Agenda.

→ CAP Theorem

→ SQL (Vs) NoSQL DBs

CAP

Consistency

⇒ Everytime we read, We get the data of latest write.

⇒ All the machines / replicas contains the same data.

Availability

⇒ If we send a query then system must be available to give you a response.

Partition Tolerance

↳ In distributed System, We can't avoid partition 100%.

⇒ System should be able to handle network partitions.

In distributed System, We can only achieve (2) out of (3).

⇒ In distributed system, Partition Tolerance is always going to be there so we can choose only one among C (vs) A.

distributed system → CP
distributed system → AP

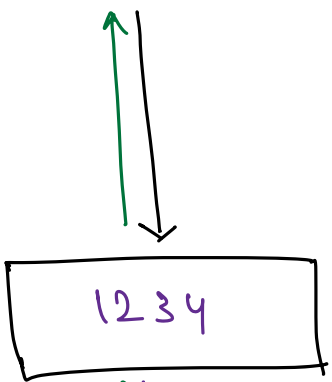
Single M/C ⇒ CA

Example

Vinay

↳ Reminder Service (X42)

1)

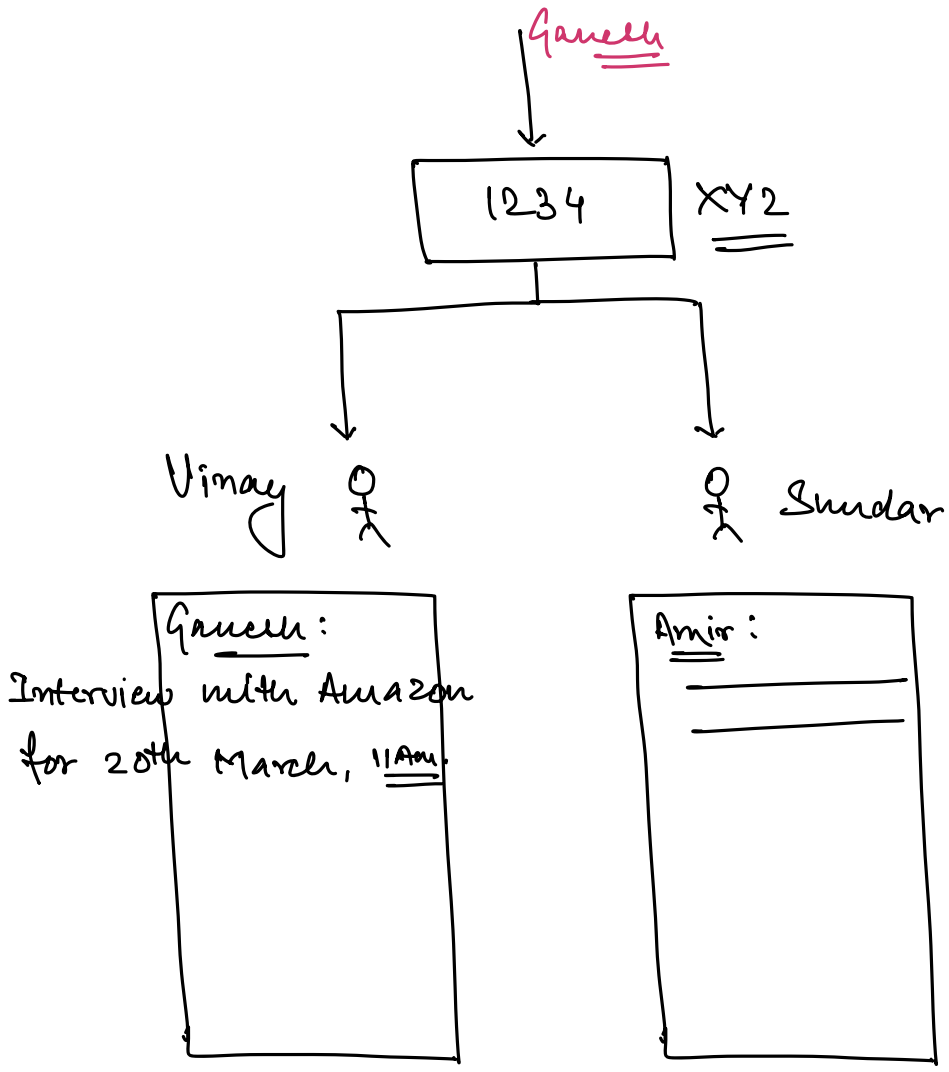


Vinay

Ganesh
Interview with Amazon on 20th Mar 11AM.

