SQL vs NoSQL

Day / Date (Tashfeen Abbasi) (i22-2041) DS-D (i22-1851) (Laiba Mazhar)

- DML (insert, update, delete, select)
  DDL (create table, views, alter, drop...)
- Acid property
- Features of NoSQL:
  - Large data volumes (Structured & unstructured)
  - Scable replication (maintaing multiple copies of data)
  - distributed databases (multiple nodes or cluster)
  - Answers quickly
  - Schema less
  - Mostly queried, few updates
  - Simpler and fast.
- BASE properties:-
  - Basically available
    - (priotrize high availibility of data over immediate consistency)
    - data replicated / so (data not lost)
    - rather than enforcing strict consistency guarantees, NoSQL provides uninterrupted access to data.
  - Soft State
    - lack of immediate consistency, data values may change over time.
    - Stores don't have to be write-consistent, nor replicas have to be mutually consistent eg., newyork me post upload hui, wahi ek dum dakh lae ek banday nay. PK tokoyo me kuch seconds baad.
  - Eventually consistent.
    - NoSQL do not guarantee immediate consistancy. Strive to achieve consistency over time.

RDBMS → • two tables are needed.
• great at consistency
• okay at availability
• not so great at partioning.

Day / Date

- **TYPES OF NoSQL**

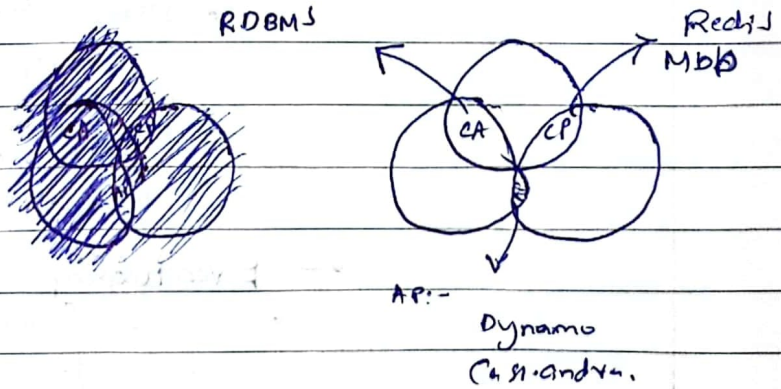
  - Mongodb (document)
  - Redis (key-value)
  - Cassandra, Hbase, Bigtable (wide-columns) (more flexibility in schema design)
  - Neo4j, Gremlin (graph database)

- **Cap theorum / Brewer theorum**

  - Consistency (most recent writes or error received e.g., all nodes see same data)
  - Availability (System remains responsive despite failures)
  - Partition tolerance (System continue to operate despite network partitions)

- **Why We don't use RDBMS for all types of data**

  We can use 3 features but not at same time.



Maxim..

Day / Date

- CA (if network issue occurs, sacrifice on partition tolerance and becomes unavailable.
- CP (availability is compromised, as system becomes ~~unavailable or experience down time~~)
- AP (prioritize uninterrupted access to data)

no

- Mathematical Model of RDBMS
  - Based on Relational Algebra
  - Writing SQL is possible, but is not a straightforward query.

- SQL vs NoSQL: Architectural difference
  - data placement on Disc in row and col
  - Architecture of Storage media
  - Query execution

- Types of Storage Media
  - Primary (CPU cache memory, main memory)
  - Secondary (HDD, SSD)
  - Tertiary (Removeable media)
  - Cache (• Static RAM • DRAM)
  - Mass (CD-ROM, DVD, tape drivers)

- Storage Organization of data bases
  - Persist data → • Stored beyond the lifespan of program's execution
  - Transient data → • managed by using data structures and memory management techniques
  - File organization → • records are physically placed on disk
    • records are accessed

Maxim

Day / Date

- **Single-sided disk and Disk Pack**
  - Single-sided disk:
    - records data on one side of disk
  - Disk Pack:
    - Collection of HDD stacked together.
    - data stored on both sides.
    - double-sided
- **Sector on a Disk**
  - Smallest unit of data
  - Generally, 512 bytes of data, newer may use 4096 bytes (4kb) to improve efficiency
- **Disk- Tracks**
  - Information stored on disk → Concentric circles of small width.
  - Each circle is called track
  - In disk pack, tracks with same diameter on different surfaces are called cylinder
  - Data on cylinder is retrived much faster
- **Tracks and Sectors:**
  - Tracks on disk → • from few thousand to 152,000
    • Capacity of track ranges from tens of kilobytes to 150 kbytes.
  - Tracks have more information → • divided into small sectors
    • hardcoded the division of sectors and can not be changed.
  - Blocks → • multiple sectors grouped together.
    • fixed during initialization & can not be changed dynamically.

