DBMS - QUERIES - II

1. Joins
   - Inner Join
   - Outer Joins
   - Queries

2. Aggregation
   - functions
   - Group By

(3) Leet code SQL questions

- 8 questions
- 3 queries
- 5 HV

Sub-queries

Index

Query opt.

window functio.

Joins

why?

① → when we want info. across tables

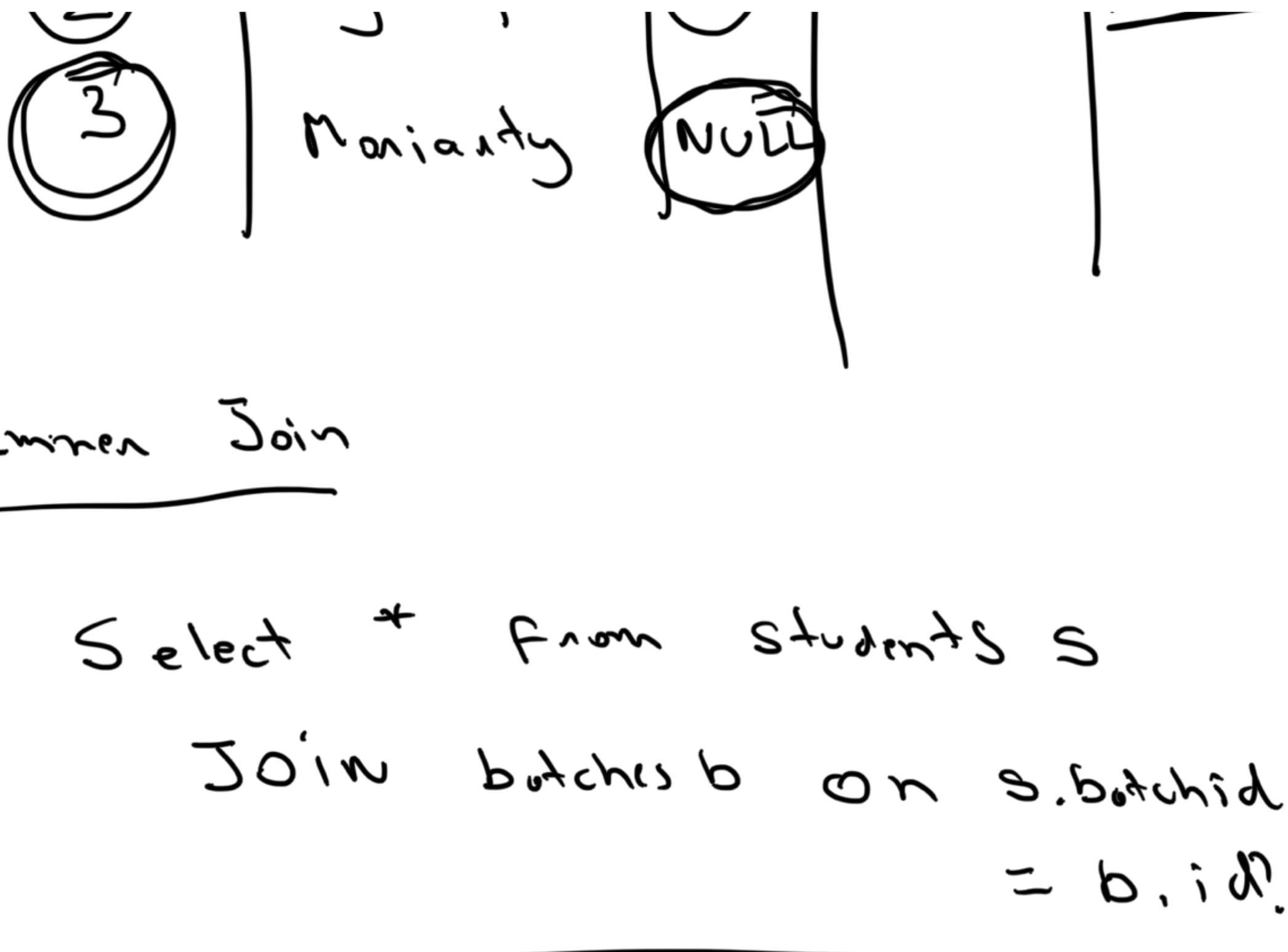
Inner Join

Students

SID	First Name
1	John
2	Mycroft

Batches

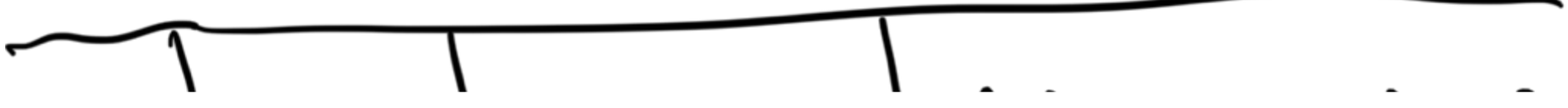
BID	Name
1	Sherlock
2	Crime



Inner Join

```
Select * from students s
    Join batches b on s.batchid
        = b.id;
```

Get all students with batches



SID	BID	NAME	BATCH NAME
1	1	John	Sherlock
2	1	Mycroft	Sherlock

INNER JOIN

```
For s in students
    For b in batches
        if s.batch_id == b.id
```

ADD ROW

if not batch id :

