

Introduction to DBMS & Relational Model

TABLE OF CONTENTS

1. Scaler's curriculum for SQL ✓
2. What is Database? ✓ ✓
3. Why are we studying it?
4. Types of databases
5. Introduction to RDBMS .
6. Introduction to Keys
 6.1. Super Keys }
 ↑



Notes



DSA



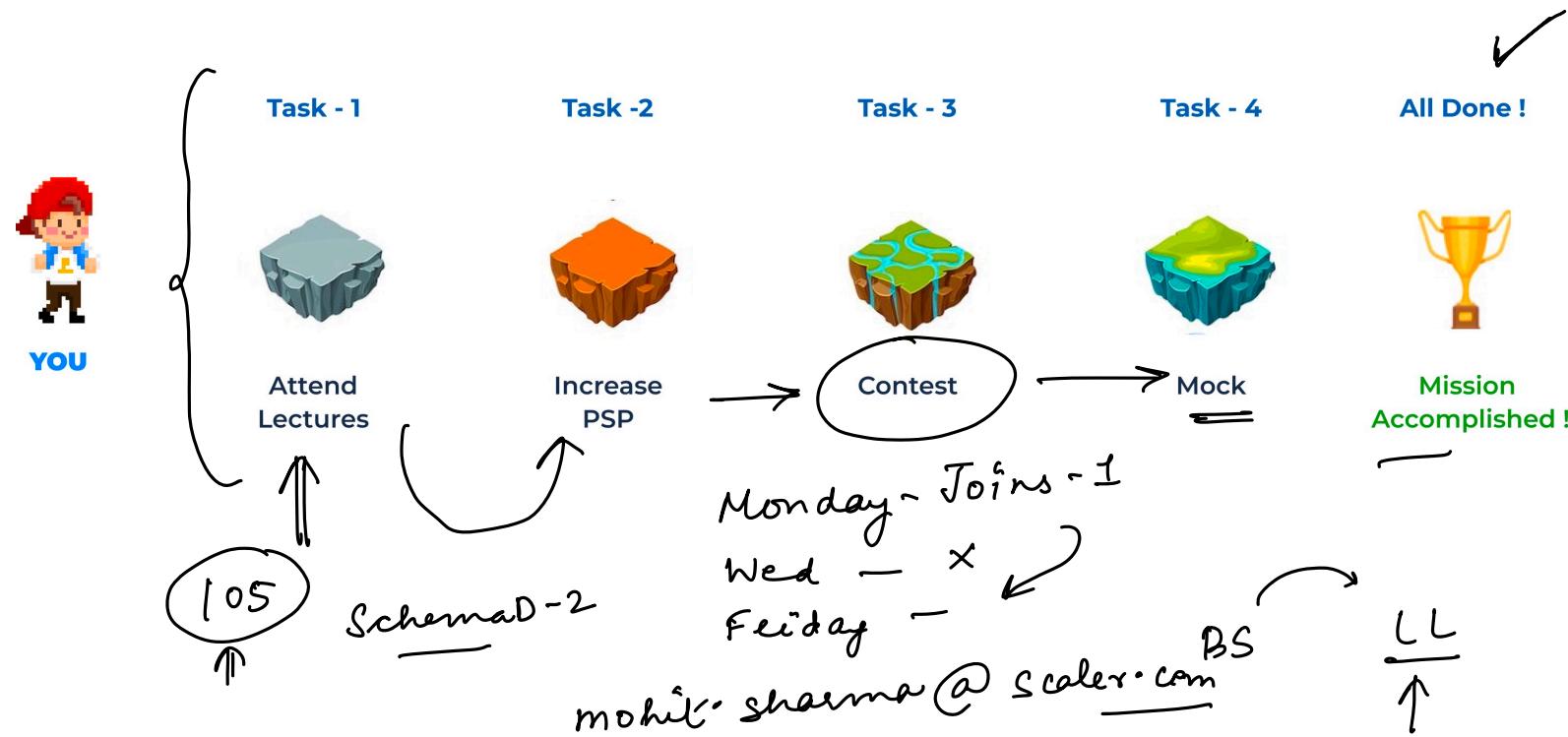
SQL (NON-DSA)



Not a DSA fan?

