

Agenda

- a. Repeated fields ✓
- b. writing efficient queries ✓
- c. Indexing ✓
- d. Need of partitioning
- e. FAANG questions

Nested Fields

(107)

Table Σ Name: Mcfly , Type: Frisbee
 L Petz_and_toys PT

Select Toy.name , Toy.Type , Toy.Toy_name , Toy.Toy_type
 From
 Petz_and_toys PT

Mcfly / Frisbee
 |

(2) Repeated Fields

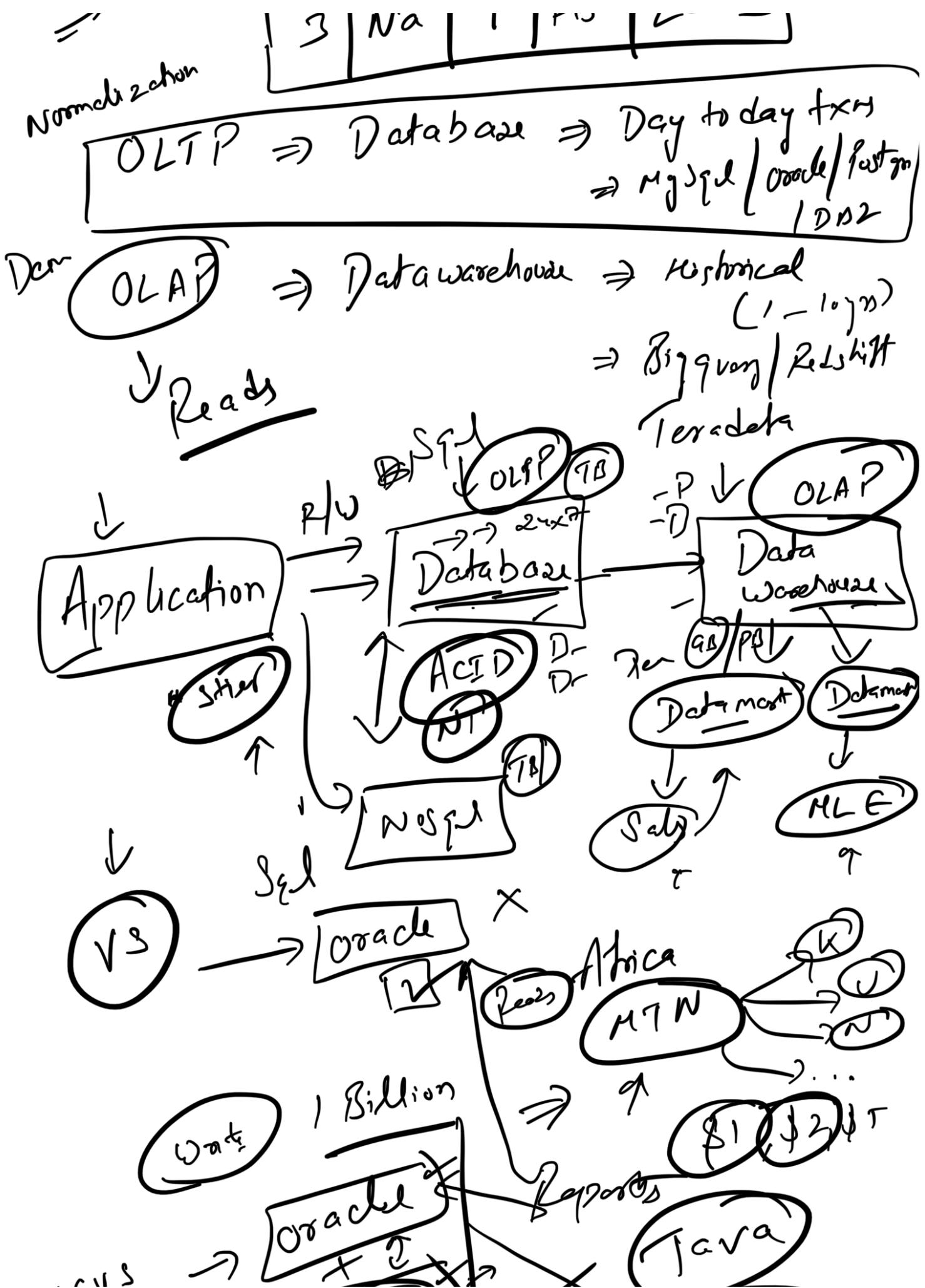
Pets

ID	Name	Age	Animal
1	Moon	9	Dog
2	Ripley	7	Cat
3	Napoleon	1	Fish

Toy-Type

ID	Type	Pet-ID
1	Frisbee	1
2	Bone	1
3	Rope	1
4	Feather	2
5	Ball	2
6	Castle	3

ID	Name	Age	Animal	TOYS
1	Moon	9	Dog	[Frisbee, Bone, Rope]
2	Ripley	7	Cat	[Feather, Ball]



Nuv



UNNEST()

