SUNBEAM

Fundamentals of Data Engineering

Trainer: Nilesh Ghule

Big Data

5 que

Quiz 1 → 15th }
Quiz 2 → 16th } 10 mcq.

D.E

module end quiz } 20 mcq.

1. Databases
2. Big Data
3. Cloud Computing
4. ...

Pre-requisites

1. Computer Fundamentals
2. Operating Systems
3. Networking
4. Programming : Python, Java, ...

Introduction

- **Big Data Fundamentals**
  - Evolution of Data Engineering | V’s: Volume, Velocity, Variety, Veracity, Value
- **Databases**
  - RDBMS - ACID, SQL (basic concept only) I NoSQL - BASE, CAP theorem
- **Data warehouse - OLAP vs OLTP**
  - Data cleansing, Data transformations and Data modelling I Data warehouse vs Data mart
- **Data Engineering Life Cycle**
  - Source → Ingestion → Storage → Transformation → Serving
  - Ingestion: ETL vs ELT
  - Storage: Distributed storage, Storage services I 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

Database & Warehouse

- ✓ File IO - Data
- ✓ 1970: E.F. Codd
  → Relational DBMS
- ✓ 1980: RDBMS popular
- ✓ 1991: Data Warehouse
  ↳ Data analysis.

SQL

Internet & DotCom

- ✓ 1993: WWW
  - business on internet.
- ✓ 1995: Java popular

NoSQL Database

- ✓ 1998: Carlo Strozzi
- ✓ High performance, Huge Data, ...

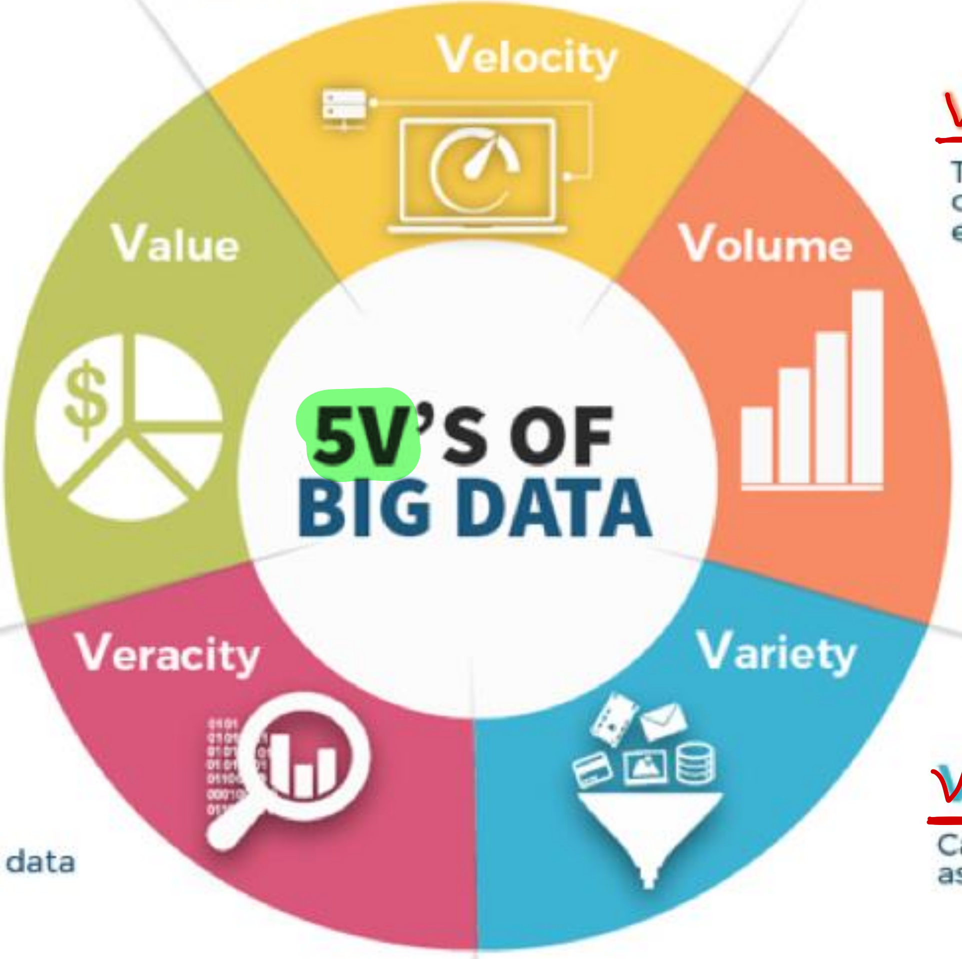
MPP & Big Data Tech

- ✓ multi-core: 2000s
- ✓ GPU for data process: 2005
- ✓ Grid Computing - Super Computer
- ✓ Distributed Computing Storage

Cloud Computing

- ✓ Data Centers
  ↳ AWS, GCP, Azure, Smart
- ✓ Pay per use.