Dive into SQL's path to success

- How should you tackle **SQL module?**





Curriculum..

1. Intro to DBMS and SQL

2. Keys →

3. Crud → *2

4. Joins → *2 → 10 times

5. Aggregates

6. Subqueries

7. Indexes →

8. Transactions *2

9. Schema Design *2 → Scaler
Netflix

Triggers / Procedures

CTE

window fns

Extra references

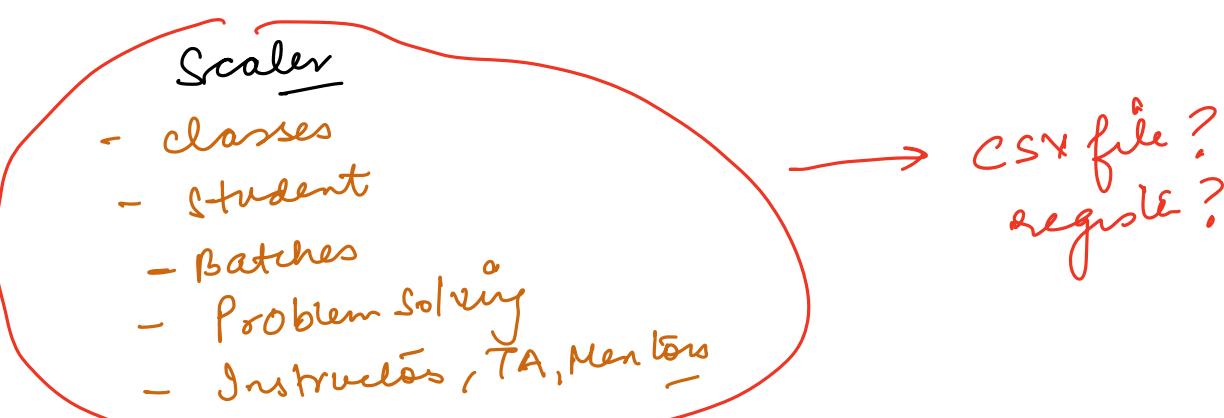
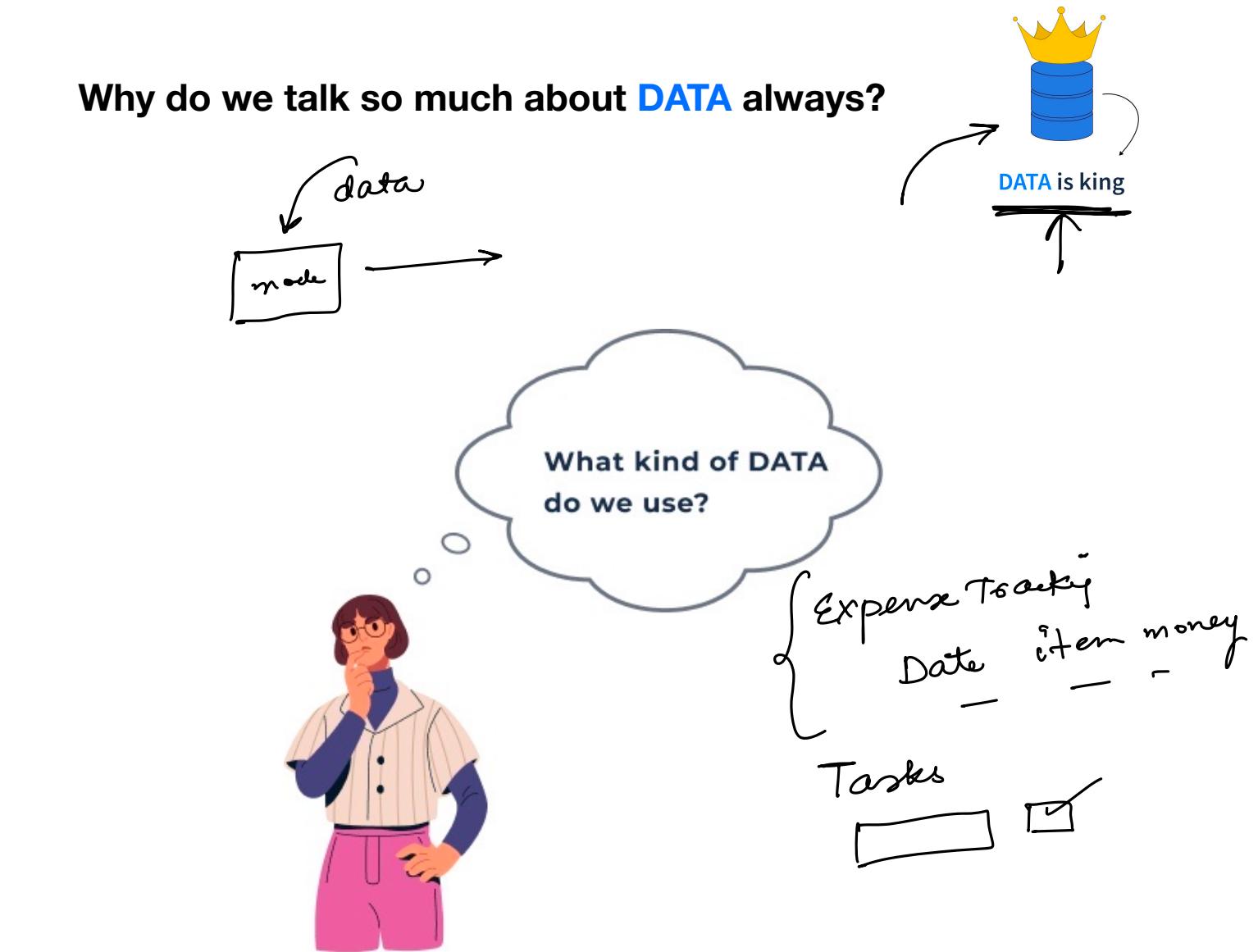
What isn't covered