Select
ID, name, Age, toy-type

From

Pets_and_toys, UNNEST(Toys)
as Toy-type

1	Moon	9	Forsbee
1	Moon	9	Bone
1	Moon	9	Rope

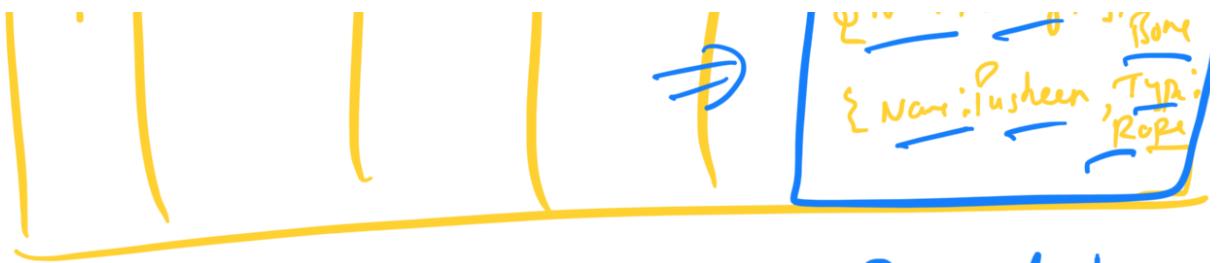
Nested and Repeated Data

1	moon	9	Dog
---	------	---	-----

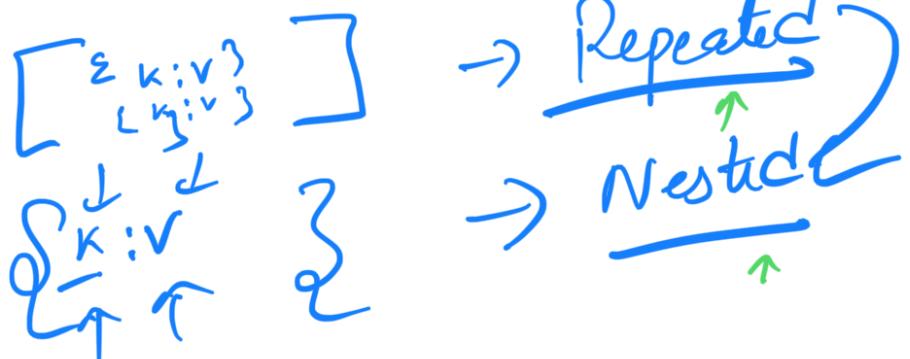
	more-toys	Name	Type	Pet-ID
1	Mitty		Forsbee	1
2	Sally		Bone	1
3	Pytham		Rope	1

DWH

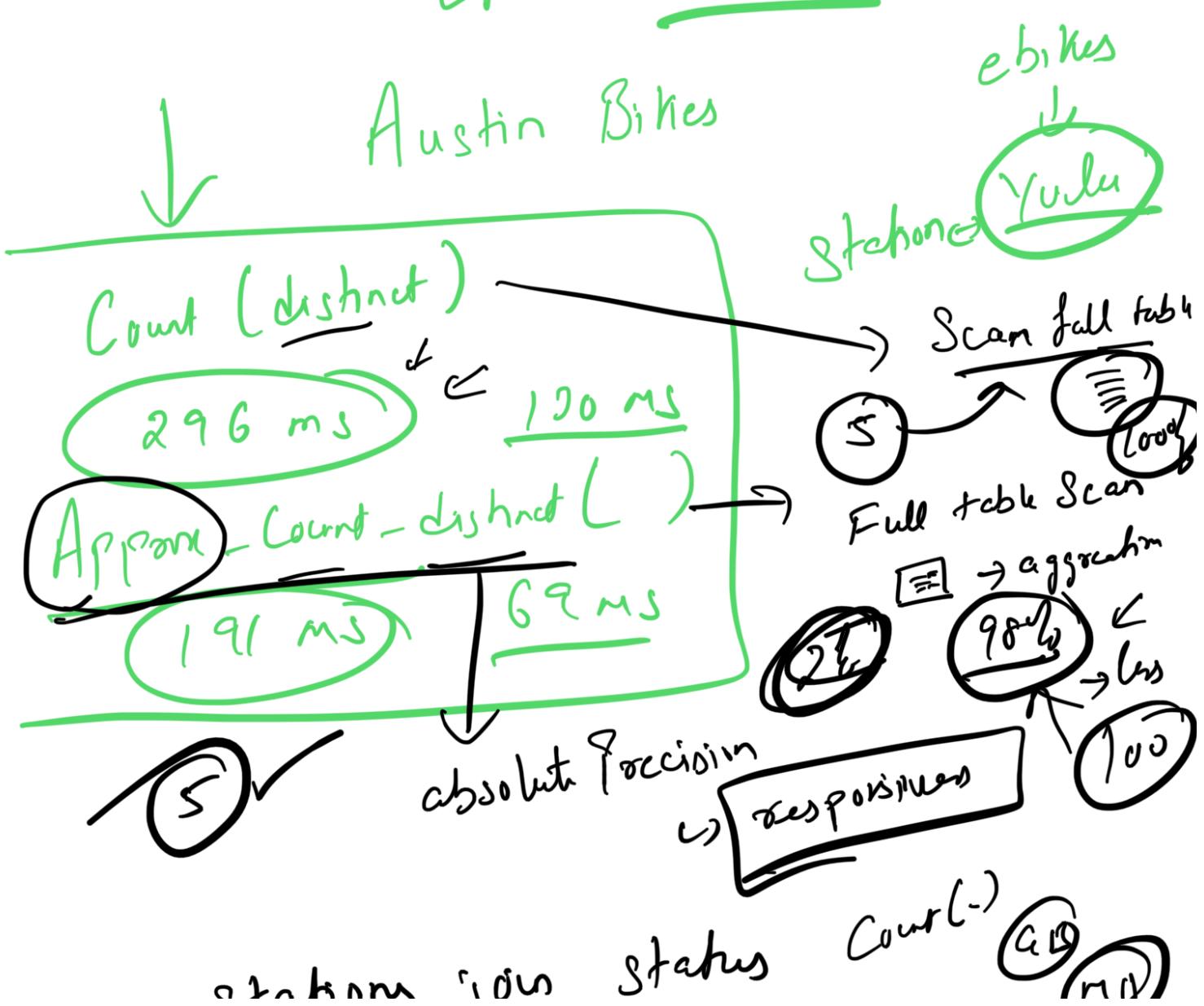
T	I	Moon	9	Dog	[{ Name: "mitty", Type: "Forsbee" }, { Name: "Sally", Type: "Bone" }]
---	---	------	---	-----	-------------------------------------------------------------------------

