SUNBEAM

Fundamentals of Data Engineering

Trainer: Pradnyaa S. Dindorkar

BigData
CCAT exam

@ 5 questions -> MCQ

✓ 1 day ->
✓ 2 day
✓ 3 day -> Quiz 1 ->10
4 day -> Quiz 2 -> 10
5 day -> End quiz ->20
@10
@20

per-requistes
1: appln dev / programming
2: Database
3: Networking
4: OS - Linux

PG-DBDA

Introduction

- **Big Data Fundamentals**
  - Evolution of Data Enggineering | V's: Volume, Velocity, Variety, Veracity, Value
- **Databases**
  - RDBMS - ACID, SQL (basic concept only) | NoSQL - BASE, CAP theorem
- **Data warehouse - OLAP vs OLTP**
  - Data cleansing, Data transformations and Data modelling | Data warehouse vs Data mart
- **Data Engineering Life Cycle**
  - Source → Ingestion → Storage → Transformation → Serving
  - Ingestion: ETL vs ELT
  - Storage: Distributed storage, Storage services | Processing: Batch vs Stream
- **Cloud computing fundamentals**
  - Virtualization, Scaling, Elasticity, Cloud service models, Vendors
- **Big Data Technologies**
  - Frameworks: Hadoop, Hive, Spark, Kafka
  - Applications and Job profiles.

Data Engineering at a Glance

before 1970 file i/o

Database & Warehouse

1970

RDBMS

CRUD

@1990

DWH

kb-mb

@1980

Internet & DotCom

@1991-1995

gb-tb

data burst@2000

@1998

NoSQL Database

CRUD

hugh data

4+2

3,4,5 core

massive parallel processing

MPP & Big Data Tech

@2003

google

Cloud Computing

rent

AWS

@2010

Big Data characteristics

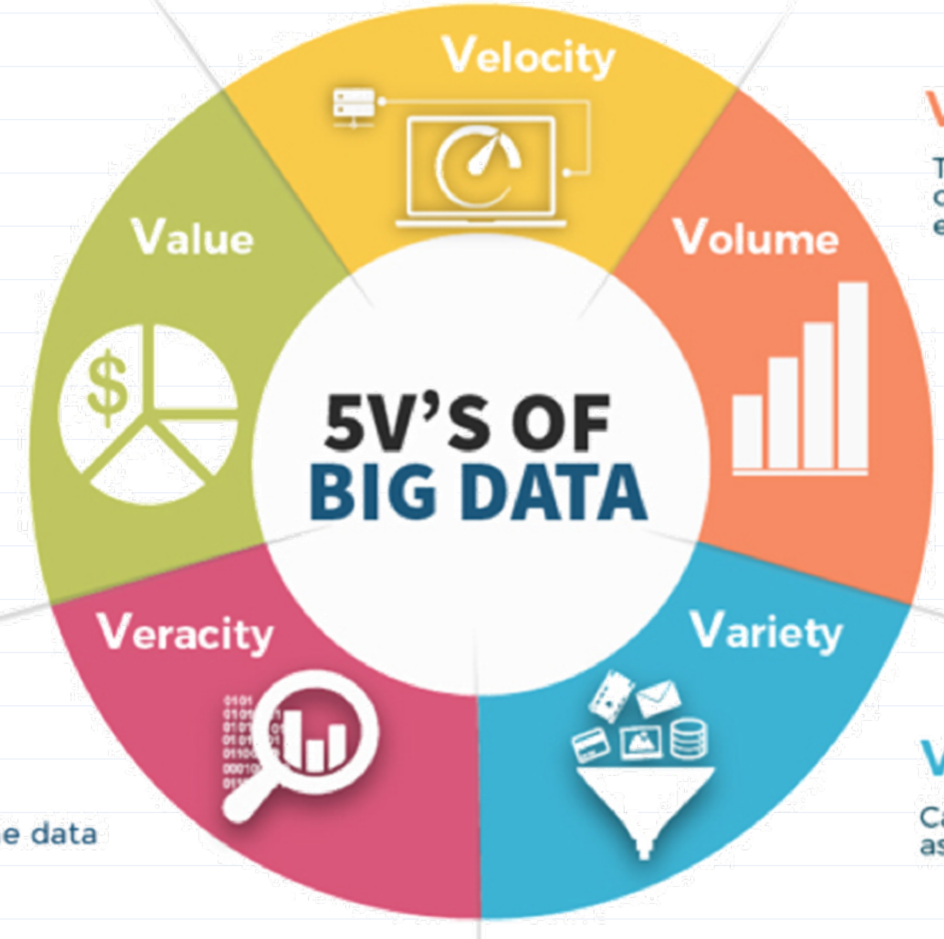
Velocity
Speed at which data is emanating and changes are occurring between the diverse data sets

Volume   100s TB and above

This refers to the sheer volume of data being generated every second.