Student batches ADD

Students with both ids

Left outer Join

(1) Students that have batchid

(2) Students that donot have batchid

SID
1
2

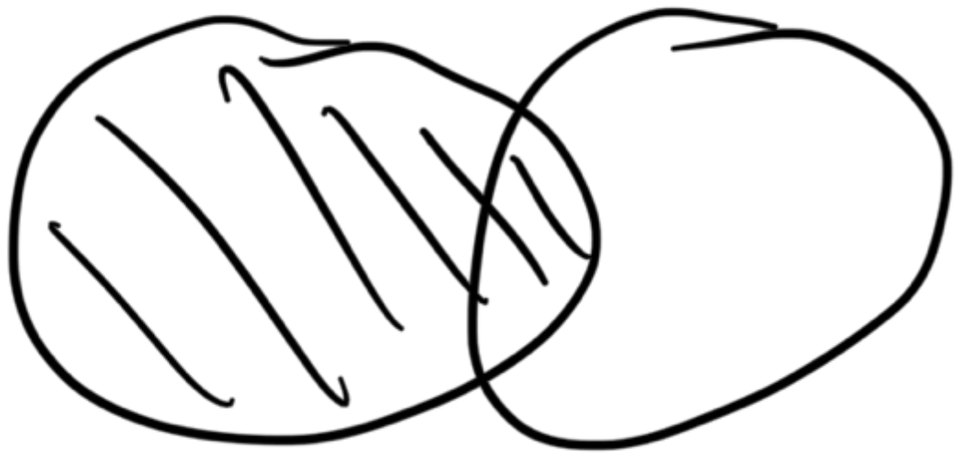
Name
John
Mycroft

BID
1
1

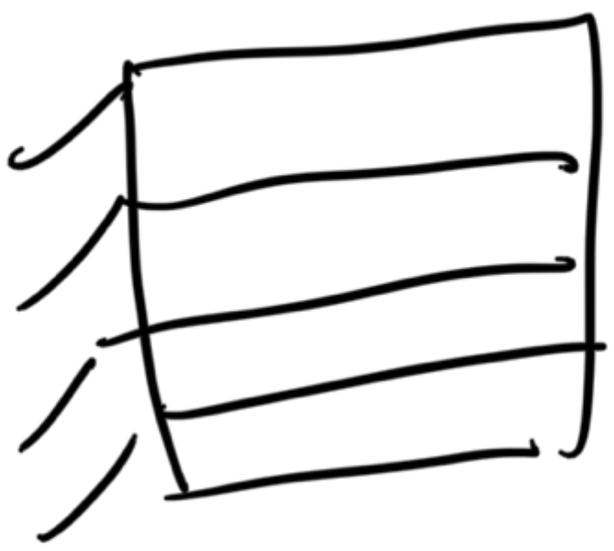
NAME
Sherlock
Sherlock
NULL

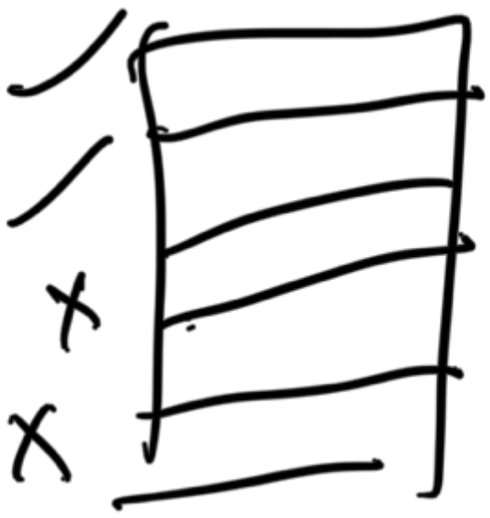
3

Monicity



Left +





① Inner Join

② Outer Join

- Left Join
- Right Join

Students - batches

1	John	1	Sherlock
			Sherlock

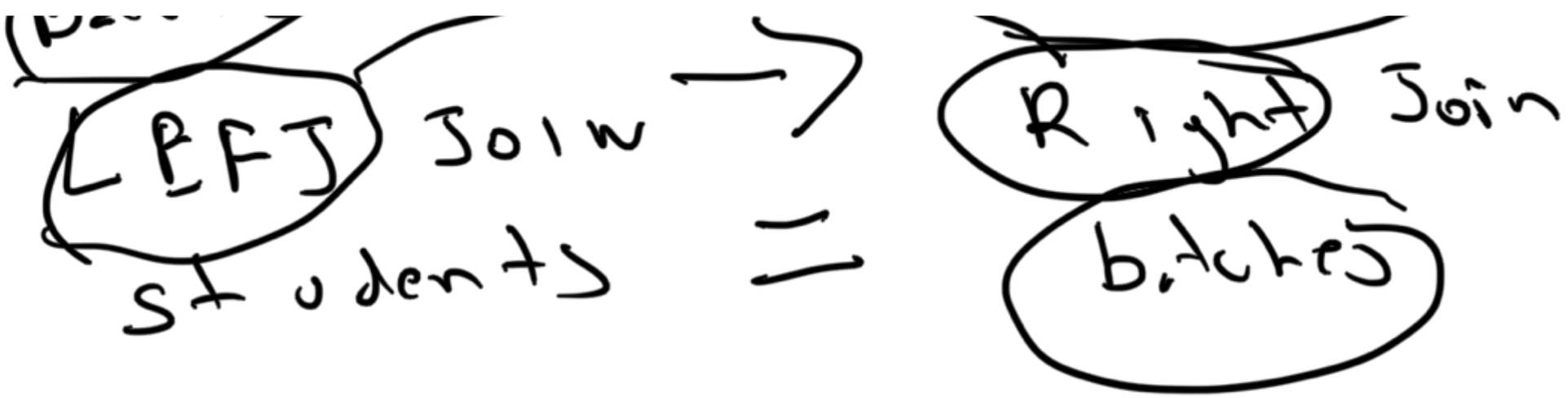
2	My [illegible]	[illegible]	
NULL	NULL	3	Crime

