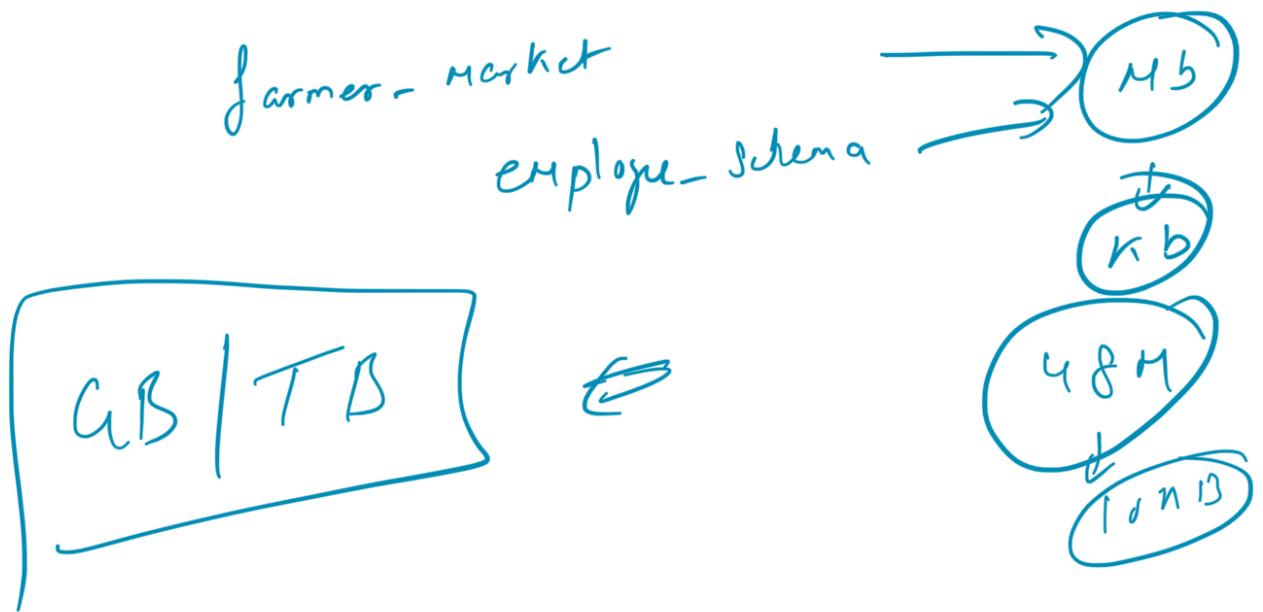


Agenda : Optimisations in SQL



(a) Crypto Dataset
 ↳ Currency → Coins

↳ Investors
 ↳ Minors



(i) Count (*), Count (1), Count (col.name)
 ↳ nulls ↳ nulls ↳ not nulls

⇒ Same time

→ No difference in performance

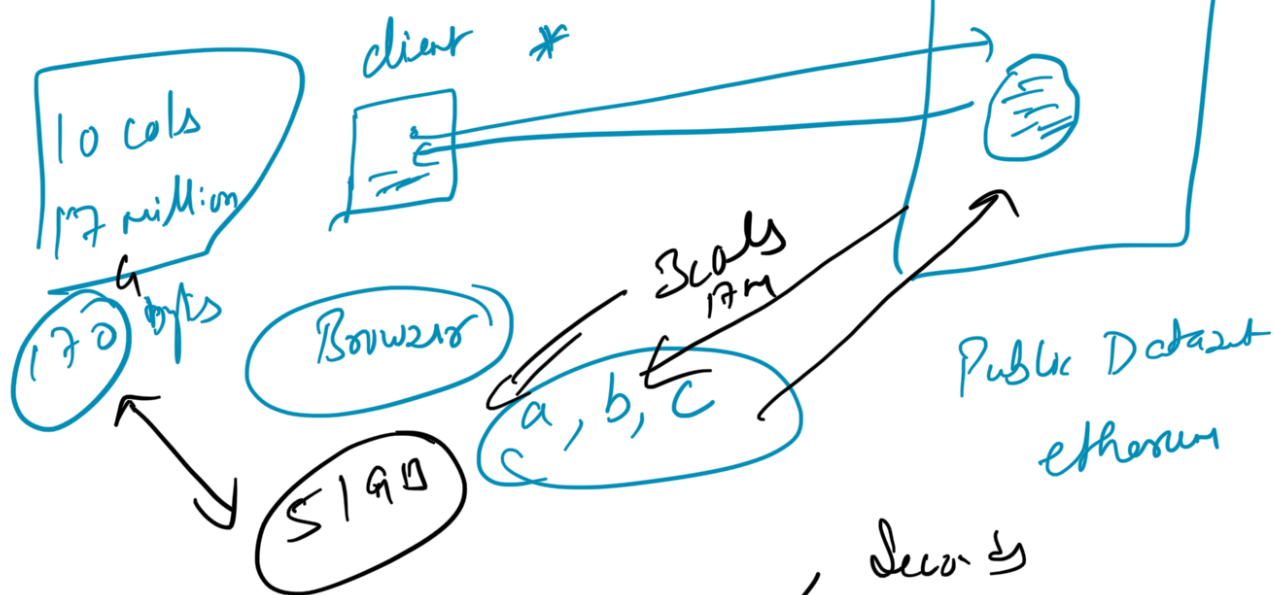
Limited number of columns

(2)

*

)

Server (Google)



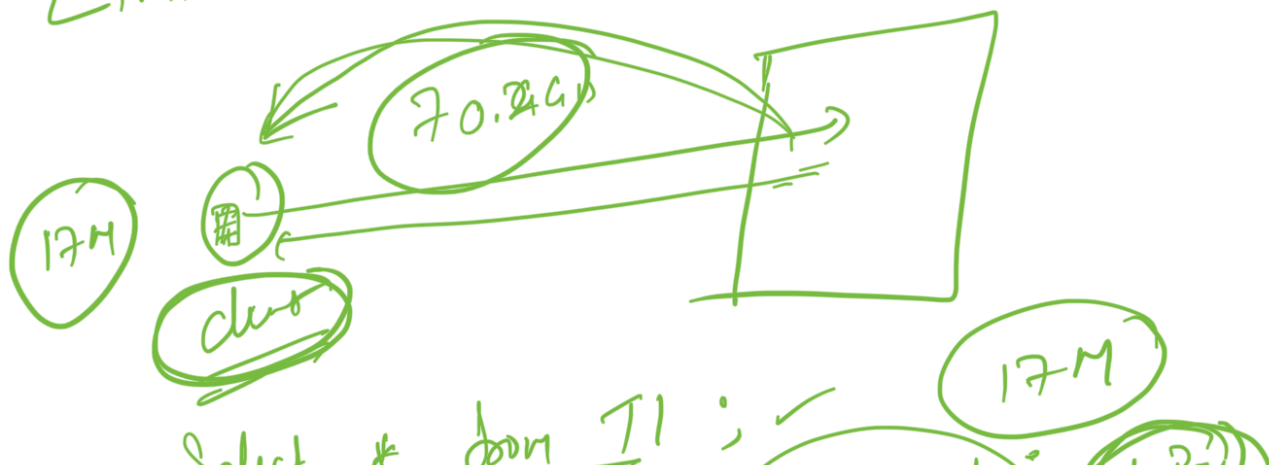
↓ (1) → Time elapsed = how much query has actually run?

↓ (2) Slot Consumed = RAM + CPU

↓ (3) Bytes spilled = Slot

B/kb

(3) Limit is a trap



→ Select * from T1 (limit 1) (1 row)

✓ → Select * from T1 (17 rows)

2.6 kb
32 sec

4) Exists () ✓

Select Exist (2)

Join employee ;

True / false

Hand-drawn diagram of a 2D array with 3 rows and 3 columns. The first column is labeled 'id' in a circle. The first row is labeled '1', the second row '2', and the third row '3'. The first cell of the second row contains the value '2' in a circle. The first cell of the third row contains the value '2' in a circle. A green line is drawn above the array.

⇒ Use exists() instead of count()

