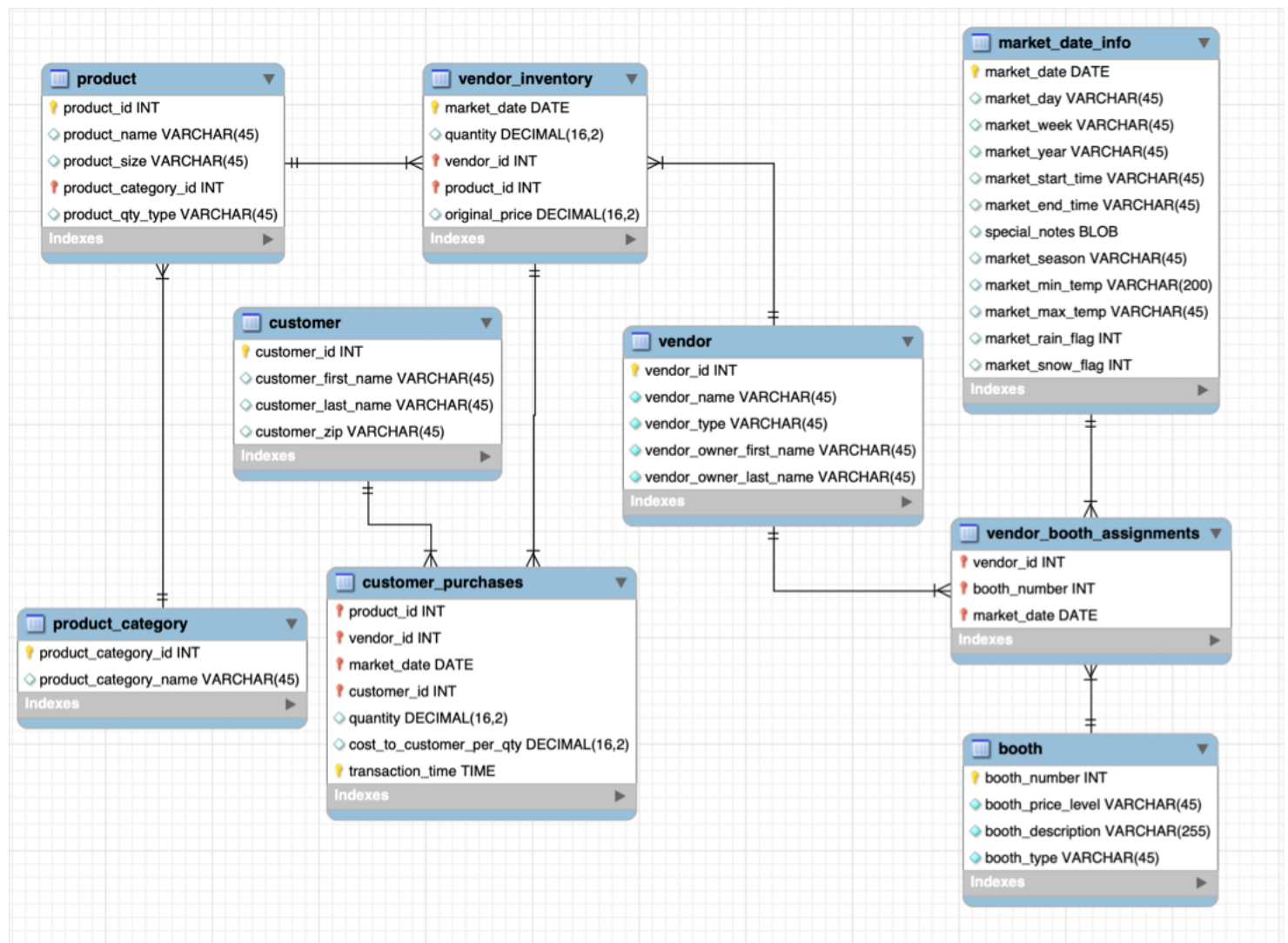