1. Get all matching rows
2. If Left, get all rows from left table

If Right, get all rows from right table

Matches

Students



Inner — only matching

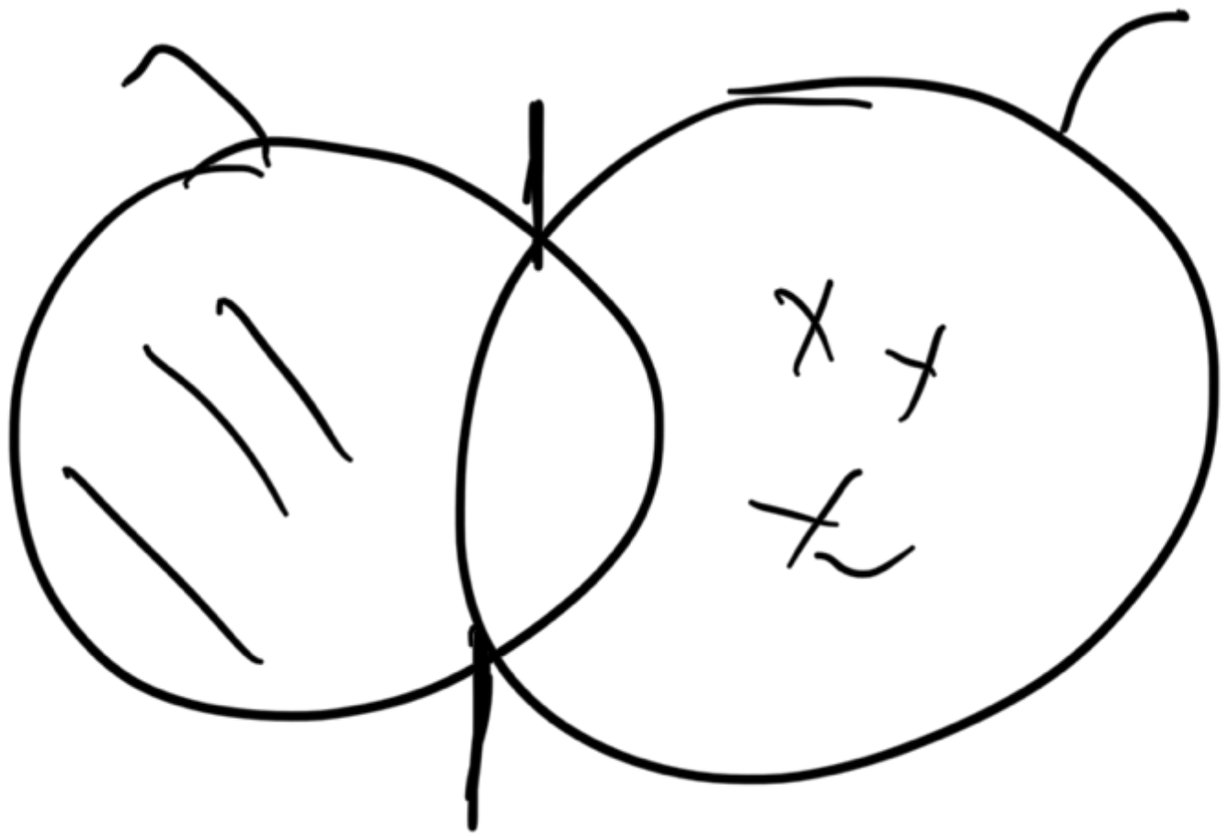
Left outer — ALL in left + matching in right