Big Data characteristics



Velocity

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

Volume 100s TB+

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

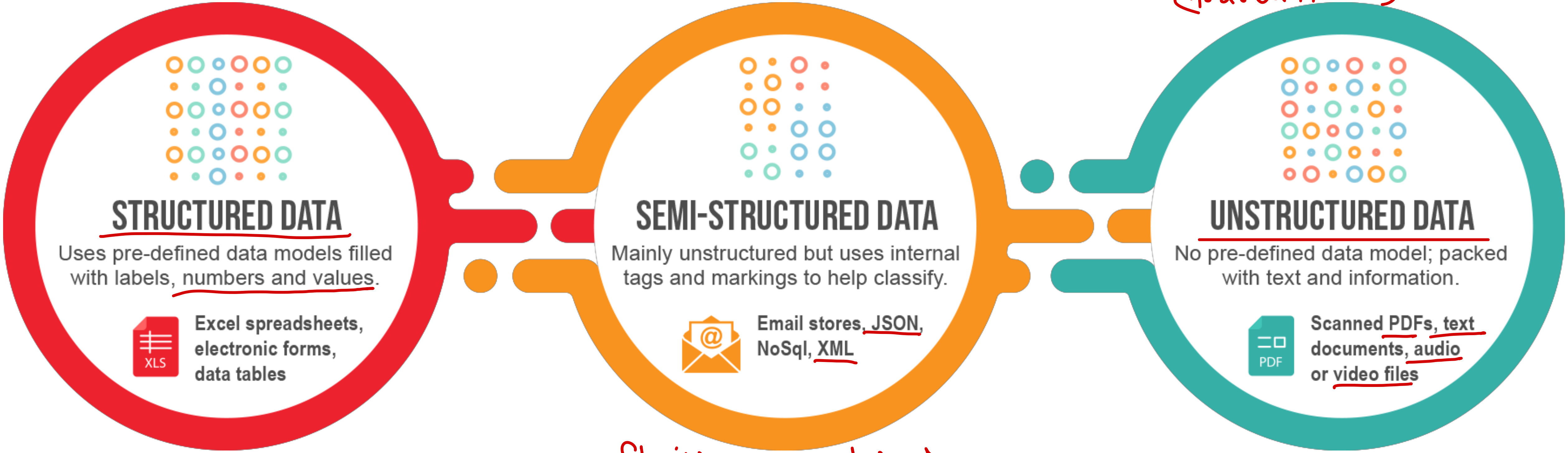
Can use structured as well as unstructured data.

Types of Data

RDBMS 2

Col1	Col2	Col3

files have fixed formats but contents cannot be understood programmatic (traditional).



STRUCTURED DATA

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

XLS — 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.