Value

Having access to big data is all well and good but that's only useful if we can turn it into a value.



Veracity
Data reliability and trust.
Verifying and validating the data

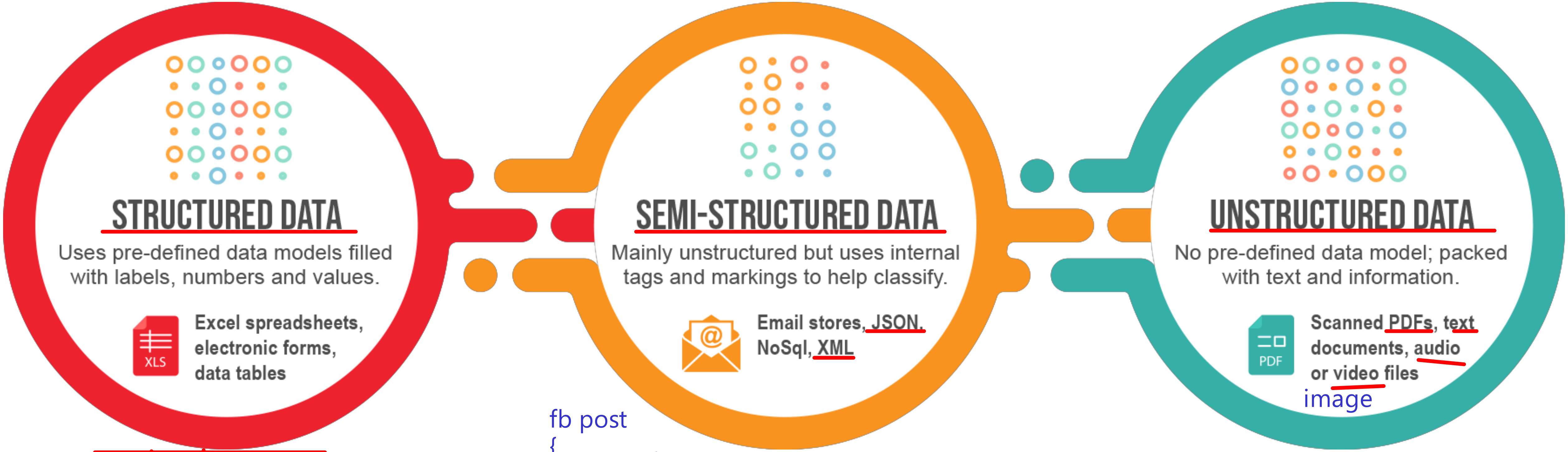
Variety
Can use structured as well as unstructured data.

Types of Data

fb => 2008

fixed schema

flexible schema



STRUCTURED DATA

Uses pre-defined data models filled with labels, numbers and values.

Excel spreadsheets, electronic forms, data tables

SEMI-STRUCTURED DATA

Mainly unstructured but uses internal tags and markings to help classify.

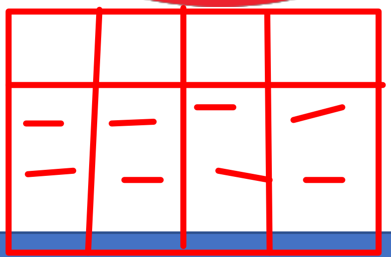
Email stores, JSON, NoSql, XML

UNSTRUCTURED DATA

No pre-defined data model; packed with text and information.

Scanned PDFs, text documents, audio or video files

image



fb post
{
post:
likes:
image: png
jpg
gif

video:
tag:
loc:
reaction:
feeling:

ML

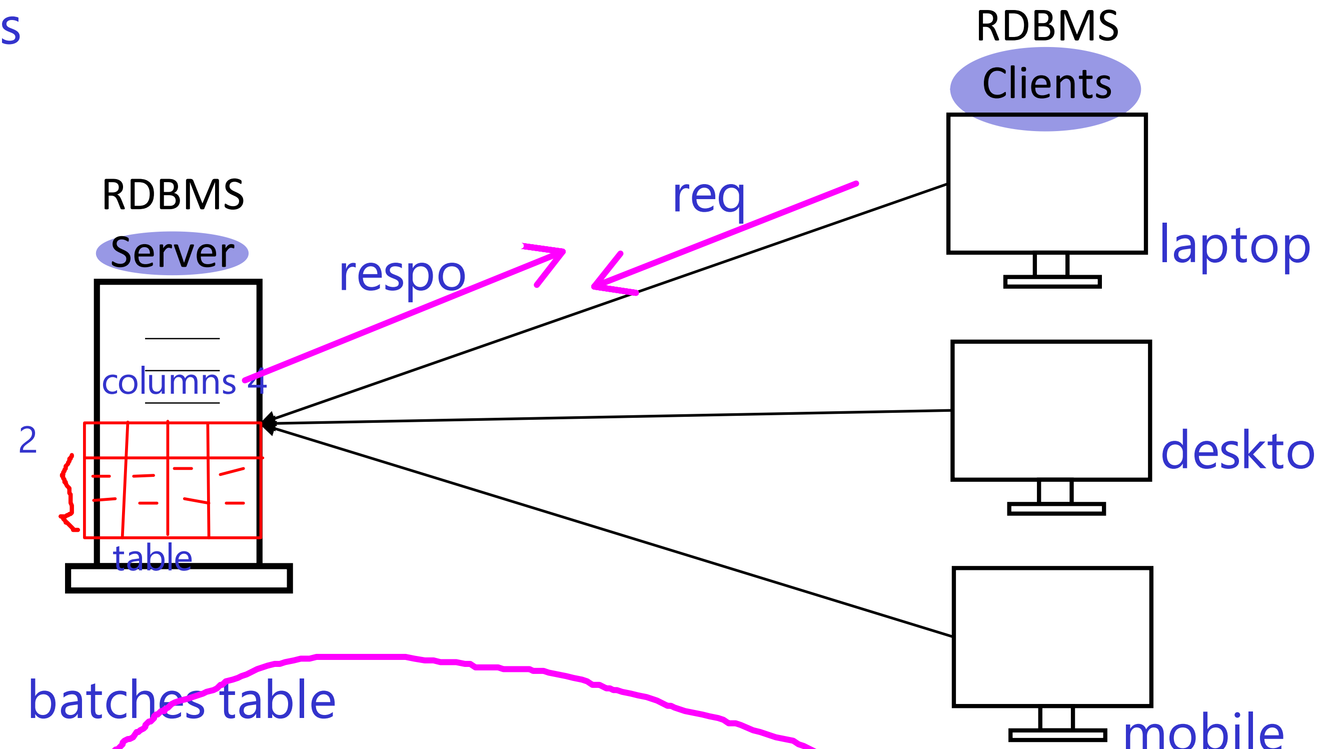
RDBMS relational DBMS

rows

- **Every enterprise application need to manage data.**
- **RDBMS is relational DBMS than manages structured data.**
- **Data is organized into tables, rows and columns. Tables are related to each other.**
- **All enterprise RDBMS follow server-client architecture, have built-in** join **relational capabilities, fully ACID transactions, based on Codd's rules.**

IBM IBM

- **DB2, Oracle, MS-SQL, MySQL, Postgre-SQL, MS-Access, SQLite, etc.**



batches table

id	name	date	ty
1	OM50	1	PC
2	PH24	30	pc
3	PH25	22	pc
4	CH06	5	pc

students

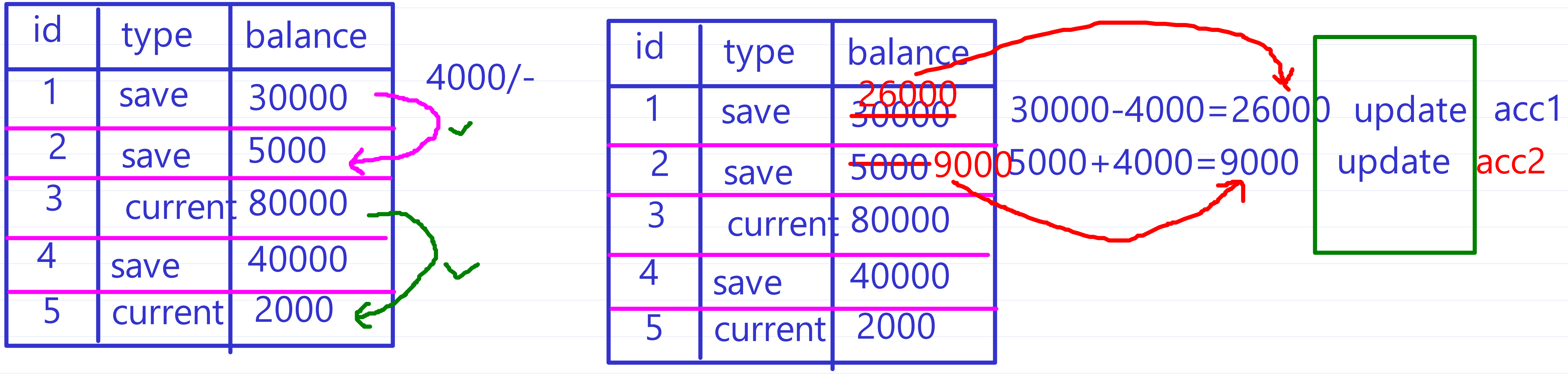
roll	name	batchid
1	a	1
2	b	3
3	c	3
4	d	1
5	e	1
6	f	4
7	G	2

