


Agenda:

- 1) What is Database
- 2) Why learn DB
- 3) Scale Curriculum
- 4) Types of DB
- 5) Intro to Relational DB
- 6) Keys ⇒ entire class (PK, FK, . . .)
- 7) Installation & Setup [MySQL, MySQL workbench]

Daily life Data :

- 1) Contacts ✓
- 2) Financial list / Banking data ✓
- 3) Todo list ✓
- 4) Photos / Songs / Assets / PDFs ✓
- 5) login / Password ✓
- 6) Messages ✓
- 7) Shopping list ✓

Excel sheet.

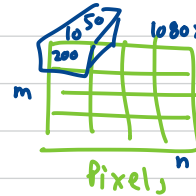
Item	Qty	Date.
Apples	2 kg	24 - Dec
mangoes	3 kg	26 - Dec
⋮	⋮	⋮

large Binary object.

Apps
↑
BLOB storage

• G-Drive , icloud

- Notes
- Excel , Sheet
- Document
- Notion



large Binary object
200, 10, 50
 $P = (R, G, B)$

Organisation → 1) Scaler

Students

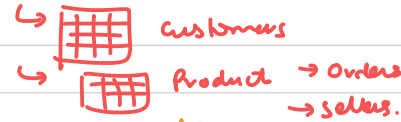
	Name	email	batch
1			
2			
3			
4			

Instructor

Videos			
id	Name	url	
1	Xyz	google.com/cloud/file.mp4	
2			

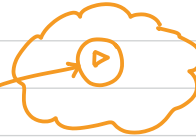
1000 video

2) Amazon



Binary objects
on BlobStorage

PDF, mp4, png, jpeg,



⑦

⑪

student.csv

```
name, batch, psp, attendance, coins, rank
Naman, (1) 94, (100) 0, 1
Amit, (2) 81, 70, 400, 1
Aditya, 1, 31, 100, 100, 2)
```

③ Aditya, Kumar
⑤

every batch.

Average attendance of Students in Batch 1.

↳ open the file

↳ Read line by line

↳ Tokenise the line ;

↳ token[1], token[3]

Batch id attendance

↳ add attendance & take average.

↳ create & update hashmap.

[File handling]

Java

↳ open

↳ read / search

↳ write.

tokens = s.split()

["Naman", "1", "94", "0", "100", "..."]

key

value → list of att.
→ 100

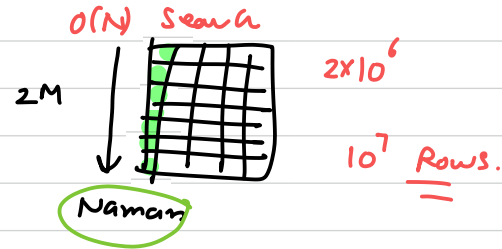
1 → [100, 100]

2 → [70]

⋮

Disadvantages of Files \Rightarrow you should DB.

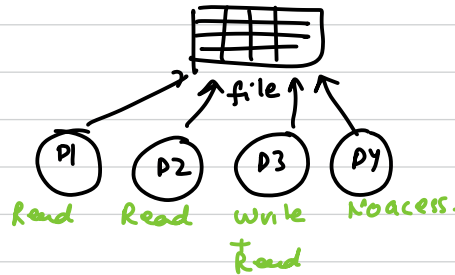
① Inefficient



② Integrity

↳ correctness of data
↳ datatype, wrong value, etc.

③ Security



$O(\log N)$ Using Indexing \rightarrow DB.

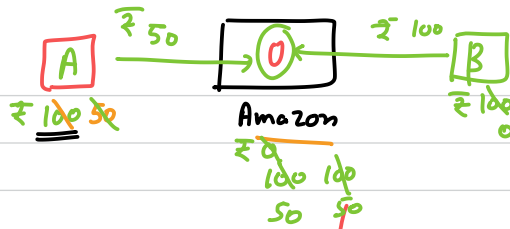
Separate class.

Balanced BST.
B+ Tree

④ Concurrency

\rightarrow Multiple users are working on same file.