1. Distributed databases
2. Scalability and related concepts like NoSQL
3. Database Sharding
4. Master Slave architecture
5. Database Replication

High level Design /
distributed systems



Why do we talk so much about **DATA** always?



Applications :



To-Do



Excel





< Question > : Find avg psp of students corresponding to their batches.



txt file? CSV, Excel

| | A | B | C | D |
|---|----|----------|-----|------------|
| 1 | ID | NAME | PSP | ATTENDANCE |
| 2 | 1 | Himanshu | 80 | 85 |
| 3 | 2 | Rahul | 75 | 90 |
| 4 | 3 | Krish | 95 | 95 |
| 5 | 4 | Rahul | 92 | 85 |
| 6 | 5 | Rohit | 80 | 88 |

b - id
↓



Drawbacks :

1. Inefficient

→ All of the tools that we discussed doesn't provide easy retrieval and manipulation of data.

2. Data Integrity

| K16 | A | B | C | D |
|-----|----|---------------|--------|------------|
| 1 | ID | NAME | PSP | ATTENDANCE |
| 2 | 1 | 95 + Himanshu | 80 | 85 |
| 3 | 2 | Rahul | 75 | 90 |
| 4 | 3 | Krish | 95 | 95 |
| 5 | 4 | Rahul | Topper | 85 |
| 6 | 5 | Rohit | 80 | 88 |