PDF — Scanned PDFs, text documents, audio or video files

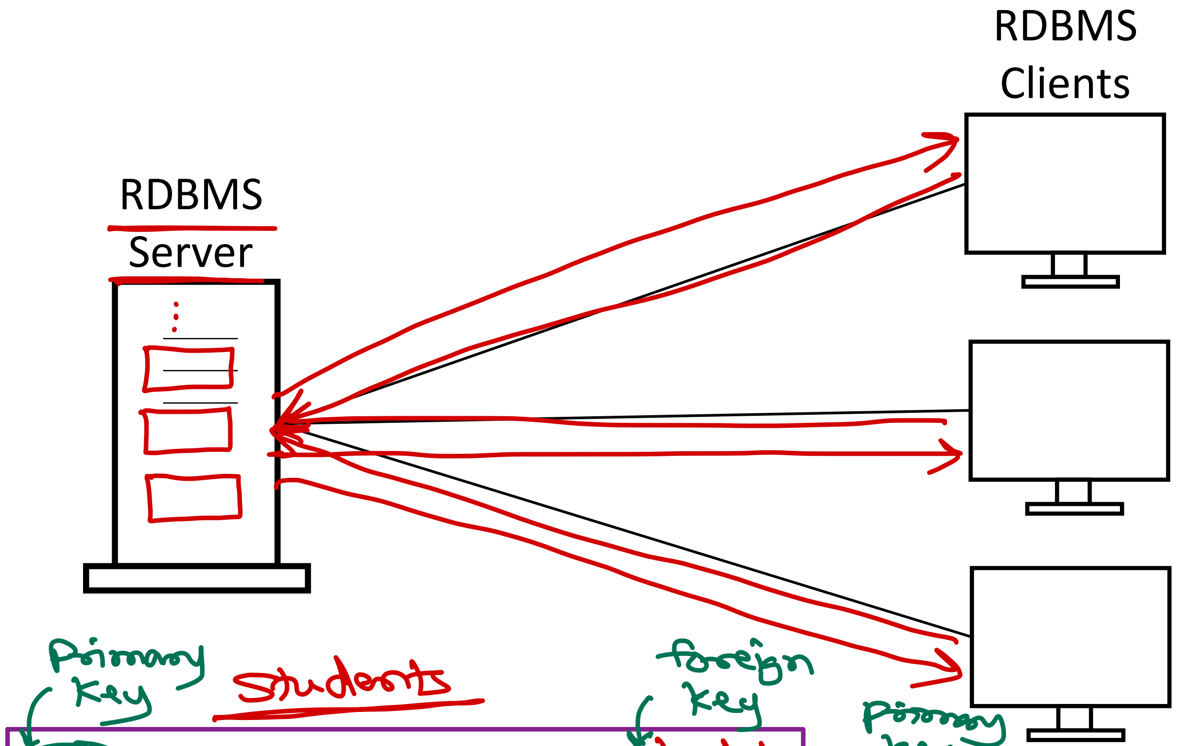
flexible schema/structure.

RDBMS

DBMS : CRUD ops → Create/Insert, Retrieve/Select, Update/change, Delete

1970 : Codd rules → RDBMS

- 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 relational capabilities, fully ACID transactions, based on Codd's rules.
- DB2, Oracle, MS-SQL, MySQL, Postgre-SQL, MS-Access, SQLite, etc.



Students

Primary Key → roll

Foreign key → batch

roll	name	marks	batch
1	James	97	OM49
2	John	93	PH23
3	Mary	95	PM30
4	Smith	96	PH23
5	David	94	PH23

batches

Primary key → batch

batch	loc	course
OM49	Online	PreCAT
PM30	Market	PreCAT
PH23	Hinj	PreCAT

* — 1

SQL – Structured Query language

- RDBMS data is processed with SQL queries.
- ANSI standardised in 1986. / ISO std in 1987.
- Five major categories: related to structure / schema of data.
  - DDL: Data Definition Language e.g. CREATE, ALTER, DROP, RENAME.
    - CREATE TABLE people(id INT, name CHAR(40), birth DATE);
  - DML: Data Manipulation Language e.g. INSERT, UPDATE, DELETE.
    - INSERT INTO people VALUES(1, ‘Nilesh’, ‘1983-09-28’);
    - UPDATE people SET name=‘NILESH’ WHERE id=1;
    - DELETE FROM people WHERE id=1;
  - DQL: Data Query Language e.g. SELECT.
    - SELECT * FROM people;
  - DCL: Data Control Language e.g. CREATE USER, GRANT, REVOKE.
  - TCL: Transaction Control Language e.g. SAVEPOINT, COMMIT, ROLLBACK.

RDBMS — ACID Transactions

accounts

id	type	balance
1	Saving	10000
2	Saving	2000
3	Current	50000
4	Saving	20000

3000 ✓

accounts

inconsistent

id	type	balance			
1	Saving	~~10000~~ 7000	← UPDATE	bal =7000	✓ ✓ ✗
2	Saving	~~2000~~ 5000	← UPDATE	bal =5000	✗ ✓ ✗
3	Current	50000			
4	Saving	20000			

Tx is set of DML queries that is executed as a single unit.