2 Transactions

	A	B	Amz
Init	100	100	0
Final	50	0	50

→ 200
→ 100

Debit A

- Read A → X (100)
- X = X - 50 (50)
- Write X → A (50)

Credit Amazon

- Read Amz → X (0)
- X = X + 50 (50)
- Write X → Amz

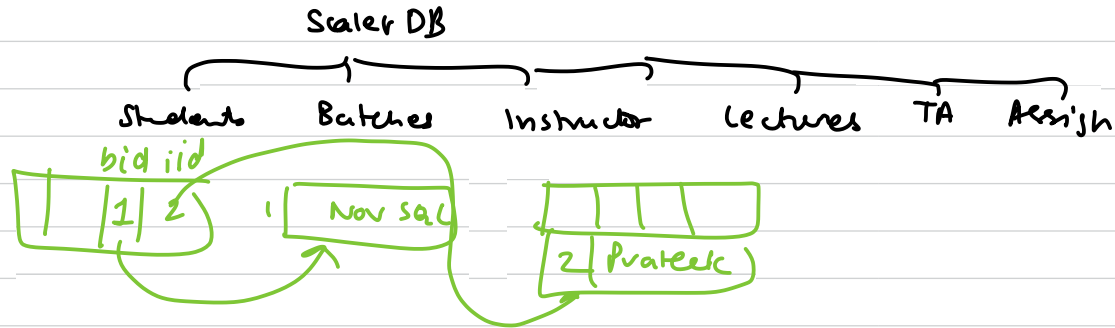
Debit B

- Read B → X 100
- X = X - 100 0
- Write X → B. 0

Credit Amazon

- Read Amz → X 0
- X = X + 100 100
- Write X → Amz

Database: colln of "related" data.

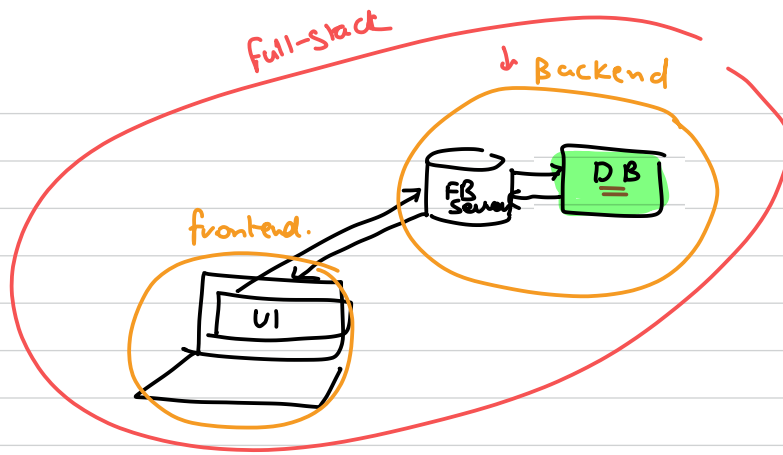


DBMS : Software that allows to efficient manage a DB.

↳ CREATE → READ
↳ UPDATE → DELETE (CRUD)

+ data integrity,
security,
concurrency.

SDE



Interviews

- ↳ SQL queries
- ↳ Schema Design.
 - ↳ Tables,
 - ↳ Relations,
 - ↳ PK, FK,
 - ↳ Indexes

Curriculum:

1. How DB work
2. SQL queries & Schema Design → SQL module
3. Scalability and Distributed DB → HLD Module.

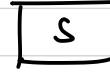
goals

- ↳ SQL code
- ↳ Internals -
- ↳ SD

Topics:

[12-13
lectures.]

- 1) Intro , Keys
- 2) Keys
- 3) CRUD -1
CRUD -2
- 4) Joins
- 5) Aggregate Queries
- 6) Subqueries
- 7) Indexing
- 8) Transaction -1
-2
- 9) Schema Design -1, 11



Types of DB

Focus-

Relational DB
RDMS

→ colln of related TABLES

→ MySQL, OracleDB,
PostgreSQL
SQL Server.