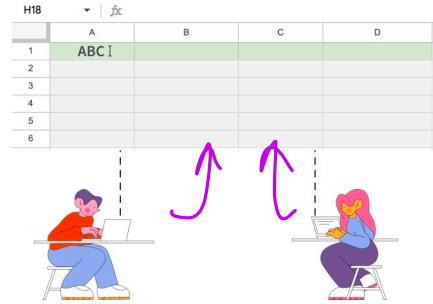
Inaccuracy



3. Concurrency

↓
If multiple people
works on the
same data at
same moment

↓
Inconsistency



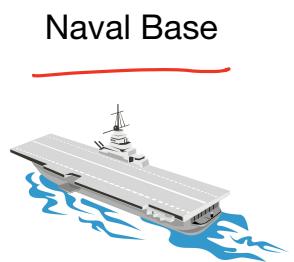
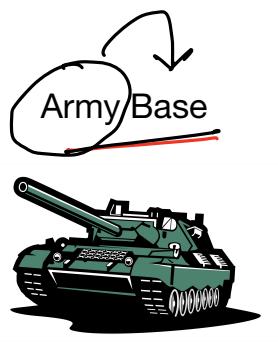
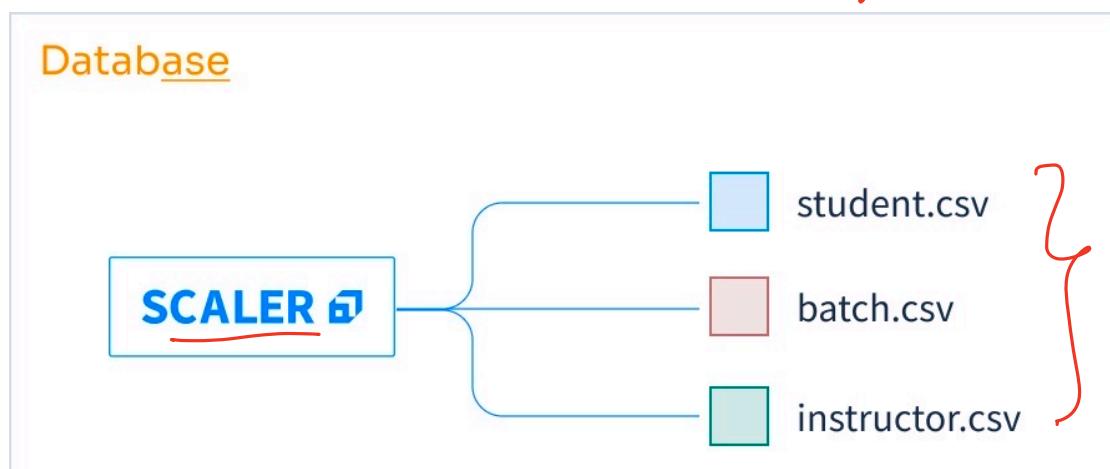
4. Security Issues



↑
level of access control, Encryption



What is Database?





Database Management System

- A DBMS as the name suggests is a software system that allows to efficiently manage a database.

- A DBMS allows us to create the following :

Create ✓
Read ✓
Update ✓
Delete ✓

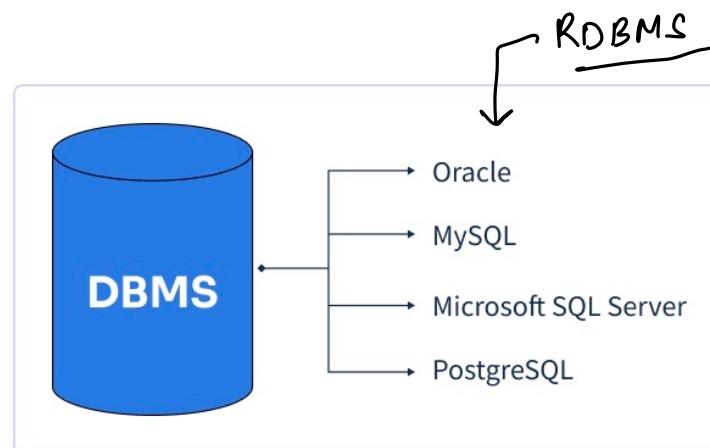
- It allows us to define rules to ensure :