Either all queries in tx should be completed → COMMIT

OR

all queries in tx should be discarded. → ROLLBACK

① START TRASACTION;

② UPDATE accounts SET bal=7000 WHERE id=1;

③ UPDATE accounts SET bal=5000 WHERE id=2;

④ COMMIT; or ROLLBACK;

① ATOMIC → All queries success or fail.

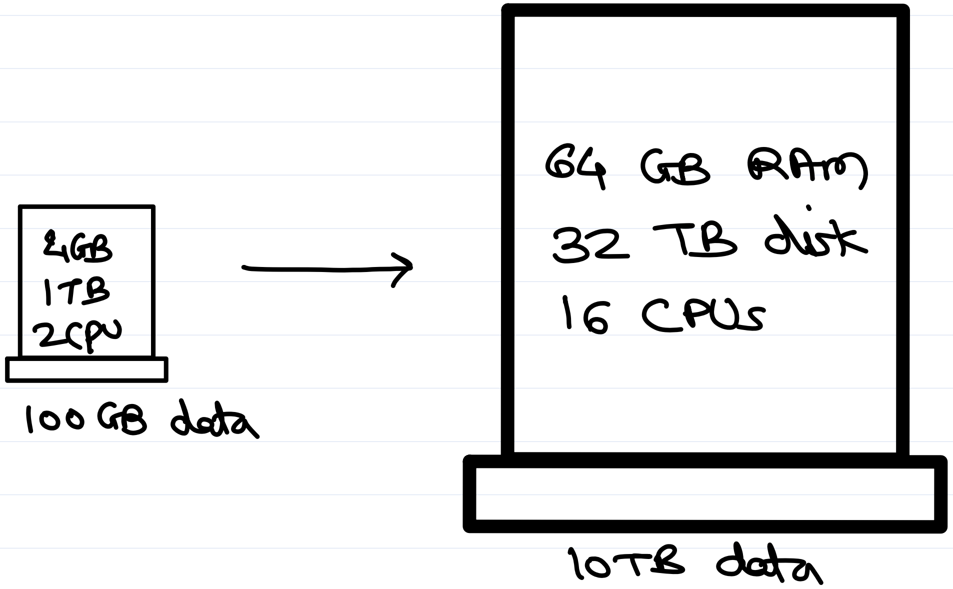
② CONSISTENT → Same data visible to all clients.

③ ISOLATED → Multiple tx will execute independently.

④ DURABLE → Final results must be stored on disk.

Scalability

Scalability
- Number of clients
- amount of data



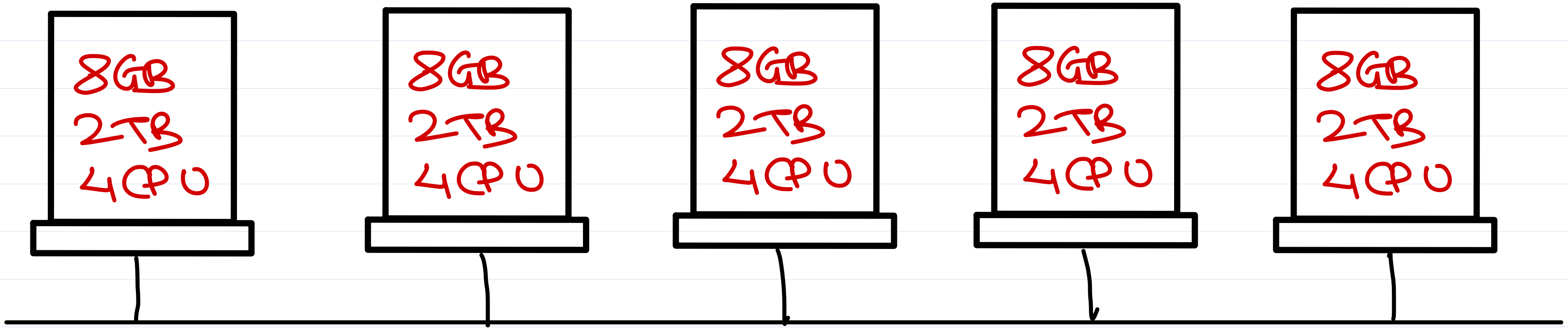
vertical scaling
OR
up-scaling

RDBMS scaling

distributed systems

horizontal scaling
OR
out-scaling

NoSQL scalability



cluster = set of computers for dedicated task in a network

NoSQL Databases

1998 - Carlo Strozzi

his own db without SQL

NoSQL = Anti-SQL.