SQL – Structured Query language

- **RDBMS data is processed with SQL queries.**
- **ANSI standardised in 1986 and ISO Standardization in 1987.**
- **Five major categories:**
  - **DDL: Data Definition Language e.g. CREATE, ALTER, DROP, RENAME.**
    - CREATE TABLE student(roll INT, name CHAR(40), batchid INT);
  - **DML: Data Manipulation Language e.g. INSERT, UPDATE, DELETE.**
    - INSERT INTO student VALUES(1, ‘Ravi’, 3);
    - UPDATE student SET name=‘Ravee’ WHERE roll=1;
    - DELETE FROM student WHERE roll=1;
  - **DQL: Data Query Language e.g. SELECT.**
    - SELECT * FROM student;
  - **DCL: Data Control Language e.g. CREATE USER, GRANT, REVOKE.**
  - **TCL: Transaction Control Language e.g. SAVEPOINT, COMMIT, ROLLBACK.**

Transaction characteristics- ACID

accounts table



id	type	balance
1	save	30000
2	save	5000
3	current	80000
4	save	40000
5	current	2000

4000/-

id	type	balance
1	save	~~30000~~ 26000
2	save	~~5000~~ 9000
3	current	80000
4	save	40000
5	current	2000

30000-4000=26000 update acc1

5000+4000=9000 update acc2

Transaction => set of DML queries executed as a single unit
i:e either all queries in Transaction are successful atomic
or all queries in Transaction are discared

consistent -> same result shown to all client

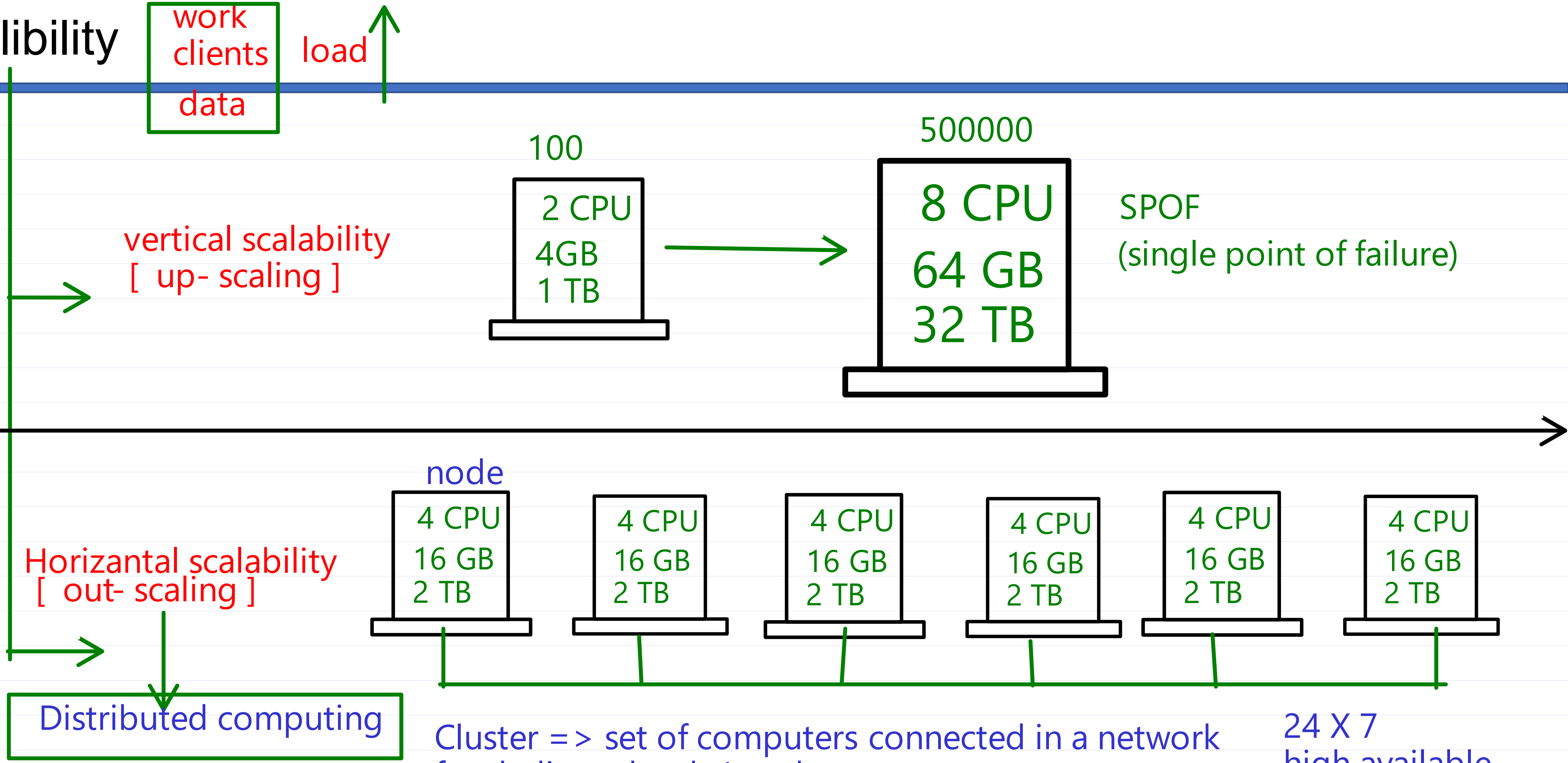
Isolated -> several Transaction can execute simultaneously without affecting each other