Amir
Date scheduled for 15th Mar, 8pm.
Date scheduled for 16th Mar, 8pm.

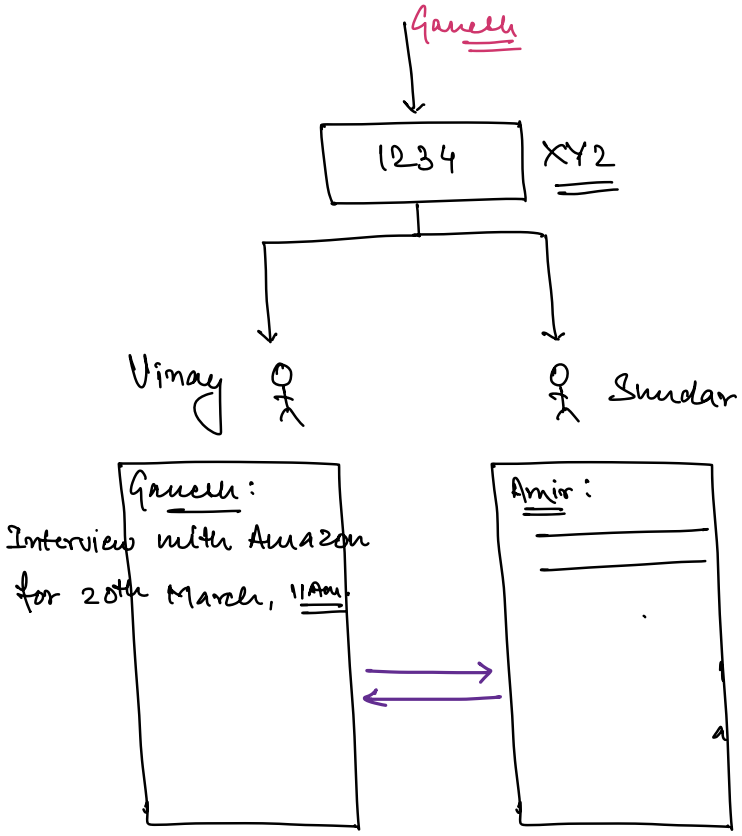
II)



PT ✓

Consistency Availability

III)



PT

Consistent.

CAP

↳ Immediate Consistency (VS) Availability.

⇒ Eventual Consistency gets achieved always once partition gets over, That means in distributed System the fight is always among Strong | Immediate consistency (VS) Availability.

LinkedIn | fb.

→ Makes post.

→ AP

↳ Eventually consistent.

Banking App"

↳ Strongly consistent.