⑤ Use Approximate Agg function

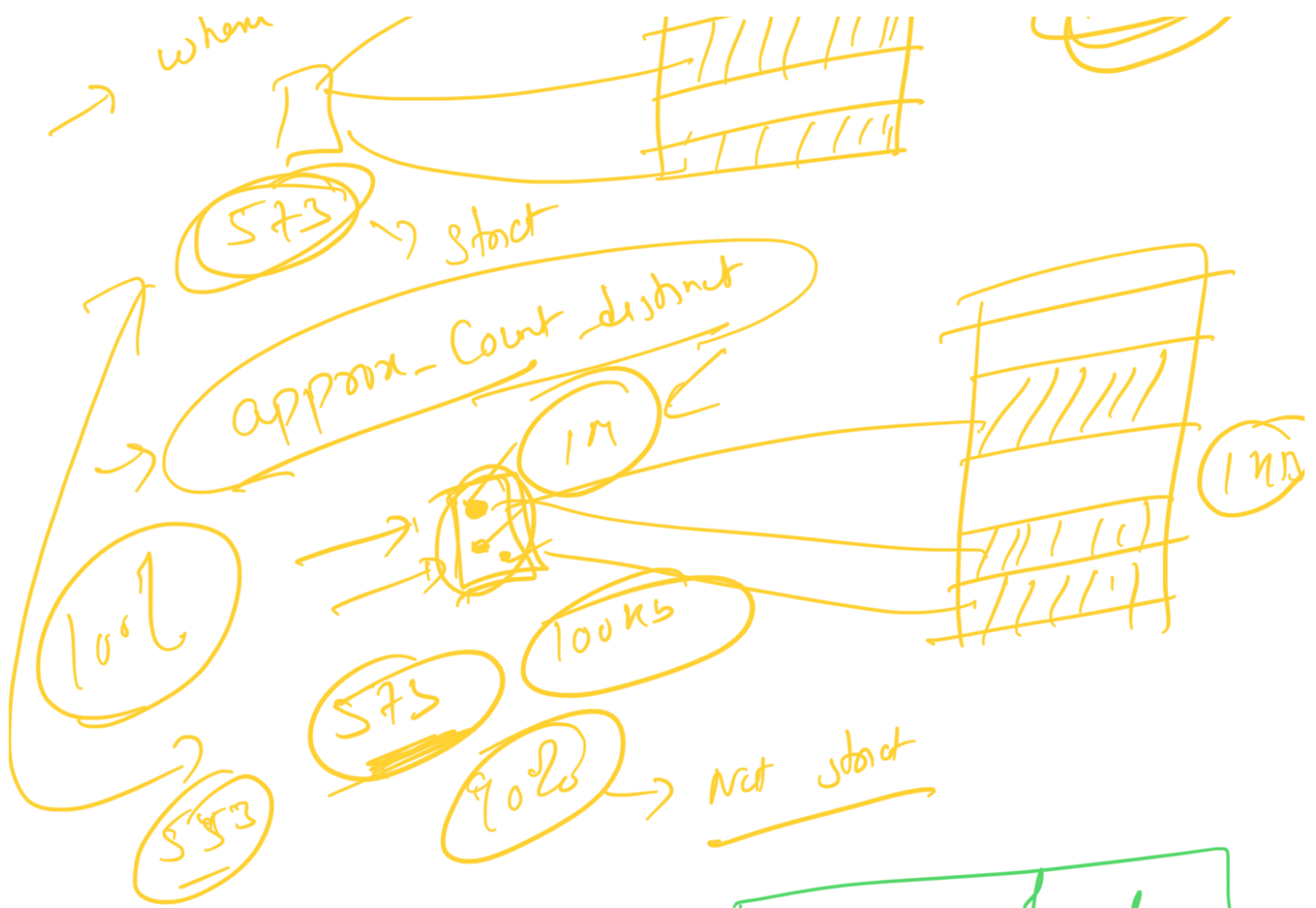
5) Use API

- Appon - Count - distinct()
- Appon - top - Count()
- Appon - top - Sum()
- ...

95%
100%

(a) $\frac{\text{Count (min)}}{a^2}$ and 'b'