Durable -> all changes are saved permanetly

Scalibility

work
clients
data

load



vertical scalability
[up- scaling]

100

2 CPU
4GB
1 TB

500000

8 CPU
64 GB
32 TB

SPOF
(single point of failure)

Horizantal scalability
[out- scaling]

node

4 CPU 16 GB 2 TB	4 CPU 16 GB 2 TB	4 CPU 16 GB 2 TB	4 CPU 16 GB 2 TB	4 CPU 16 GB 2 TB	4 CPU 16 GB 2 TB

Distributed computing

Cluster => set of computers connected in a network
for dedicated task / work

24 X 7
high available
high scalable

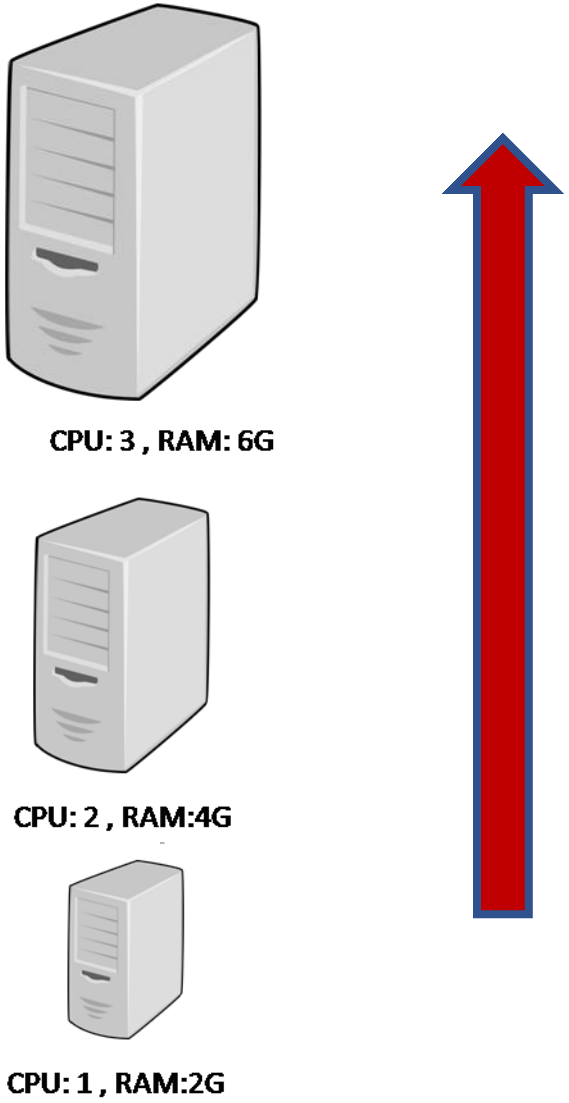
Scaling

- Scalability is the ability of a system to expand to meet your business needs.