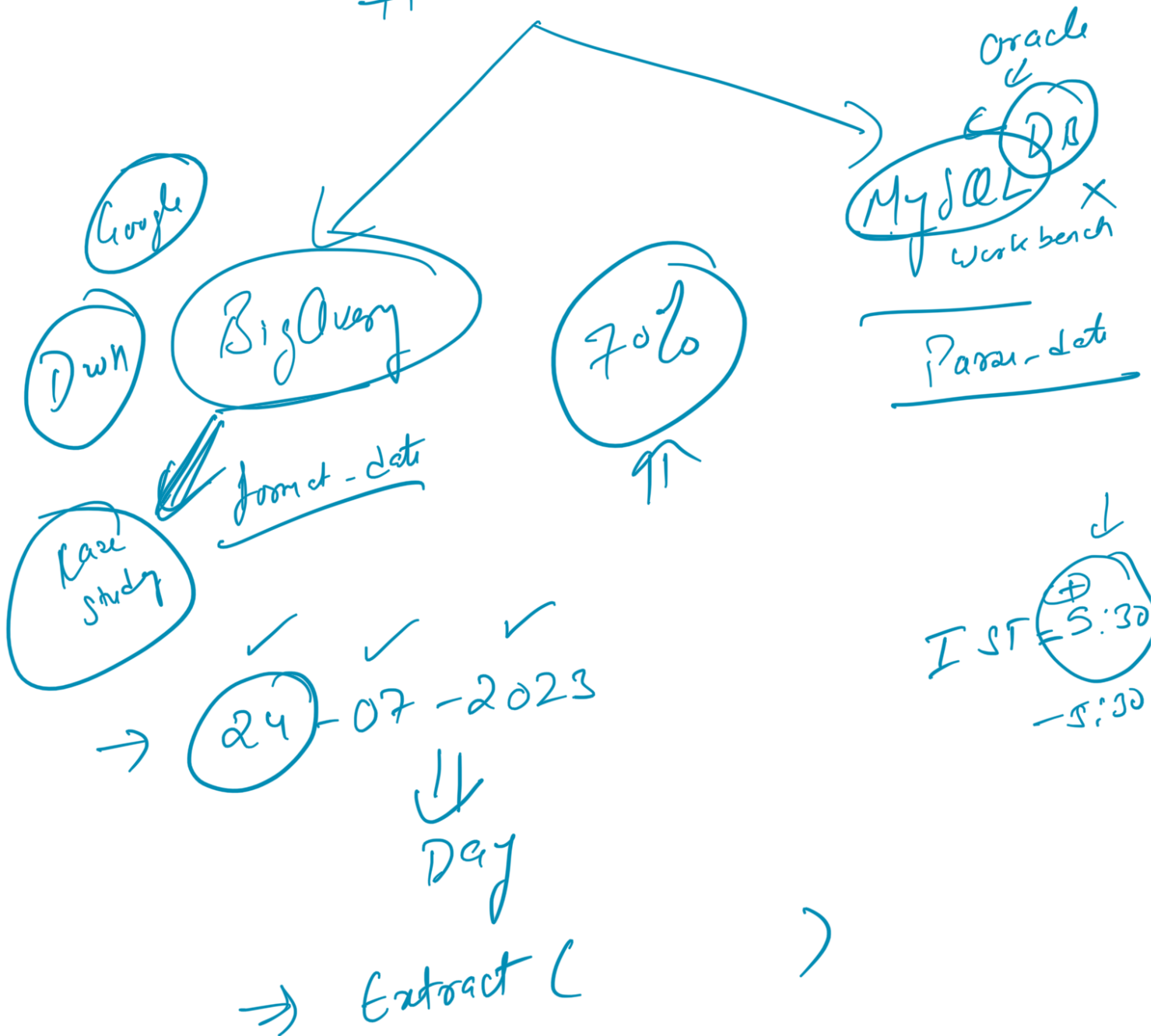# Agenda