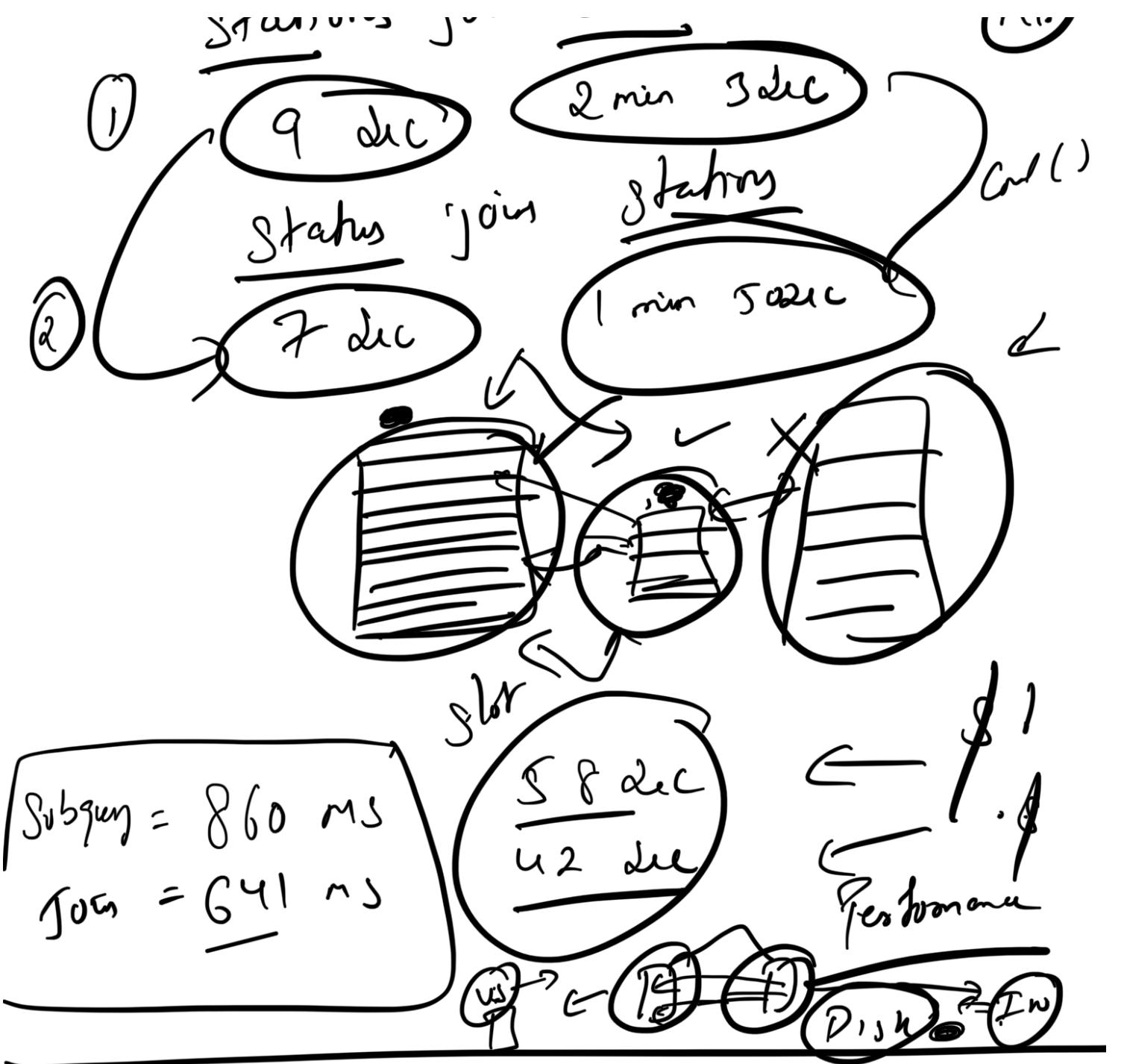


UNNEST



Efficient Queries





SQL Indexes

→ Book