- Scalability describes a elasticity of the system, ability to adapt to change and demand.

- Good scalability ensures the quality of your service.

Vertical Scaling



CPU: 3 , RAM: 6G

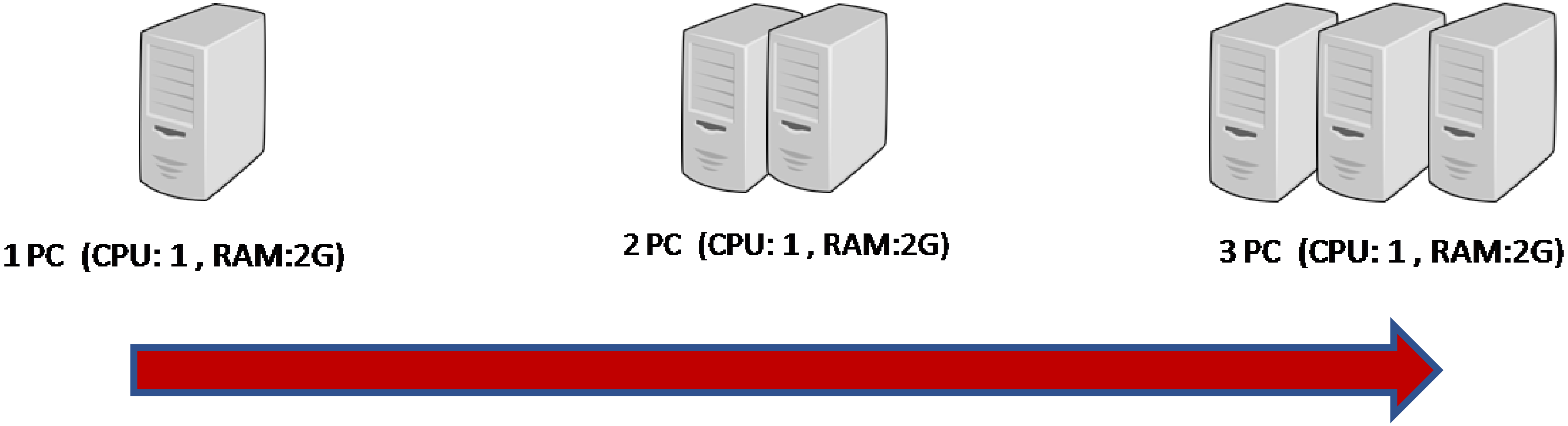
CPU: 2 , RAM:4G

CPU: 1 , RAM:2G

Vertical scaling describes adding more resources to your current machines.

- ✓ increase memory in the system
- ✓ expanding storage by adding hard drives
- ✓ upgrading the CPUs.
- ✓ upgrading network speed.

Horizontal Scaling



1 PC (CPU: 1 , RAM:2G)

2 PC (CPU: 1 , RAM:2G)

3 PC (CPU: 1 , RAM:2G)

Horizontal Scaling refers to adding additional nodes or machines to your infrastructure to cope with new demands.