Last.fm - international conference

2009 - invitation #nosql

- nosql become popular.

- Stands for Not Only SQL
- Manages structured and semi-structured data.
- Prioritizes high performance, high availability and scalability

  24×7 ✓
- Designed for Horizontal scaling. Reliable, fault tolerant, Better performance/Speed.
- No declarative query language → different languages for diff dbs.
- 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

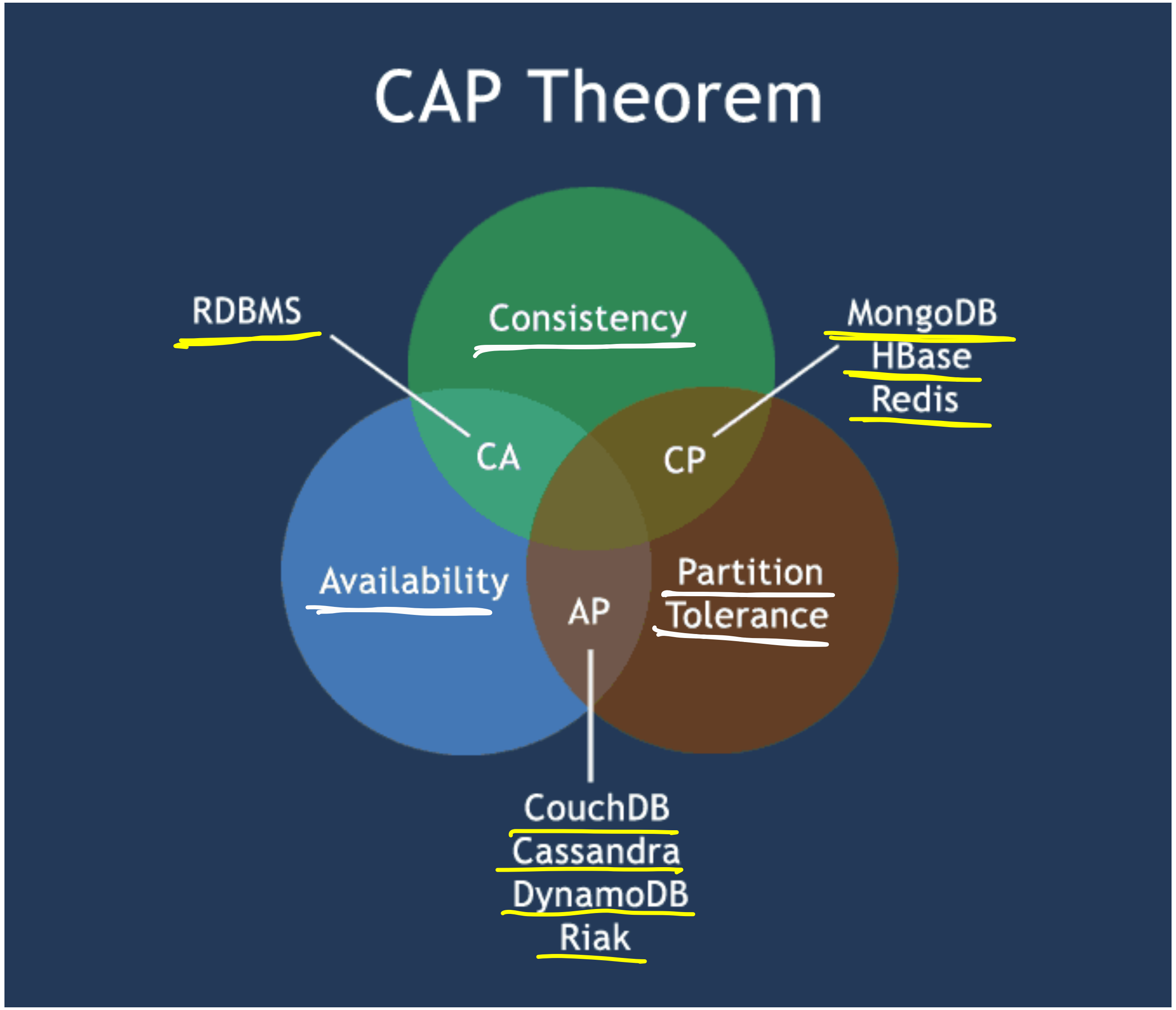
  → Consistency