Right outer — ALL IN right
+ matching in left

FULL OUTER Join — Everything

SID	NAME
1	John
2	Myroft
3	Moriarty
NULL	NULL

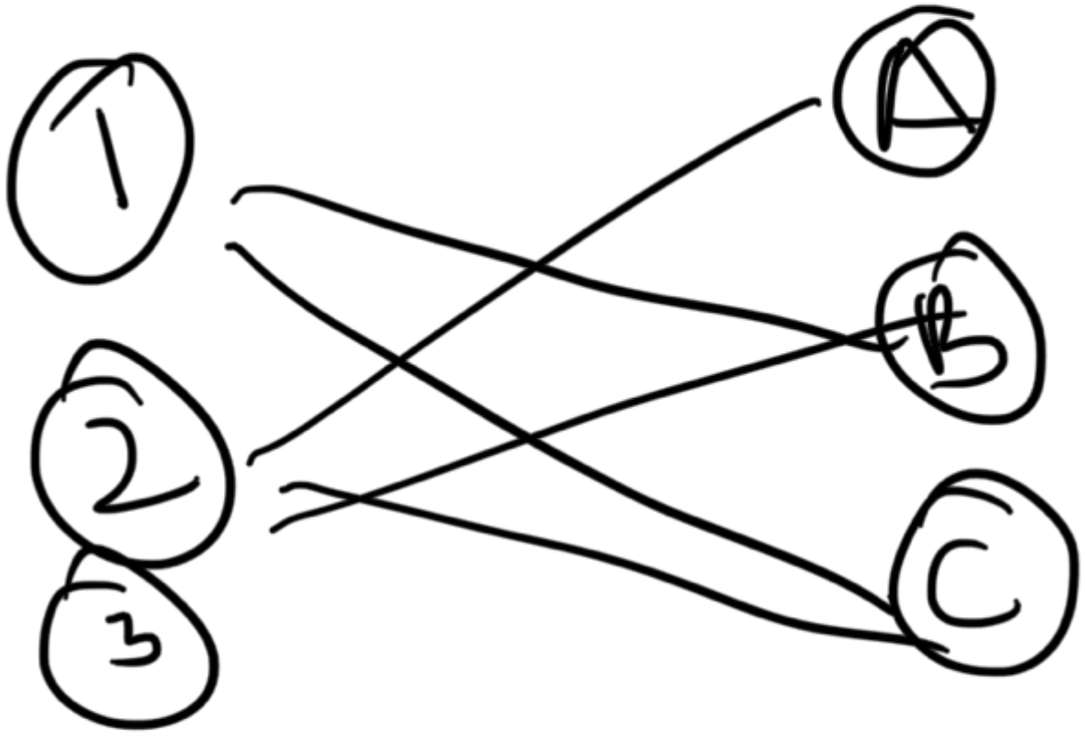
BID	NAME
1	Sherlock
1	Sherlock
NULL	NULL
2	Crime