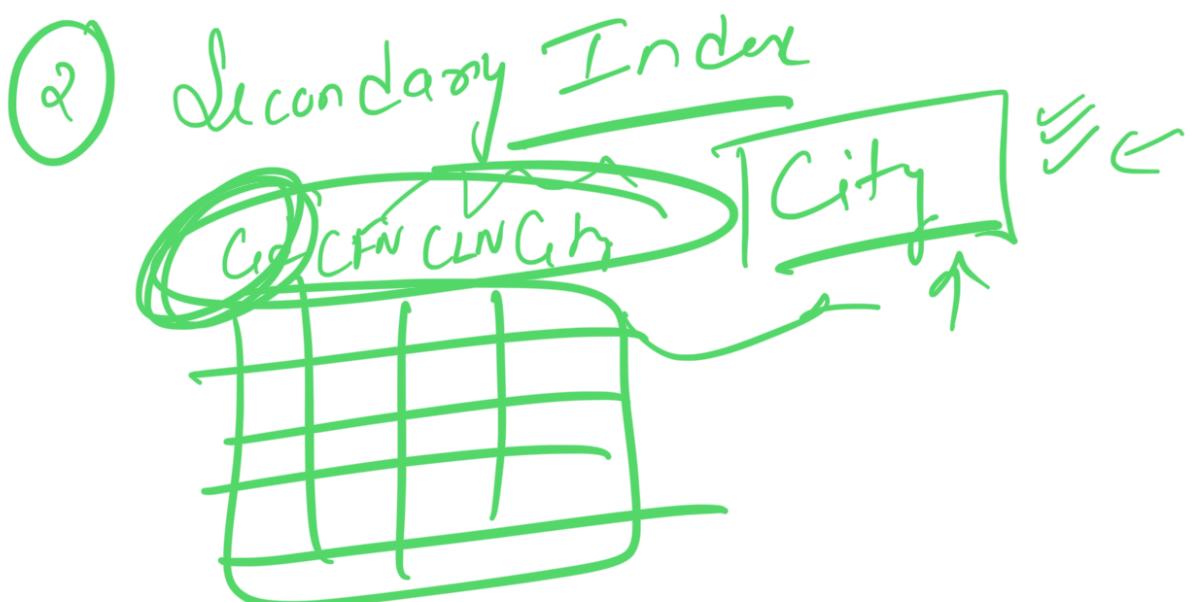
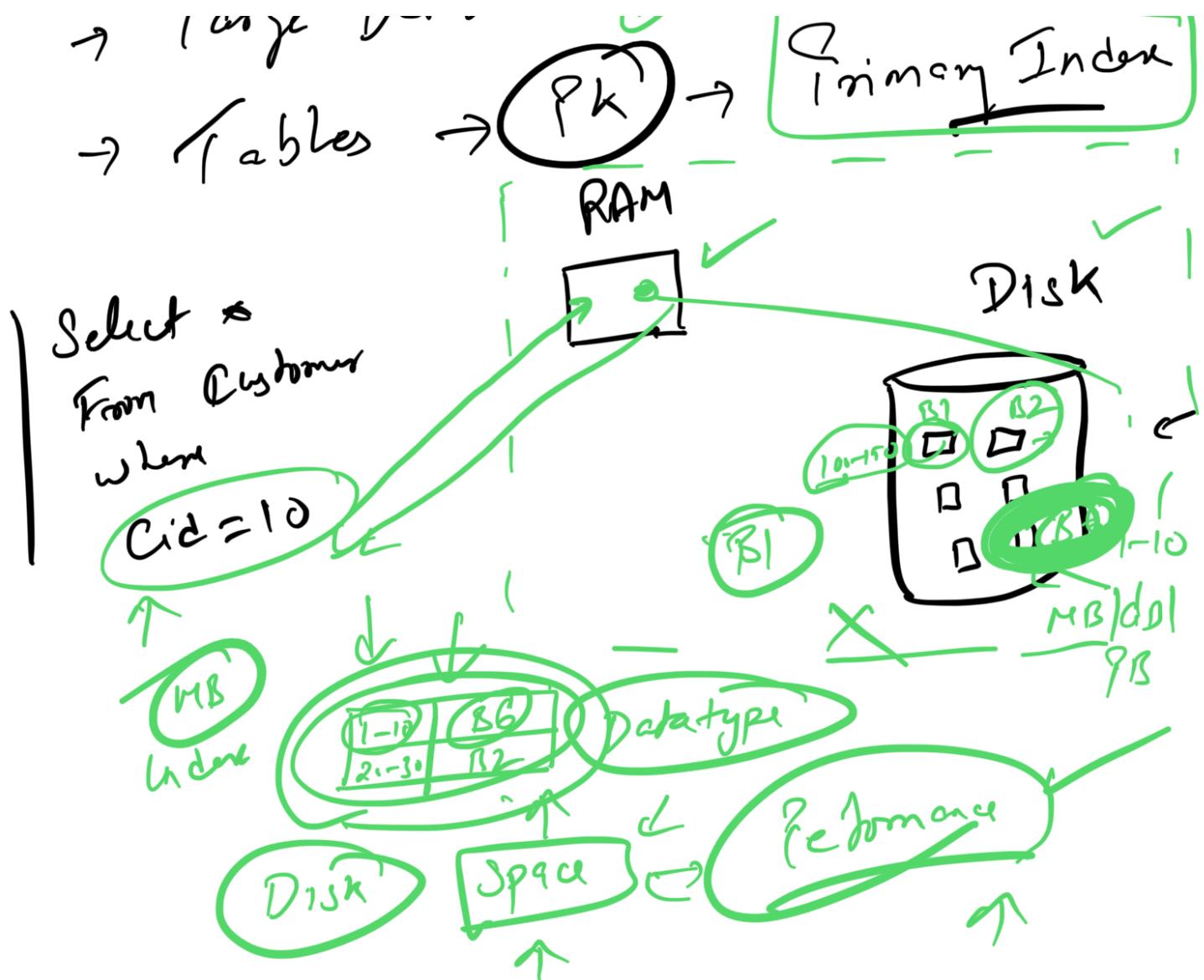
School