1. Data Integrity ✓

2. Security and ✓

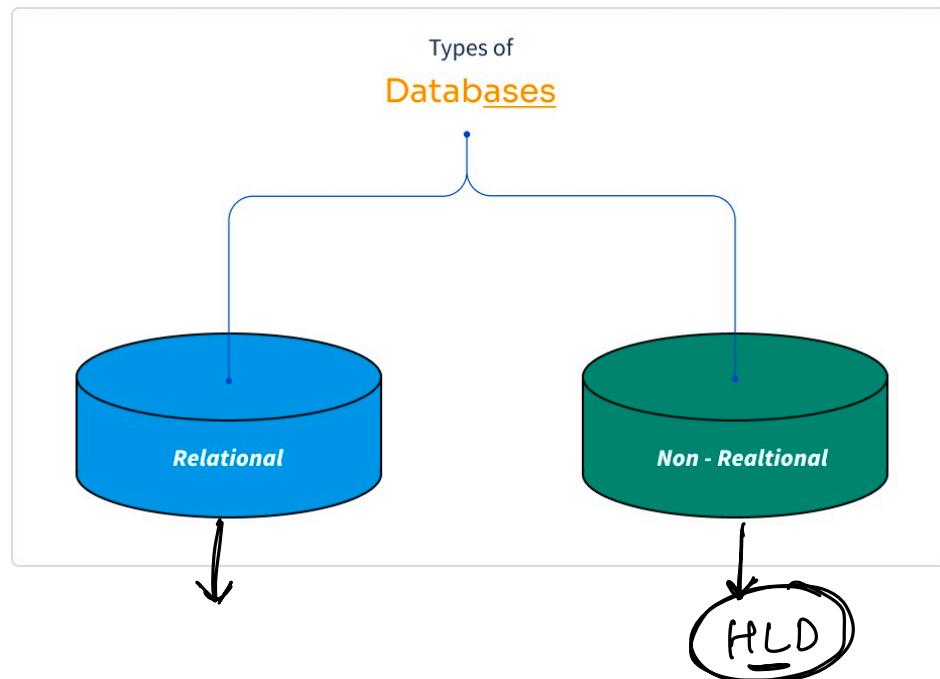
3. Concurrency ✓

SQL vs NoSQL





Types of Databases



1. Relational

allows you to store the data in the form of tables → rows → columns → field.

Students

| id | name | psp | attendance | b_id |
|----|----------|-----|------------|------|
| 1 | Himanshu | 80 | 85 | 2 |
| 2 | Rahul | 75 | 90 | 1 |
| 3 | Krish | 95 | 95 | 1 |
| 4 | Rahul | 92 | 85 | 2 |
| 5 | Rohit | 80 | 88 | 1 |

Batch

| id | name |
|----|------|
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |
| 5 | E |

Related

Annotations in red highlight specific data points: "Rahul" in the Students table, "1" in the b_id column of the Students table, and "1" in the id column of the Batch table. Orange arrows indicate relationships between the tables: one arrow points from the "b_id" column in the Students table to the "id" column in the Batch table, and another arrow points from the "id" column in the Batch table back to the "b_id" column in the Students table. A large orange oval encloses the entire Students table, and another large orange oval encloses the entire Batch table.



2. Non - Relational

- Don't store data in form of tables.
- Store data in form of documents, key-value pairs, graphs, etc.
- We will talk more about them in the [HLD Module](#).

json
{"name": "Mohit",
"id": 5}

Break : 10:25 pm



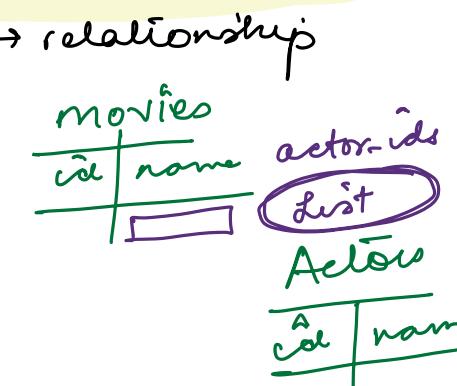


Properties of RDBMS

Rules

1. Relational Databases represent database as a collection of tables with each table storing information about something.