INNER

CROSS Join

→ Cartesian product



1 A
1 B
1 C

2 A
2 B
2 C
:

Cross join

m , n ⟹ m * n

LIKE ← LIKE

LIKE

Joins

Table + Table → New set of values (merged table)

Intermediate view

All students in batch 1

VIEW - Select * From -- where batch id=1

View

Joins □ + □ = View

Memory + Disk

↓

Volatile

Joins | Left join with multiple table

→ View → Select * from students

100

Materialised views

S

SID	BID
1	1
2	1
3	NULL X

B

BID	IID
1	1
2	2

Inner Join

SID	BID	IID
1	1	1
2	1	1

3 | NULL | NULL

INNER Join

ID	Name
1	A
2	B

1	1	1	A
2	1	1	A

3

1st Join ⇒ Left

Aggregation

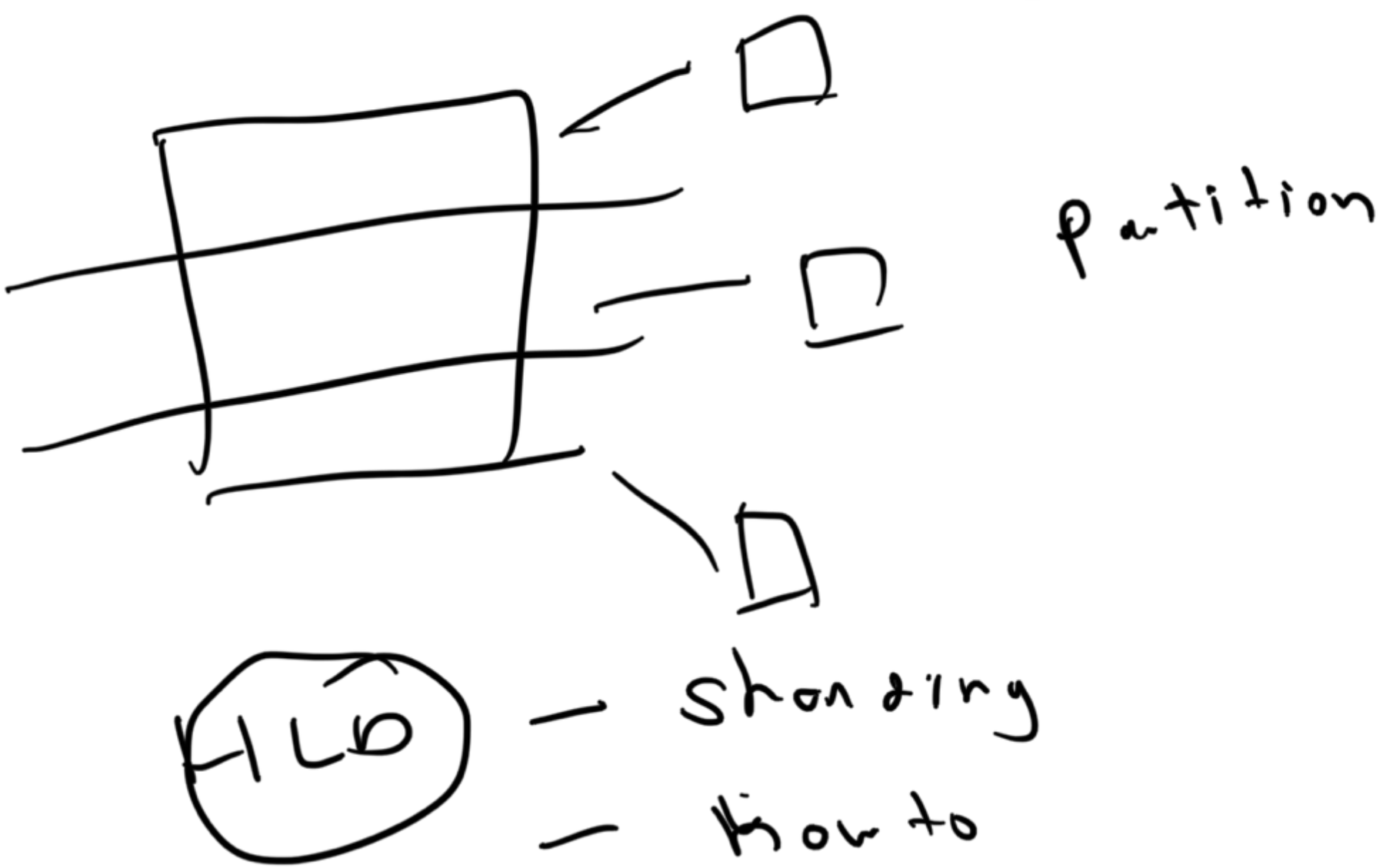
15:52 → 5:57

10:22 10:28

EXPLAIN — Plan → Performance

A lot of nouns

→ scale — vertical
— horizontal



HLD — standing
— how to

- when to
- algorithms

→ slow queries. 'g

Memory model → working set

D

FCFS
LRU

Aggregation → Summarising

SZD

IO

1	180	
2	260	
3	80	

Average IO

Single value

Average

N → M Ascribe your

	data

① Get the max. IQ in students

BY

Id DESC

1

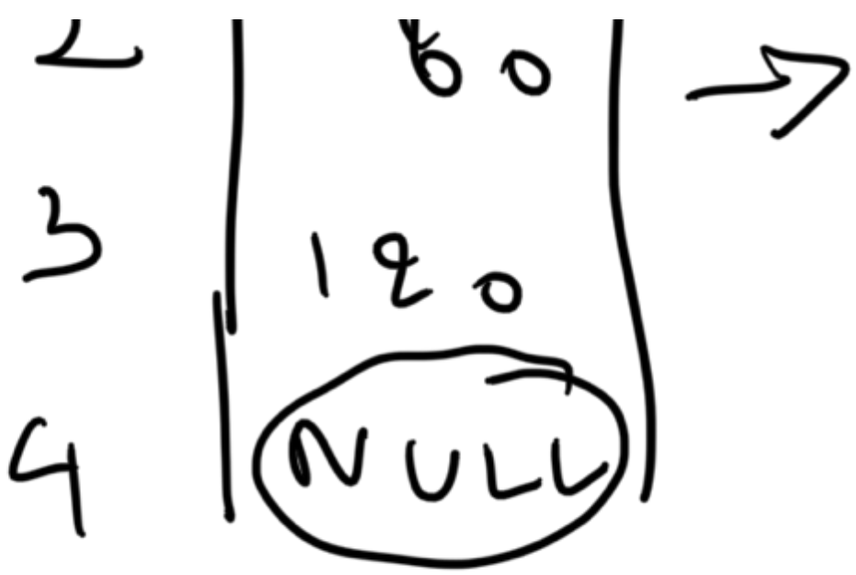
MIN