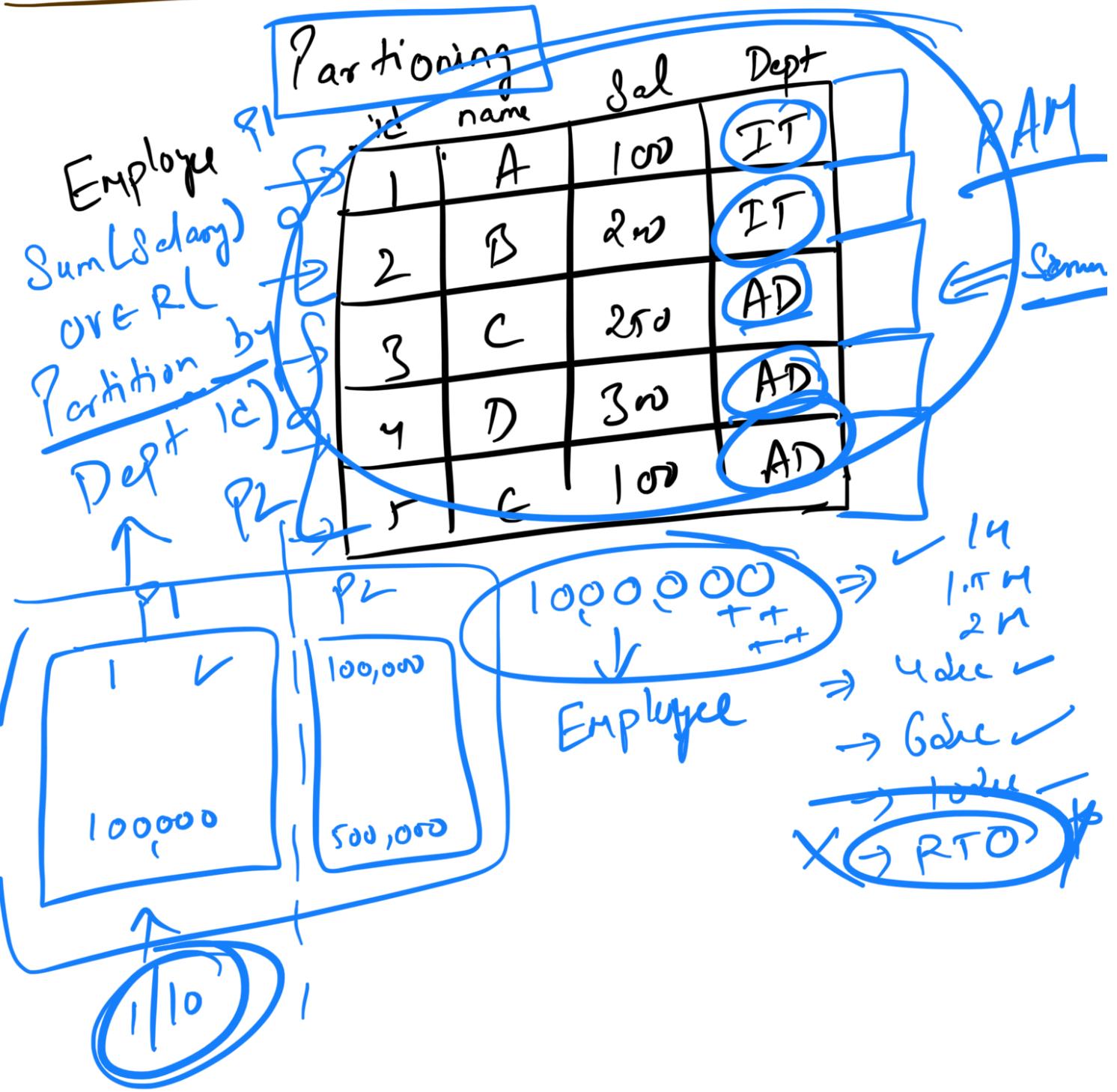
→ lookup tabl quickly retrieve data
from database

large datasets

1/s