- ✓ adding a new computer to a distributed software application

NoSQL Databases

carlo strozzi ->1998
NoSQL

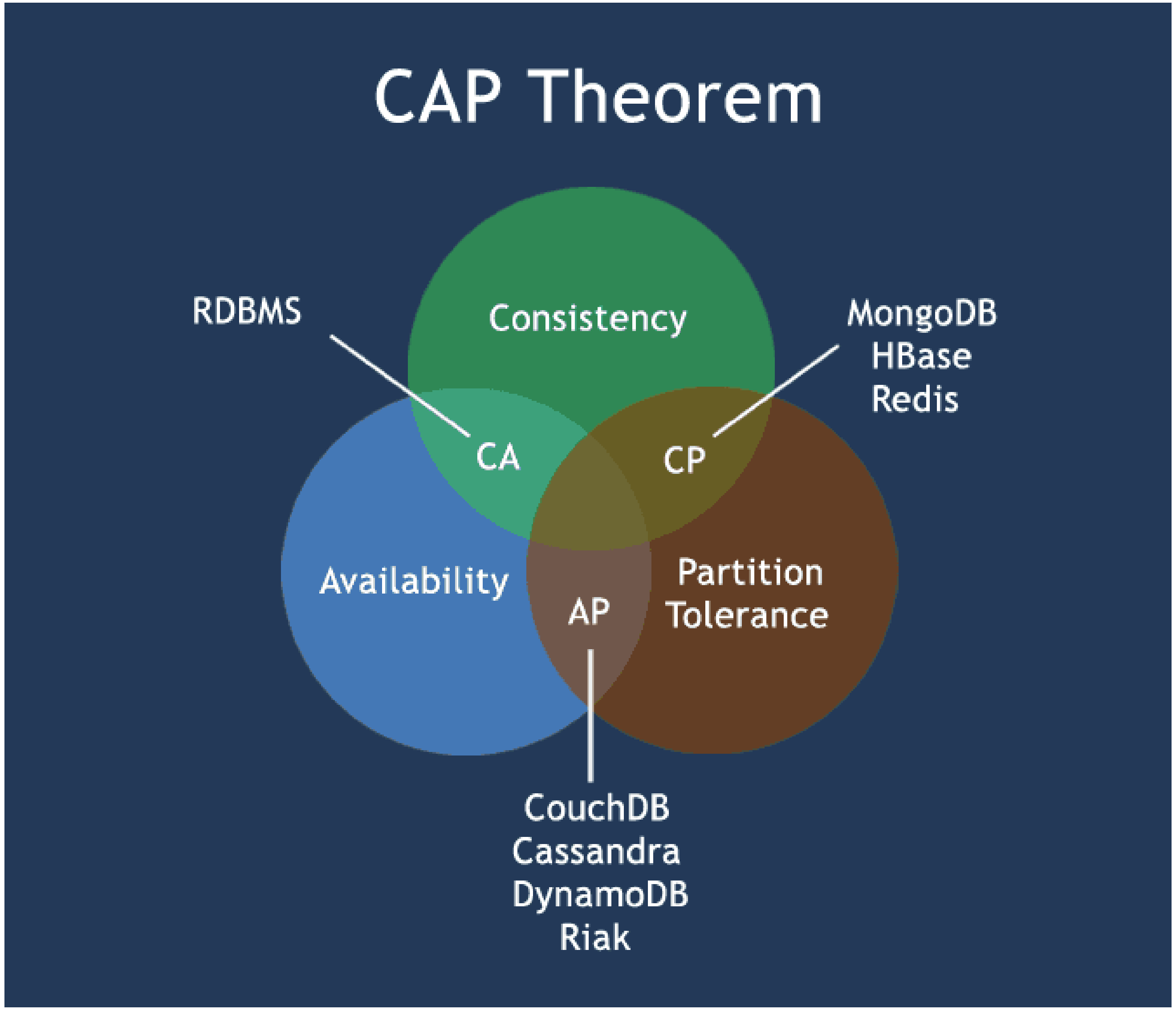
2004 ->conf
#NoSQL

@2006-7

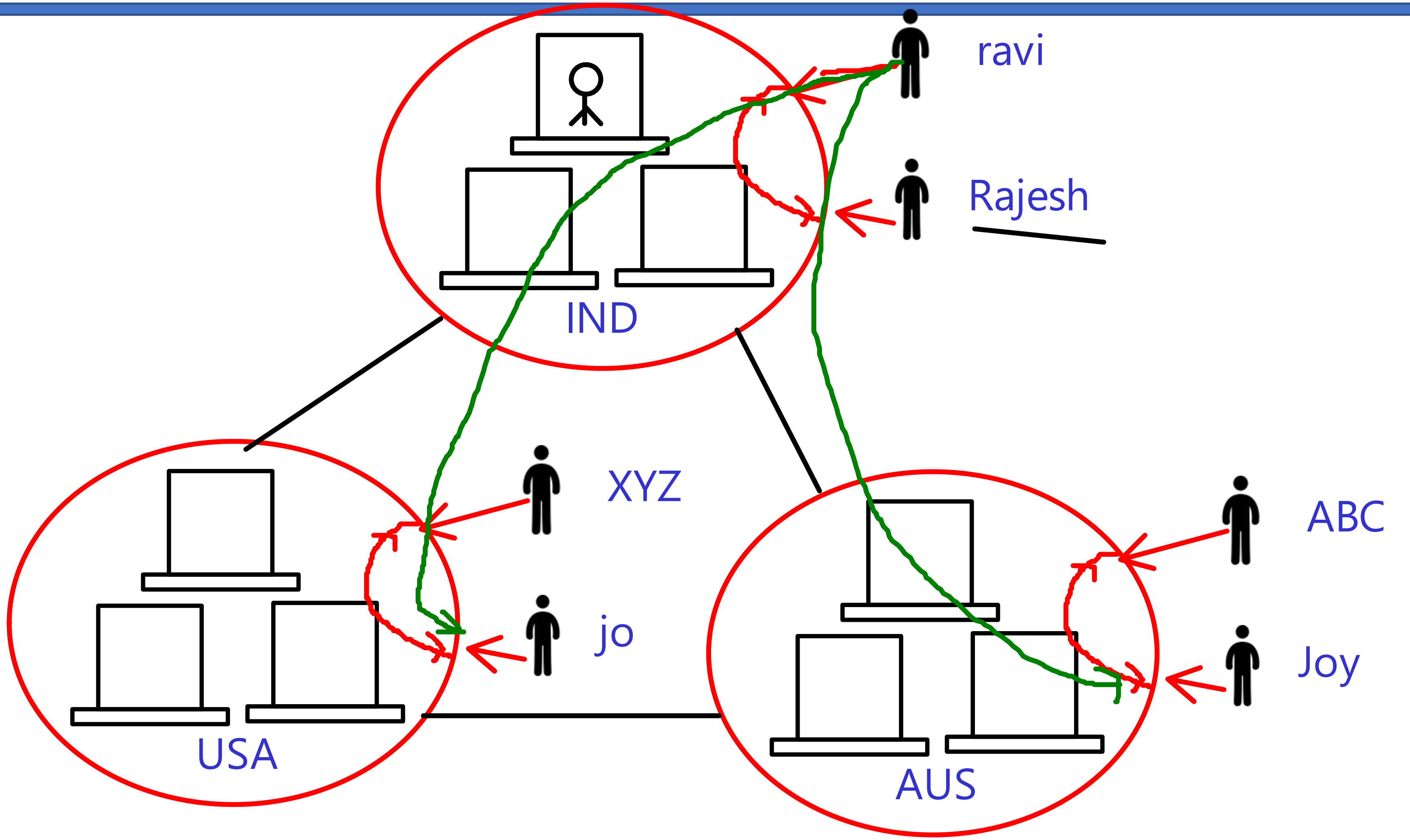
- Stands for Not Only SQL [Beyond SQL]
- Manages structured and semi-structured data.
- Prioritizes high performance, high availability and scalability
- Designed for Horizontal scaling. Reliable, fault tolerant, Better performance/Speed.
- No declarative query language
- Uses: Huge data (TBs), Many Read/Write ops, Scalable, Flexible schema.
- Don’t use if: Need high consistency, Multiple relations
- BASE transactions and Based on CAP Theorem

CAP Theorem (Brewer's Theorem)

- **Consistency - Data is consistent after operation. After an update operation, all clients see the same data.**
- **Availability - System is always on (i.e. service guarantee), no downtime.**
- **Partition Tolerance - System continues to function even the communication among the servers is unreliable.**



facebook



BASE

BA => Basically Available

system running 24X7

S => Soft state

Data is auto transferred to all node in cluster

E => Eventual consistency

same data visible to all cilent eventually

MCQ

Q: 1. ROLLBACK is _______________ type command .

A. DCL

B. TCL

C. DDL

D. DDD

MCQ

Q: 1. ROLLBACK is _______________ type command .

A. DCL

B. TCL

C. DDL