a. Date Functions

b. Time Functions

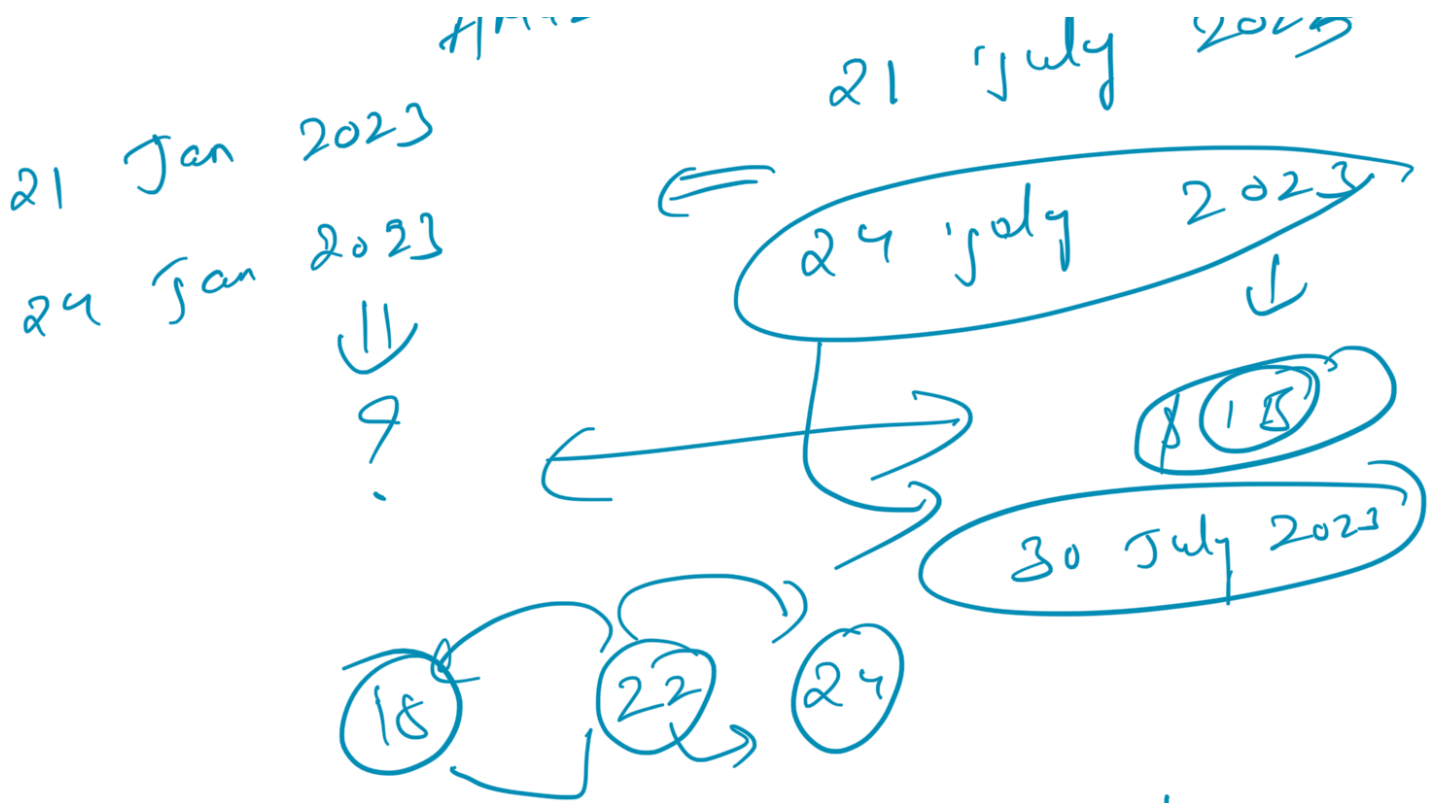$\Rightarrow$ Group by | agg | having) Joins) window functions

Date & Time functions (time zones)