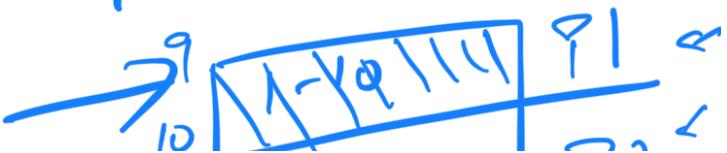
326 ms 98 ms

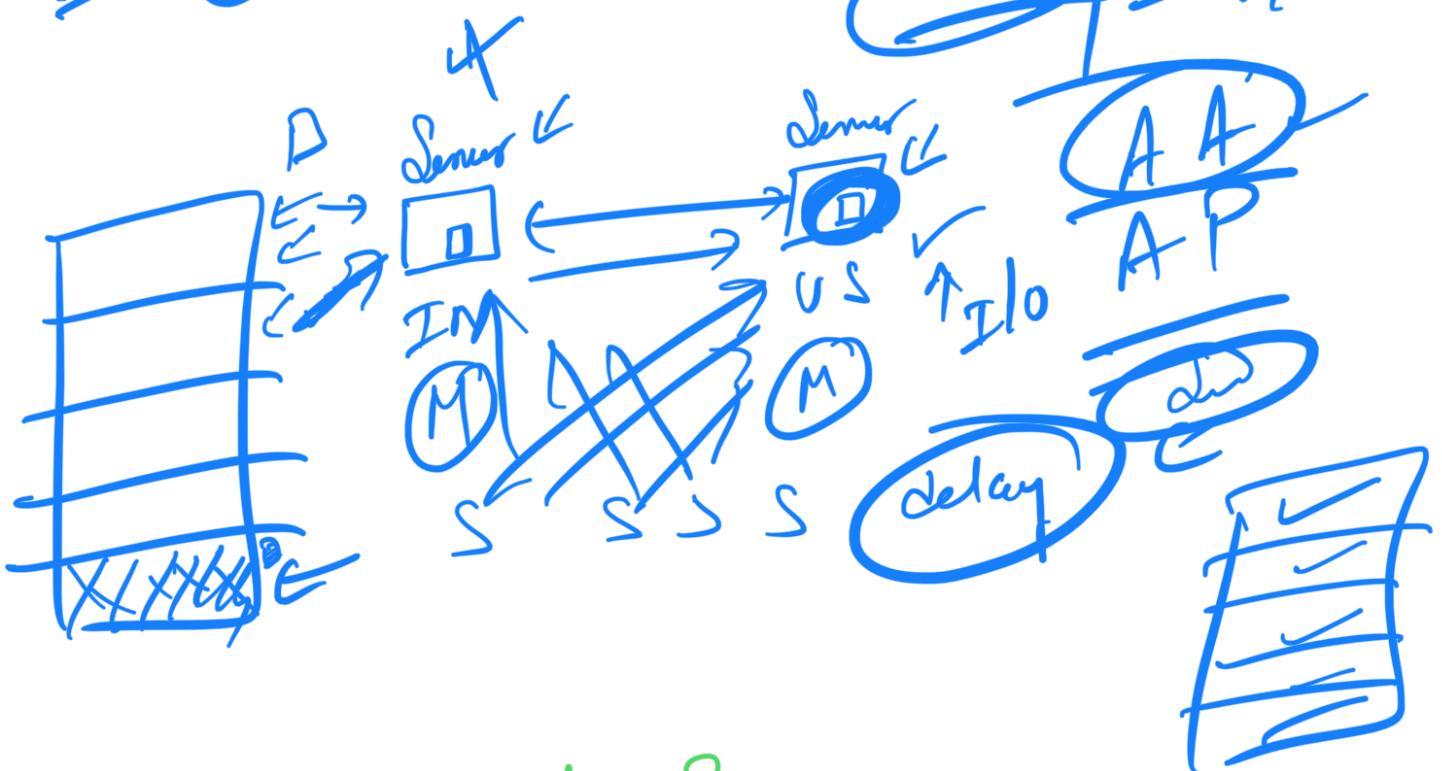
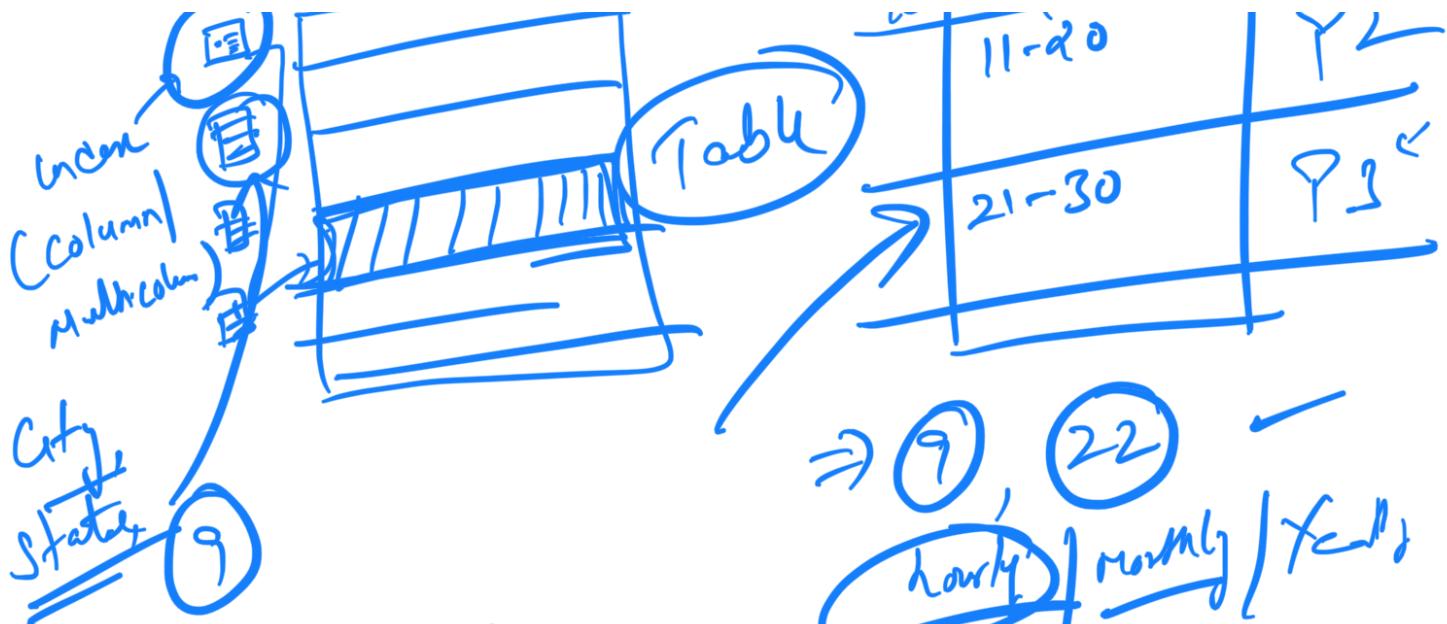