512 - 8192 bytes.
2048 Kilobytes.

Maxim...

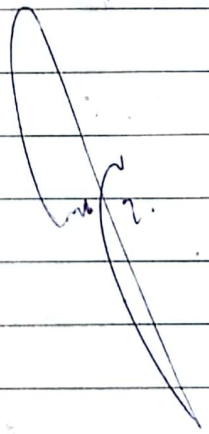
Surfaces → 10 | Blocks → 1MB

Day / Date | Tracks → 152,000 / 150 kb | Size of each block → 8192 bytes | Disc Space → ≈ 224GB

- Blocks are separated by fixed size interblock gaps.
  - gaps store control information written during disk initialization
  - This information determines which block on track follows each interblock gap. (sequential ordering)

Data Access:-

- Disk is random access addressable device.
- Transfer of data b/w main memory and disk takes place in units of disk blocks.
- Hardware address of block is combination of cylinder number + track number + block number.
- Actual hardware that reads or write a block → read/write head → part of system called disk drive



Contains read/write head ← DVD ROM ← disk drive

DVD ←

- A disk or diskpack is mounted in disk drive, which has → motor that rotates the disks.
- Read/write head → includes electronic component attached to mechanical arm.
- Diskpack with multiple surfaces controlled → several read/write heads - one for each surface.
- All arms connected to actuator ← attached to another electrical motor, which moves read/write heads and positions them precisely over the cylinder of tracks specified in block address.

Controlled by electrical motor ← device to move arms

Maxim.....

Day / Date

• Disk Access Time

Time = Seek time + Rotational delay + Transfer time + others.

- **Seek Time** → To transfer disk block, disk controller must first mechanically position R/W head on correct track. This time is seek time.
  - ↓ move head to desired track
  - 5 to 10 msec (desktop)
  - 3 to 8 msec (servers)
- **Rotational or Latency delay** → while the starting of desired block rotates into position under R/W head depends on rpm of disk.

Q:- Disks spin at 15000 RPM? Find Time Per second & Rotational delay.

1) Convert RPM to RPS

$$RPS = \frac{15000}{60} = 250 \text{ RPS}$$

2) Time per rotation

$$TPR = \frac{1}{250} = 0.004 = 4 \text{ msec}$$

3) Average rotational delay.

$$ARD = \frac{\text{Time per rotation}}{2} = 2 \text{ msec.}$$

- **Transfer Time** → 1-3 Mb/sec
  - Seek time & latency are much larger than block transfer time
  - Block size / Transfer rate
- **Other delays** →
  - Contention for Controller
  - Contention for memory
  - Typical value: 0

Maxim

Analytical queries ⇒ • Trend analysis
• dashboards
• Sales forcasting

Day / Date

All cols are read. ← Online transaction processing

- **Row Oriented Storage** (OLTP) → real time
  - Stored as rows in the disc
  - Single block I/O to the table, retrieves multiple rows with all their col
  - Multiple I/O are needed to find particular row in table but provides all col for that row,
  - DB Setup → We are not creating indexes on the columns.

use complex queries

- **Column Oriented Storage** (OLAP) (Analytical query)
  - Less I/O are required to retrive a column
  - A single block I/O read to table. Retrieves multiple columns with all matching row.
  - DBMS maintains information that which rowID is in which block on disc
  - query does not stores columns stores.

Row Based	Col based
• Optimal for R/W	Writes are slower.
• Compression is not efficient	Compression is great
• Aggregation is not efficient	Efficient aggregation
• Efficient queries when accessing multiple cols	Inefficient queries when accessing multiple cols.

no homo data.

operations like sum slower.

DBMS

Row : Inno DB

Col : Column Store.

Colum-Store optimization

Compression (10X improvement)

Maxim.....

Day / Date

Run Length encoding

- **Data Compression:** (RLE)
  - Algo to perform lossless data compression.
  - original form of data can be derived from it.
  - Stored as a single data value and count.
    - (original data) => AABCCD
    - (RLE) => 2A 1B 2C 1D
  - Save space while transmitting data.
    - Example: ABCCCCCCCCDEFGGG
    - Use flags + counts.
    - ABC!8DEFG!3