D. DDD

MCQ

Q: 1. _______________ command is used to delete table.

A. FREE

B. DELEET

C. DROP

D. RELISED

MCQ

Q: 1. _______________ command is used to delete table.

A. FREE

B. DELEET

C. DROP

D. RELISED

MCQ

Q: 1. Which one is not in v’s of Big Data?

A. Variety

B. Velocity

C. Volatile

D. volume

MCQ

Q: 1. Which one is not in v's of Big Data?

A. Variety -> Data can be unstructured, semi-structured or structured

B. Velocity -> Data generated with high speed

C. Volatile

D. volume -> Hugh amount of data

Veracity in Big data means_______________.

A. The data is generated with high speed

B. The data is huge

C. The data is reliable and trustworthy

D. The data management

MCQ

Veracity in Big data means_______________.

A. The data is generated with high speed -> Velocity

B. The data is huge -> volume

C. The data is reliable and trustworthy -> Veracity

D. The data management -> database

In RDBMS data is stored in _______________.

A. document

B. tables

C. collection

D. keys

In RDBMS data is stored in _______________.

A. document

B. tables

C. collection

D. keys

____________ introduced the NoSQL concept in 1998.

A. Cassandra

B. Devid Sam

C. Carl Strozzi

D. E.F. CODD

____________ introduced the NoSQL concept in 1998.

A. Cassandra

B. Devid Sam

C. Carl Strozzi

D. E.F. CODD

Thank you!

Pradnyaa S. Dindorkar <pradnya@sunbeaminfo.com>