② when to use partitioning

- ① Query Performance
- ② Costs \$

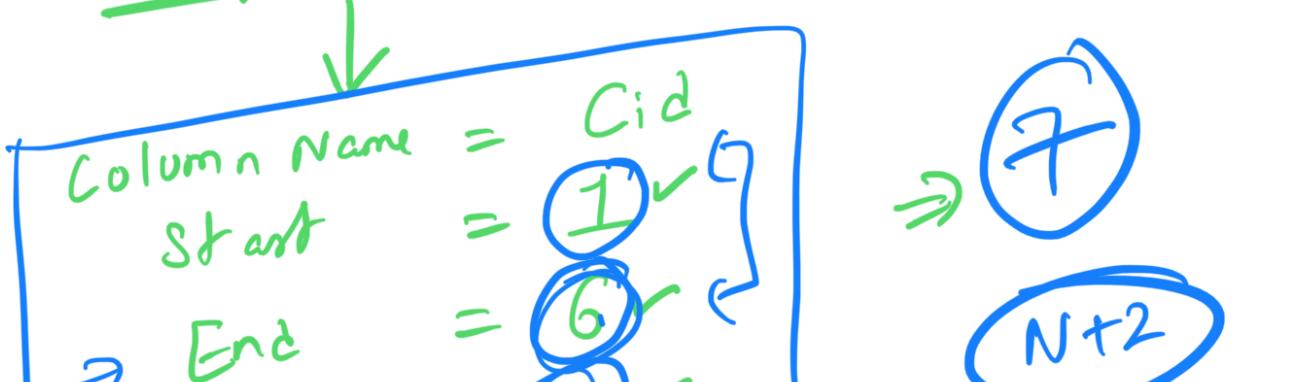
↓ ↓





Types of Partitioning

- ① Integer Range
- ② Time unit
- ③ Ingestion time



Interval = $(2^j)^v$

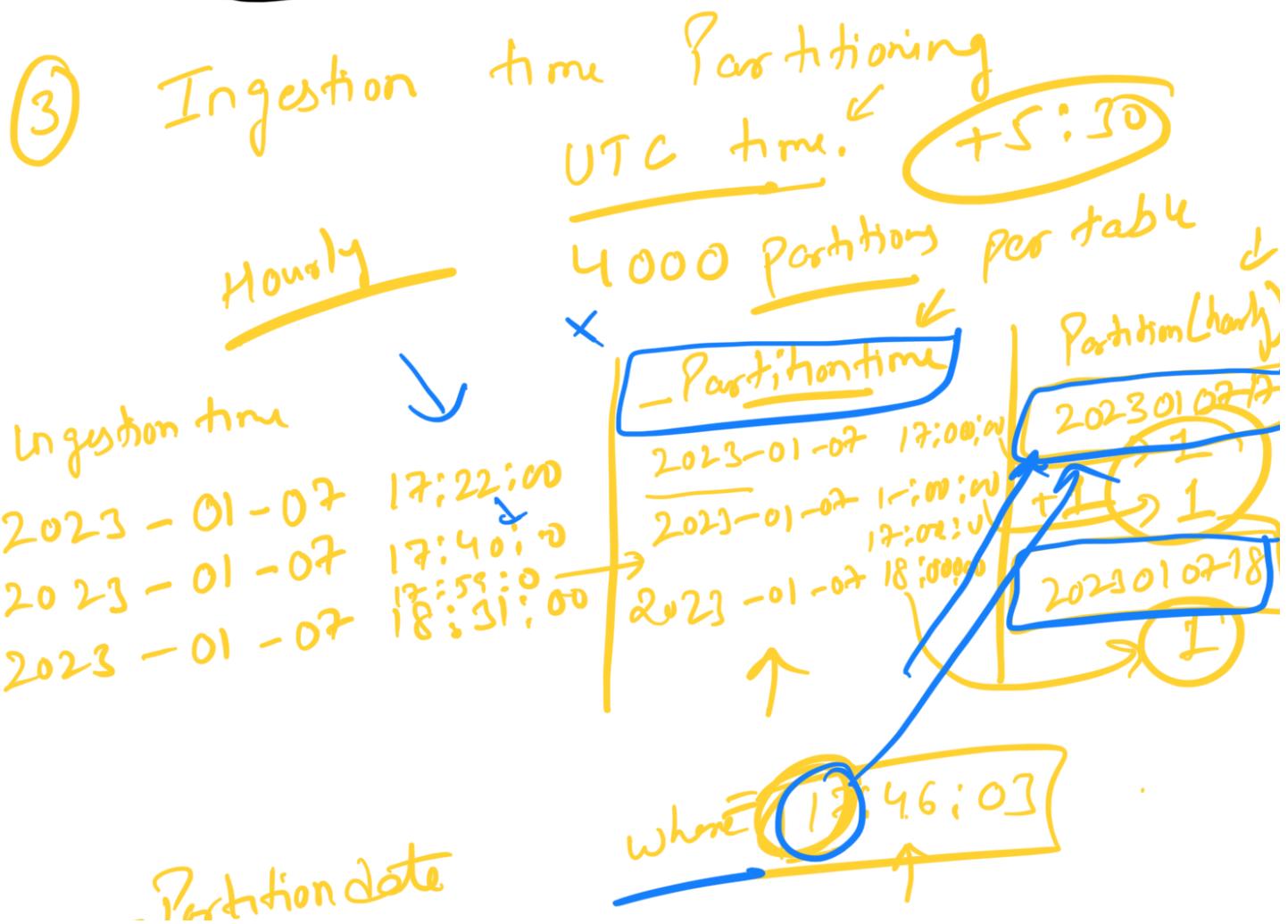
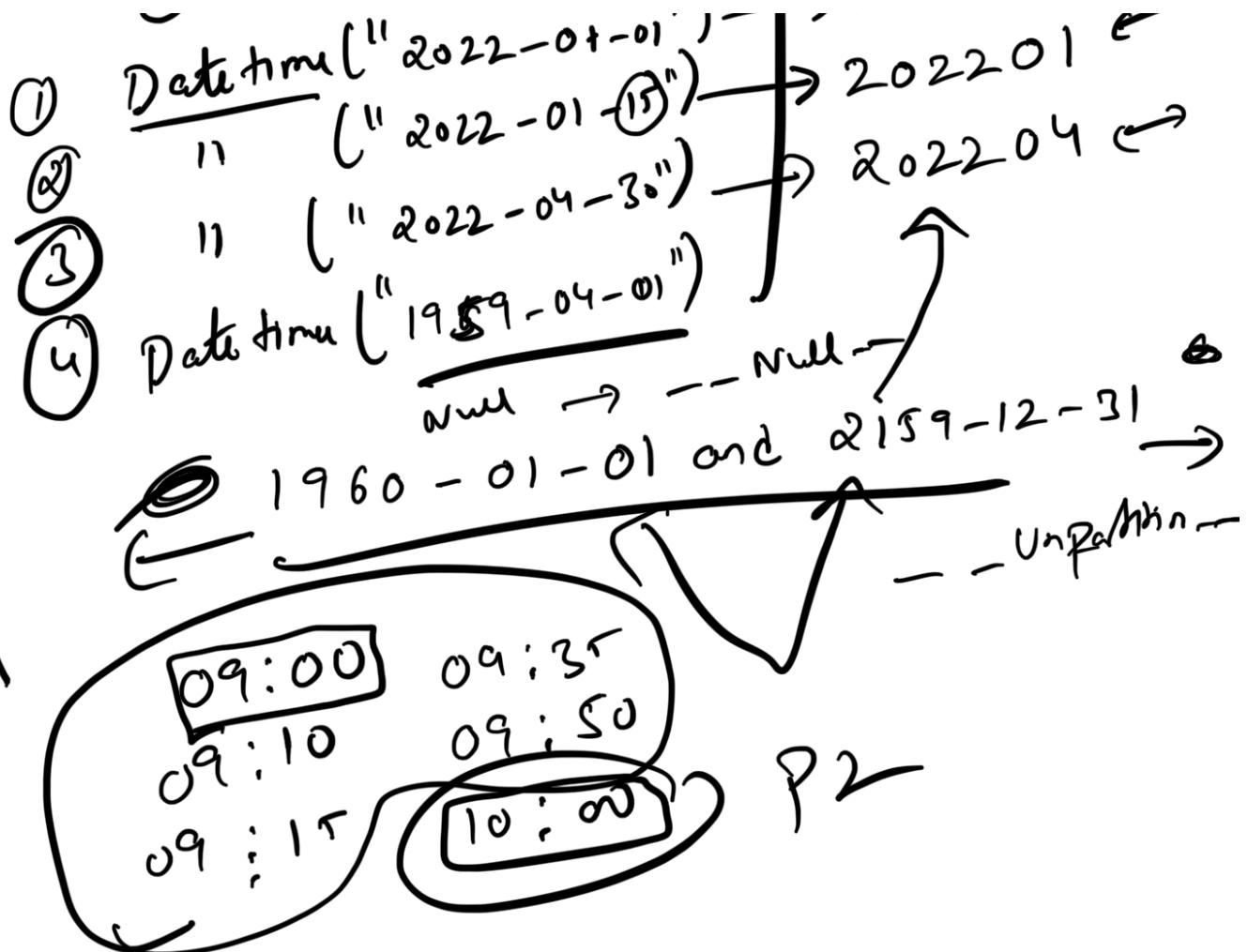
	C_id	CFN	CLN	C Address
P1	1			
P2	2			
P3	3			
	4			
	5			
B	6			
	7	-	-	-
	Null			