Select id FROM S

ORDER BY Id LIMIT 1,1

1	100

NULL != 0



Special handling for NULL values

→ Where

→ NULLS Last

Aggregate → only work on non-null values

ISO

1. MAX
2. MIN
3. COUNT
4. SUM
5. AVG

window function

Group by

ID	NAME	Last
1	Tanmay	Kacher
2	Tony	Kakkar
3	Tinmoy	Bhatt

Tanmay | Tony | Tenoy

1. Where

Select * from users
where name = Tonmoy

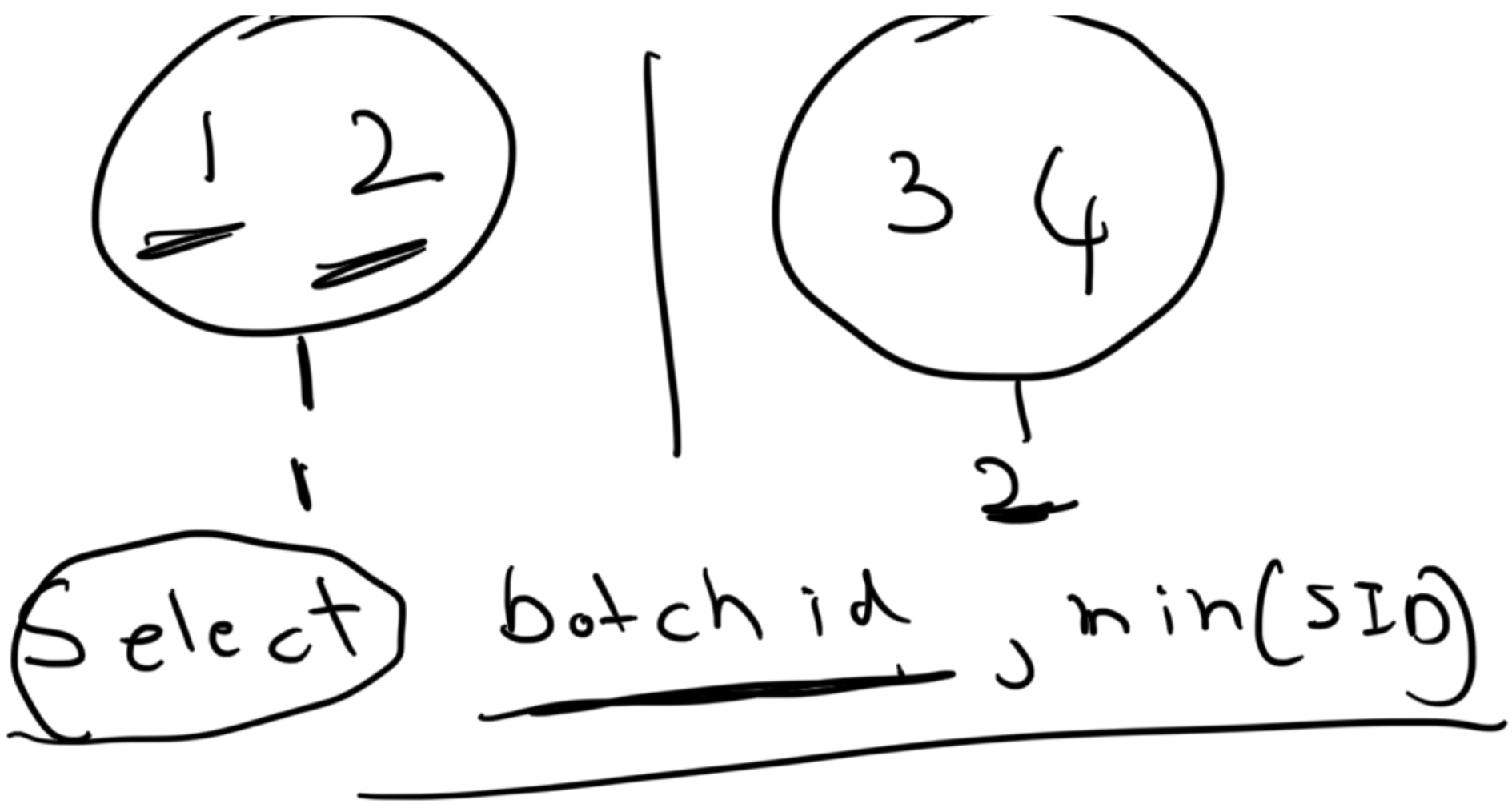
Tony

generally

S

SID	BID
1	1
2	1
3	2
4	2

GROUP BY BATCH ID


1 2
3 4
1
2
Select
batch id , min(SID)


BID
SID
1
1
2
3
1
2