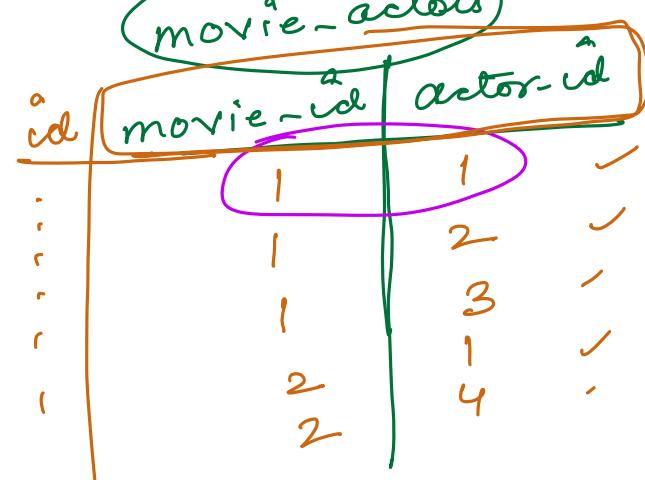
entities
Student, TA,
Batch, Instructor,
Movies, Actors



2. Every row is unique.

Students

| name | psp | attendance | b_id |
|----------|-----|------------|------|
| Himanshu | 80 | 85 | 2 |
| Rahul | 92 | 85 | 2 |
| Krish | 95 | 95 | 1 |
| Rahul | 92 | 85 | 2 |
| Rohit | 80 | 88 | 1 |



< Question > : Find psp of Rahul.



Students

| id | name | psp | attendance | b_id |
|----|----------|-----|------------|------|
| 1 | Himanshu | 80 | 85 | 2 |
| 2 | Rahul | 92 | 85 | 2 |
| 3 | Krish | 95 | 95 | 1 |
| 4 | Rahul | 92 | 85 | 2 |
| 5 | Rohit | 80 | 88 | 1 |

3. A column should have all values of same data type.

Students

| id | name | psp | attendance | b_id |
|----|----------|--------|------------|------|
| 1 | Himanshu | 80 | 85 | 2 |
| 2 | Rahul | 75 | 90 | 1 |
| 3 | Krish | 95 | 95 | 1 |
| 4 | Rahul | Topper | 85 | 2 |
| 5 | Rohit | 80 | 88 | 1 |

↑ throw an error



4. All values / cell should be atom.) atomic

atomic

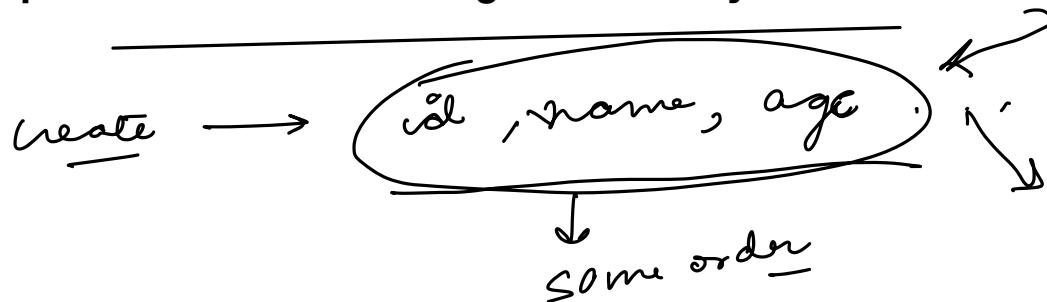
Students

| id | name | psp | phone.no | b_id |
|-----------|-------------|------------|----------------------|-------------|
| 1 | Himanshu | 80 | 956453789 | 2 |
| 2 | Rahul | 75 | 906453875 | 1 |
| 3 | Krish | 95 | 829376769, 806122348 | 1 |
| 4 | Rahul | 92 | 806122348 | 2 |
| 5 | Rohit | 80 | 762766434 | 1 |