17M



⑥ Self join vs Window function

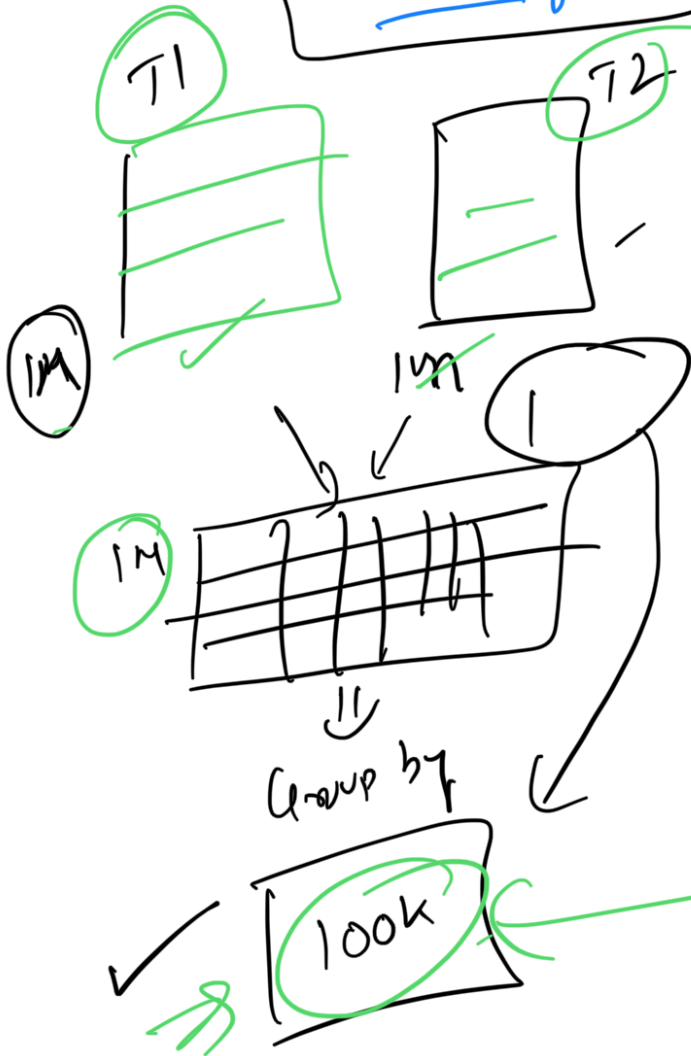
↓
Inefficient

⑦ Trim your data early and often

1				
2				
3				
4				
⋮				

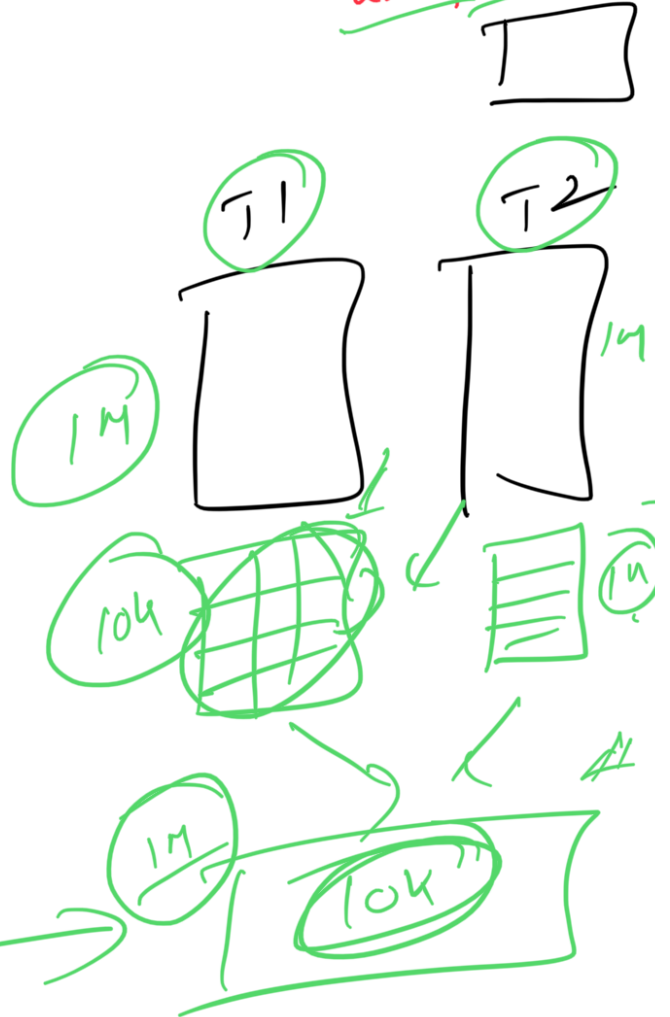
① with demo
 ① select a, b from T1)
 ② demo-1
 (select a, b from T2)
 Select a, b, c
 from demo d
 left join
 demo-1

Group by a, b



② with demo
 ① select a, b, c from T1

② demo-1
 (select a, count(b) group by a)
 select a, b, c
 from demo d
 left join demo-1



① where sequence matters

①

with

①⑥

use mem() instead of Rank()

Doubt clearing session

\$ 70,000 → 515

$$\begin{array}{c} \downarrow \downarrow \\ 20 + 16 \Rightarrow 36 \\ \downarrow \downarrow \\ 34000 \end{array}$$

$$\Rightarrow 10000, 15000$$

①	1	J	10000
②	9	J	15000
③	2	S	20000
	11	S	16000
	13	S	50000 X
④	4	J	40000

		10000	10000
		15000	25000
		40000	65000

1	16	16000
2	36	20000
	86	50000 X

① and (70 - 50) = 20
70 <

$$\text{Select } \left(\frac{70000}{36000} - \frac{36000}{36000} \right)$$

- Max(rnk) from cte
when expen = S and
rnk < 70000

Totalk Most frequently bought

1	Phone	12000
2	Tshirt	4000
3	Shoes	5000
4	Perfumes	3500