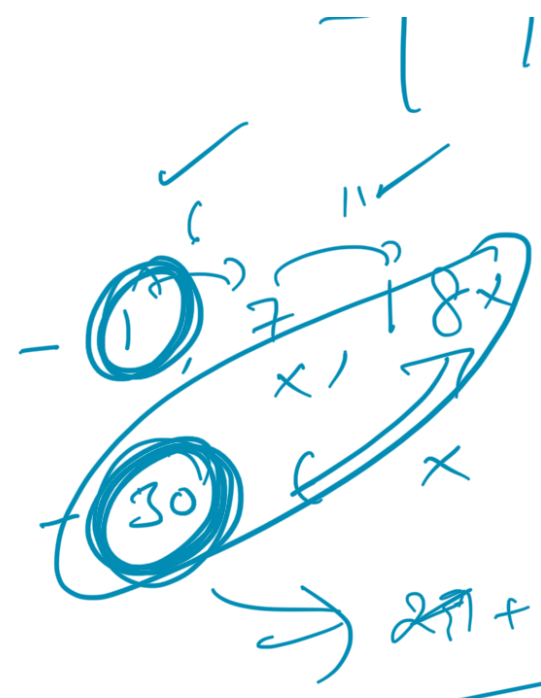BigQuery

Google

Dwn

Case study

→ format_date

7olo ↑

Oracle

MySQL DB ✗
work bench

Parse_date

IST = 5:30 ⊕
-5:30

✓ ✓ ✓
→ (24)-07-2023
⇓
Day

⇒ Extract ( )

Amazon

21 Jan 2023

24 Jan 2023
$\underset{?}{||}$

21 July 2023
24 july 2023
↓
∅ 18

30 July 2023

18 → 22 → 24

① Date_ADD(), Date_Sub
↓ ↓ ↓
⇒ Date_add (col, Interval 30 minute)
↳ 1 = Col datetime
timestamp

↳
⇒ Date_Sub (Col, Interval 30 minute)

⎰ Show — 24 / 07 / 2023   1 kg T → v1
⎱   1 kg O
      24 —   1 kg T → v2
      3 | 2 | 1   T | O

S

Jan 2019 —

Mar 2019 —

$12+28$
$+30$
70

lag(3)

$\rightarrow \dfrac{27+28+30}{} \quad = 3 \text{ mth}$

lag(col) $\twoheadrightarrow$ (1)

lag(col,2) $\Rightarrow$?

Doubt clearing session

Account Balance :-

Approach :- ① each ✓ → Join ✗
② balance → O
③ aid , day

1	07	D	2000 ← 2000
1	09	W	1000 ← 1000
1	11	D	3000 ← 4000
			7000

1-D = Sor
1-W = 100

1		D	7000		
2	07	D			
2	12	W	7000		O

~~Group by~~   Partition by ✓

Select aid, day,
Sum (Case when type = 'Deposit' then amount
           else - (amount)
                              balance)?

from trns ;

↓

2000 D$
-1000 w   ⇒ Sum ( ) on every trn
3000 D

                ⇒ Sum ( ) over (Partition by aid
                                order by day)

2000
3000
6000

2000
1000
4000

(2) Salaries After rouse

(1) → Round (, 0) ✓

(2)

Case

(a) → 0% $\boxed{\text{max (salary)}}$ of any employee
< $1000

(b) → $\boxed{24\%}$ max (salary) between 1000 & 10000

(c) → 49% max (salary) > $10000

(3) $\dfrac{Salary - Salary \times (tx/100)}{}$

$\left\{ \dfrac{Salary - Salary \times .24}{p. 49} \right.$

(4) order by

Ans

Select cid, eid, emp_name
Round (Case
when $\boxed{\text{max (salary)}}$ Over (PB cid) < $1000 then Salary
when (Max (salary) ov 11) between 1000 at 1000
then , 0) as $\boxed{Salary}$

I am feb4

order by

Orders with max qty above average

→ Imbalanced = $\max(qty) > avg(qty\ of\ every\ order)$

→ $Avg(qty) = total\ qty\ /\ no.\ of\ diff$

→ $Max(qty) = $ highest qty of any single product

→ order, 2