829376762

retrieval

5. The sequence of column is not guaranteed by RDBMS.



6. The sequence of row is not guaranteed.



blob

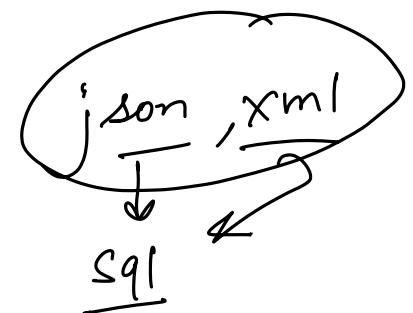
7. Name of every column has to be unique.

Students

| id | name | psp | phone.no | phone.no |
|----|----------|-----|-----------|----------|
| 1 | Himanshu | 80 | 829376769 | 72 |
| 2 | Rahul | 75 | 956453789 | Null |
| 3 | Krish | 95 | 906453875 | Null |
| 4 | Rahul | 92 | 806122348 | 98 |
| 5 | Rohit | 80 | 762766434 | 78 |

X

↑ X



< Question > : What is the phone number of Rahul?



Keys

Students

| name | psp | attendance | b_id |
|----------|-----|------------|------|
| Himanshu | 80 | 85 | 2 |
| Rahul | 92 | 85 | 2 |
| Krish | 95 | 95 | 1 |
| Rahul | 92 | 85 | 2 |
| Rohit | 80 | 88 | 1 |

(1)
(2)

which Rahul
should be
updated?

{ Each row should
be unique .

↓
keys

< Question > : Update psp of Rahul to 100.

Definition : Keys helps us to uniquely identify a row.



Types of Keys

1. Super Keys
2. Candidate Keys
3. Primary Keys
4. Composite Keys
5. Foreign Keys

1. Super Keys

Students

| id | name | psp | attendance | b_id |
|----|----------|-----|------------|------|
| 1 | Himanshu | 80 | 85 | 2 |
| 2 | Rahul | 75 | 90 | 1 |
| 3 | Krish | 95 | 95 | 1 |
| 4 | Rahul | 92 | 85 | 2 |
| 5 | Rohit | 80 | 88 | 1 |

1. Can ' name ' column uniquely identify row?

2. Can ' batch ' column uniquely identify row?

names can be
duplicate

multiple people
can belong to same
batch

**Column Name****Super Key**

name



psp



attendance



psp, name



id



id, name, psp, b_id



b_id



id, name



email



email, name



email, psp



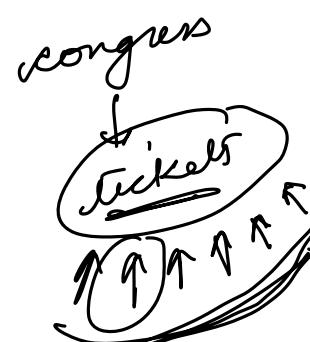
Mohit 90%
Mohit 90%

Definition : It uses a column / combination of columns to uniquely

identify a row.

- In case of super key it can use redundant columns.

candidate: minimal set of columns that can uniquely represent a row.





cd, name, psp, b-^acd

SK

10

X

id , ~~now~~, psp , $b\text{-}\overset{\circ}{\text{id}}$

1

X

~~id / name~~, PSP, b-^aid

1

~~sd, nsp, PSP, k^a~~

X

X

~~cd, name, pp, b^acd~~

✓

2

id, email
email

1

X

4

1

1



Announcement

- What **NEXT?**



10-15 min

1. Assignment / Homework