↳ CP

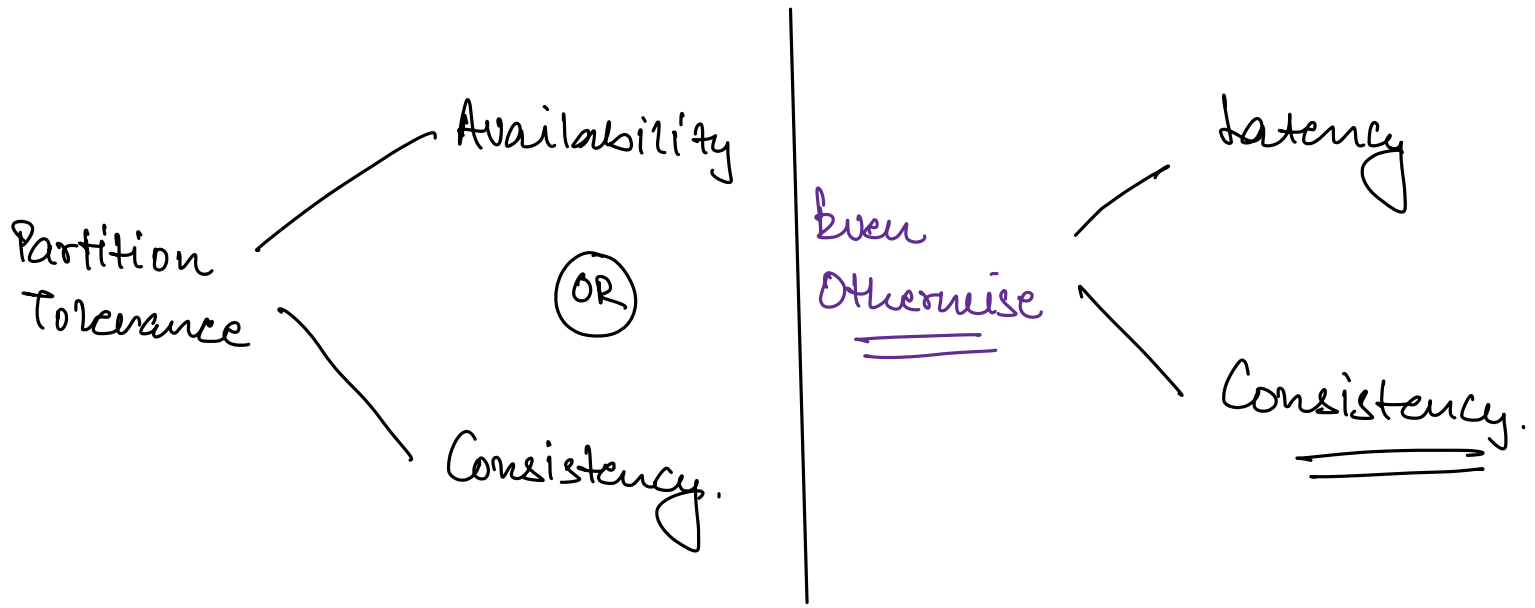
IRCTC.

→ Strongly consistent.

→ CP

BMS.

PACELC Theorem.



⇒ If we want our system to be highly / Strongly consistent then we'll have to take the trade off of latency that means for Strongly consistent systems, latency will be HIGH.

Databases.

↳ SQL (vs) NoSQL.

↓

→ Not a DB type

→ Query language.

→ Relational DB.

Strengths of SQL DBs.

→ All these strengths are applicable at low scale.
(qps | amount of Data)

→ Data gets stored in the form of tables.

Normalization.

↳ Prevents anomalies & reduces redundancy

products

id	title	...	category_id
1	iphone		101

Categories.

id	- - - - - - -
101	Mobile

→ Structured / fixed Schema.

↳ Tables, attributes & even the size of each attr is fixed.

↳ Changing the schema is difficult.

↳ let's say if we want to add / remove a column in the table then the entire table needs to rewritten.

↳ Data Migration

↳ Requires some down time.

↳ Extremely slow.

ACID.

Atomicity : Either the complete transaction should take place (or) nothing should take place.

: DB shouldn't be in the partial state at any point.

Consistency

ACID Consistency ≠ CAP Consistency.

↓

No stale reads.

ACID Consistency :-

ensures that the database remains in a valid state before and after a transaction.

⇒ DB Constraints are enforced properly.

↳ FK Constrains | NULL | Unique.

Isolation

↳ Multiple transactions running together Shouldn't Interfere with each other.

Durability.

↳ Changes should be persisted.

⇒ SQL DBs are very mature.

⇒ SQL DBs provides very powerful querying Capabilities.

Weakness of SQL DBs.

=> All the strengths becomes weaknesses at HIGH scale.

=> Normalization.

Student_Courses

Student	Courses.
Sundar	HLD, LLD
Ajay	DBMS
Ganesh	DBMS, HLD

→

Student_id	name
1	Sundar
2	Ajay
3	Ganesh

Course_id	name
1	DBMS
2	LLD
3	HLD

Student_Courses

Student_id	Course_id
1	2
1	3
2	1
3	1
3	2

=> Lot of tables.

=> Querying the DB becomes complex due HIGH no. of JOINS.

=> JOINS are slow.

↑

Normalized DB.

Fixed Schema.

⇒ Amazon's Product DB.

ACID.

⇒ NoSQL