  → Availability

  → Partition Tolerance

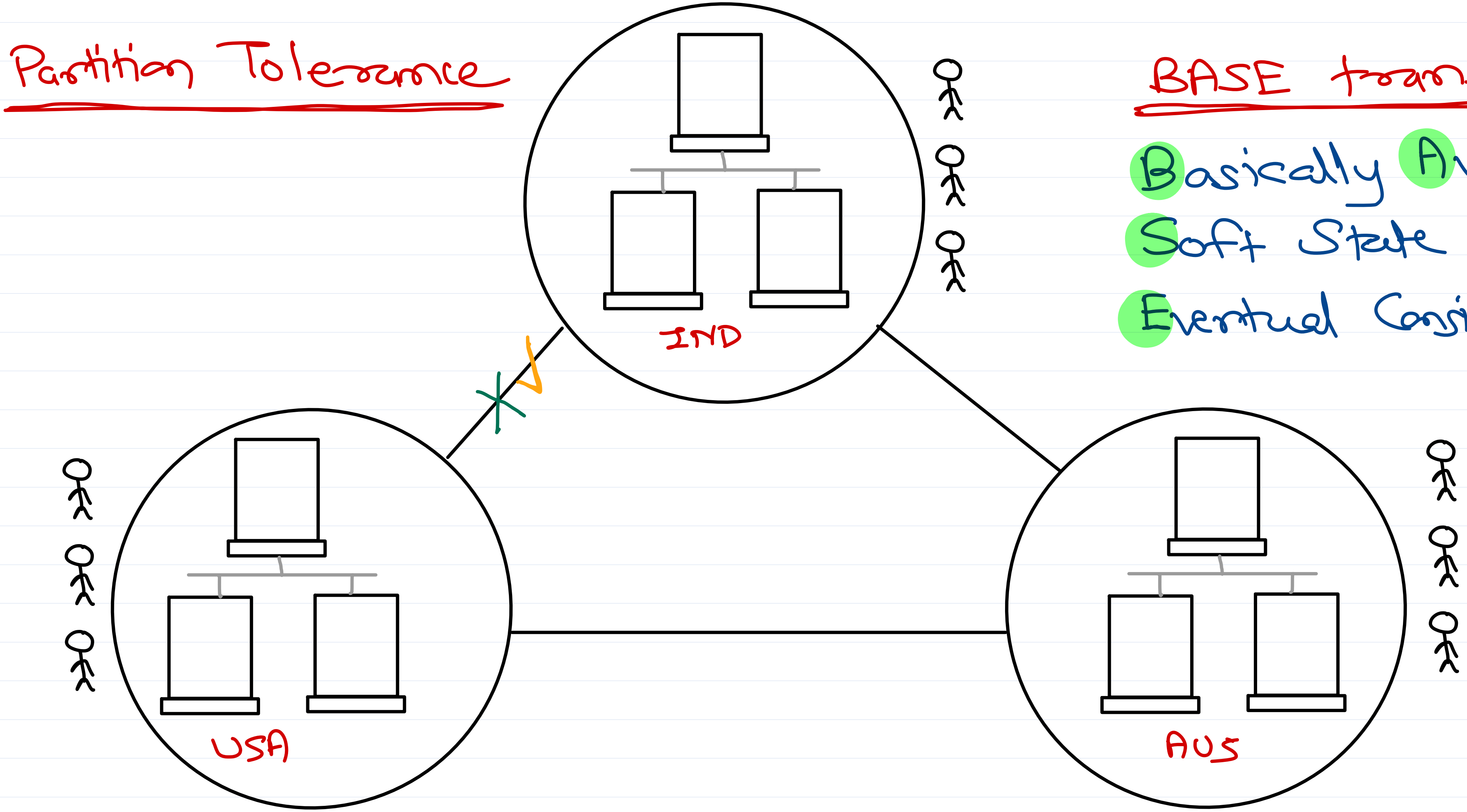
CAP Theorem - a.k.a. Brewer's Theorm

- 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. 24x7
- Partition Tolerance - System continues to function even the communication among the servers is unreliable.



NoSQL — Partition Tolerance and BASE Transactions

Partition Tolerance



BASE transactions

Basically Available

Soft State

Eventual Consistency

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>