Malli -

Unpartitioned

Null

- ② Time unit Column Partitioning
- DATE → Daily / Monthly / yearly
 - TIME STAMP → hours / D / m / Y
 - DATETIME → hourly / D / M / Y
 - UNPARTITIONED
 - Null
- Column value ↗ Partition (Monthly)
- ↑
2022 01 ↘



1

Fb-Comments-Count

User-id (int),

Created-at (datetime)

No-of-Comments (int)

)

Qn. Return the total no. of comments received for each user in the last 30 days. Assume today as 2022-06-15

An. Select User-id, SUM(No-of-Comments)

From Table

Where Created-at Between ("2022-06-15")

AND ("2022 - 06 - 15", Interval 30 day)

Group by User-id;

2

Fb-Comments-Count

↳ User-id int

↳ Created-at datetime

↳ No-of-Comment int

Fb-active-users

↳ User-id int

↳ Name varchar

↳ States varchar

↳ Country varchar

M. Which Countries have rankings in the Entrepreneur

base on the no. of Comments Dec 2021 vs Jan 2022?

Dec 2021

Jan 2022

Avoid gaps between ranks when ranking (conting)

↓ Dec 2021 ✓

1	US	168
2	IN	142
3	AU	106
4	NZ	94
5	DN	30

↓ Jan 2022 ✓

1.	NZ	140
2	AU	120
3	IN	98
4	US	52

X



Dec

1	AU	96
2	USA	75
3	UK	73
4	India	12

Jan

1	USA	123
2	India	96
3	UK	7
4	AU	29

↑

q

USA
India

Google:

① google-gmail-labels

② google-gmail-labels
 ↳ email-id set,
 ↳ Inher varched ✓

id int,
from_user varchar,
to_user varchar ✓
day int

Qn. Find the number of emails received by each user under each built in email label,
→ Promotion → shopping
→ Social

Anst $\rightarrow P = 100, S = 10, S = 20$

An.

Ans
①

Select
e.to_user,
l.labels,
Count(*) As Cnt
From emails e inner join labels l
on e.user_id = l.user_id
Group by e.to_user, l.labels);

Select
 \rightarrow Sum (Case when label = "Promotion" Then Cnt
else 0 END) as promotion_Count
label = "Social")
 \rightarrow
From (~~temp~~) temp

Group by + user
orders by + user;

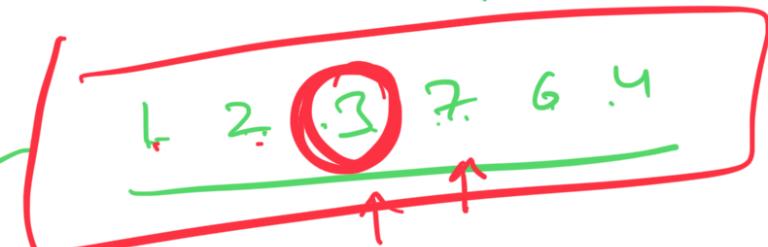
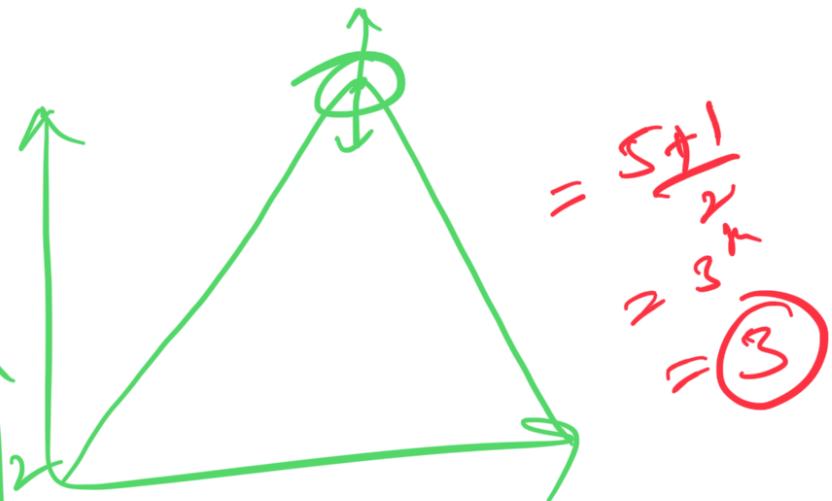
Median =

$$\text{Odd} = \frac{n+1}{2}$$

$$\text{even } \left(\frac{n}{2} \right) + \left(\frac{n+1}{2} \right) / 2$$

$$\Rightarrow \frac{s_m + s_{m+1}}{2}$$

$$s - s_{\text{order}} \neq 5$$



ID	Company	Salary
2	S	4212
6	S	4142
1	S	2053
3	S	1678
7	S	1643
4	S	1413
9	S	1384
5	S	568
8	S	526

$$\frac{9+1}{2} \Rightarrow 5^{\text{th}}$$

$$\frac{n}{2}, \frac{n+1}{2}$$