Non-Relational DB/
NO-SQL DBMS

→ everything except a table

→ document

→ K-V pairs

→ json

→ graph DB

⇒ Graph, MongoDB, Firebase

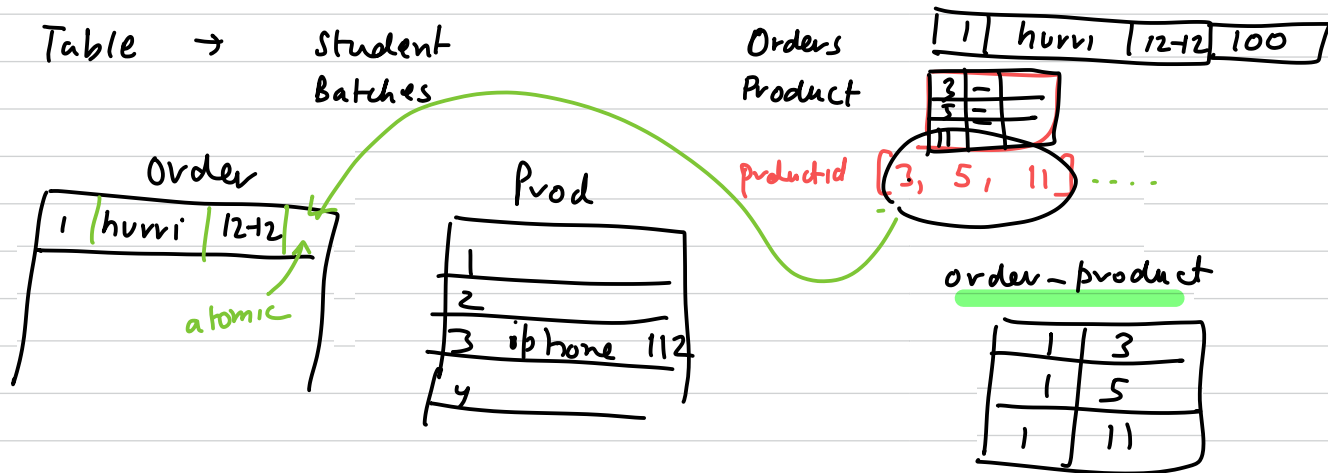
→ HLD Module

Properties of RDBMS :

①

Relational Databases represent a database as a collection of tables with each table storing information about something. This something can be an **entity** or a **relationship** between entities.

Example: I may have a table called students to store information about students of my batch (an entity). Similarly I may have a table called student_batches to store information about which student is in which batch (a relationship between entities).



2

Every row is unique. This means that in a table, no 2 rows can have same values for all columns. Example: In the students table, no 2 students can have same name, attendance and psp. There will be something different for example we might also want to store their roll number to distinguish 2 students having the same name.

3

All of the values present in a column hold the same data type. Example: In the students table, the name column will have string values, attendance column will have integer values and psp column will have float values. It cannot happen that for some students psp is a String.

4

Values are atomic.

Atomic

7	15	1 Dec
7	47	22 Dec
7	98	24 Dec

Person → orders

#	15
#	47
#	98

Personid	orders
7	15, 47, 98

list

Not atomic.

" 97, Iblur layout
Bangalore - 110085 "

okay!

Address

Find customers who
ordered pincode 110085.

1 city 2 state 3 pincode

```
[ SELECT city, pincode.  
  FROM customers;
```

5

The columns sequence is not guaranteed. This is very important. SQL standard doesn't guarantee that the columns will be stored in the same sequence as you define them. So, if you have a table with 3 columns: name, attendance, psp, it is not guaranteed that the data will be stored in the same sequence. So it is recommended to not rely on the sequence of columns and always use column names while writing queries. While MySQL guarantees that the order of columns shall be same as defined at time of creating table, it is not a part of SQL standard and hence not guaranteed by all databases and relying on order can cause issues if in future a new column is added in between.

6

The rows sequence is not guaranteed. Similar to columns, SQL doesn't guarantee the order in which rows shall be returned after any query. So, if you want to get rows in a particular order, you should always use ORDER BY clause in your query which we will learn about in the next class. So when you write an SQL query, don't assume that the first row will always be the same. The order of rows may change across multiple runs of same query. Having said that, MySQL does return rows in order of their primary key (we will learn about this later today), but again, don't rely on that as not guaranteed by SQL standard.

<u>id</u>	Name
1	A
2	B
3	C

ORDER BY

7

Name of every ^{col} should be unique.

X

id	name	name
1	Prat	Navay
2	Prat	Navay