage.

→ NOSQL database allows a flexible schema wherein each rows stores only the required data. Null values are not stored

→ NOSQL allows multiple value to be stored in a single column.

[See ex for Infy].

Replication:

→ NOSQL servers usually run on low cost hardware which can fail anytime.

→ Therefore, the data is replicated & stored on multiple servers.

→ In case of any server fails, the data is still available in another server

R.F. = No. of copies = 3

Types of NoSQL:→

→ NoSQL database store data in four ways.

i) Key-value Store →
eg. (Riak, Redis)

ii) Column family Store
(e.g. HBase, Cassandra)

3. Document-oriented database →

eg. → MongodB

4. Graph database →:

eg. Neo4.

---

ch: Get (key) →
Returns the value associated with provided key

② Put (key, value) →
Associated value with key

✓ ✱ ✱
See characteristics at time of

i) Key-value Stores:→

→ Simplest form of NoSQL database.

→ It stores information as an attribute (key) & its value.

Ex →

key →:

"Qbt 001"

value →:

{itemR: [{product: "HP laptop", qty: 1},

{product: "Sandisk16GB", qty: 3}] }

operations on a key-value store →:

3. Multi-get (key1, key2 ... keyN)
→ Returns the list of values ass. with list of keys

4. Delete (key) →
Remove the entry for