

Pre-Aggregation： SQL 優化的 **「曉」**之戰略

如何停止在中間結果上浪費資源



Based on: 情報の存在論

任務：追獵叛忍



曉組織如何尋找間諜？
他們會逐村、逐個忍者詢問「你是不是叛忍」嗎？

不。那樣太慢了。

戰略：先彙整，後接觸

- 任务失敗率
- 行蹤中斷
- 查克拉特徵異常
- 忍術使用頻率

任務紀錄
(Mission Logs)

彙整
(Aggregate)

行為模式
(Noto Serif TC Light)

黑市名冊
(Black Market List)

JOIN

暴力法的代價

Table A:
1,300,000 Rows

中間結果
(Intermediate Results)

B (5 Rows)

C (5 Rows)

D (5 Rows)

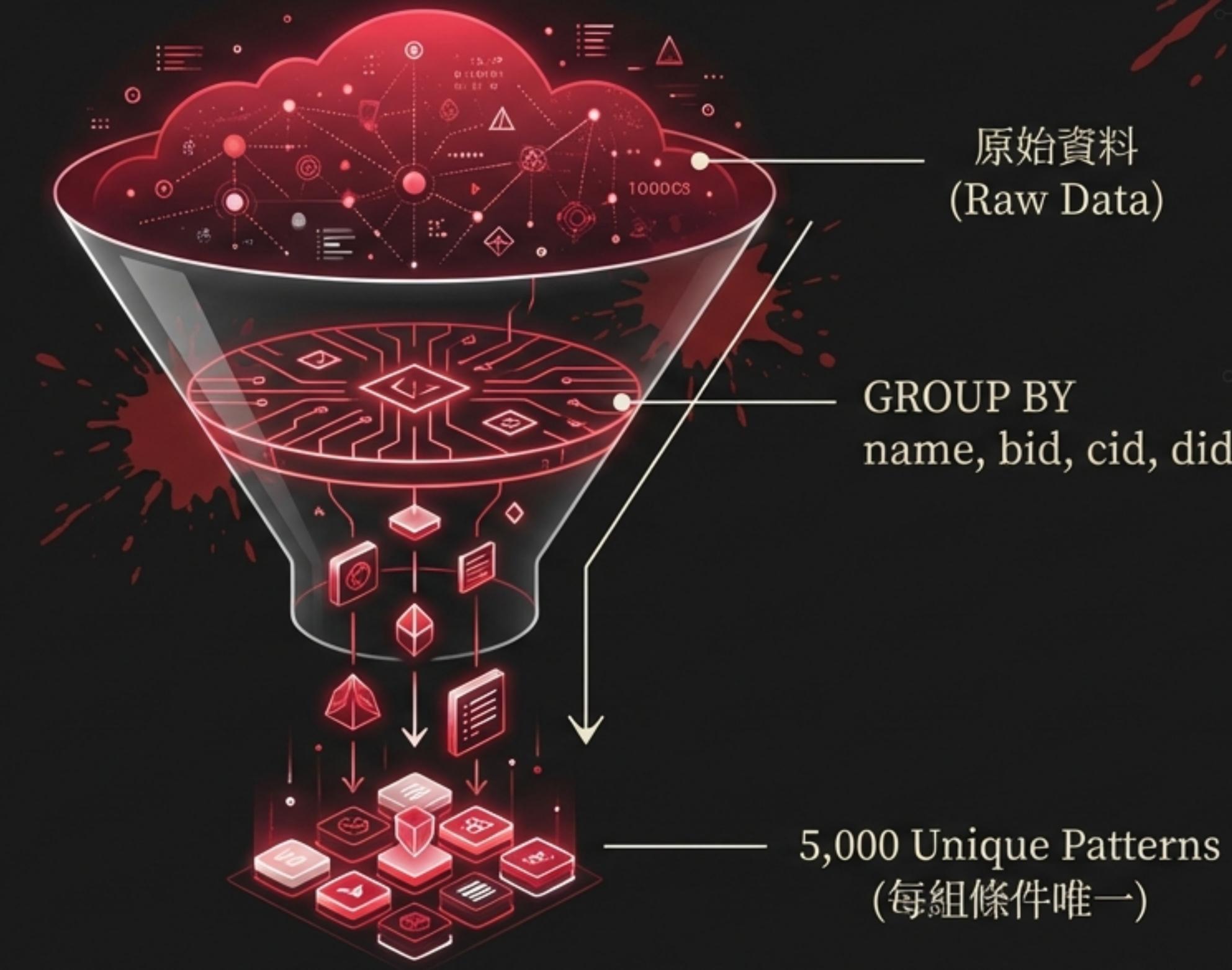
```
FROM a \
  \JOIN b ...
  \JOIN c ...
  \JOIN d ...
GROUP BY ...
```

查克拉耗損：中間結果的膨脹

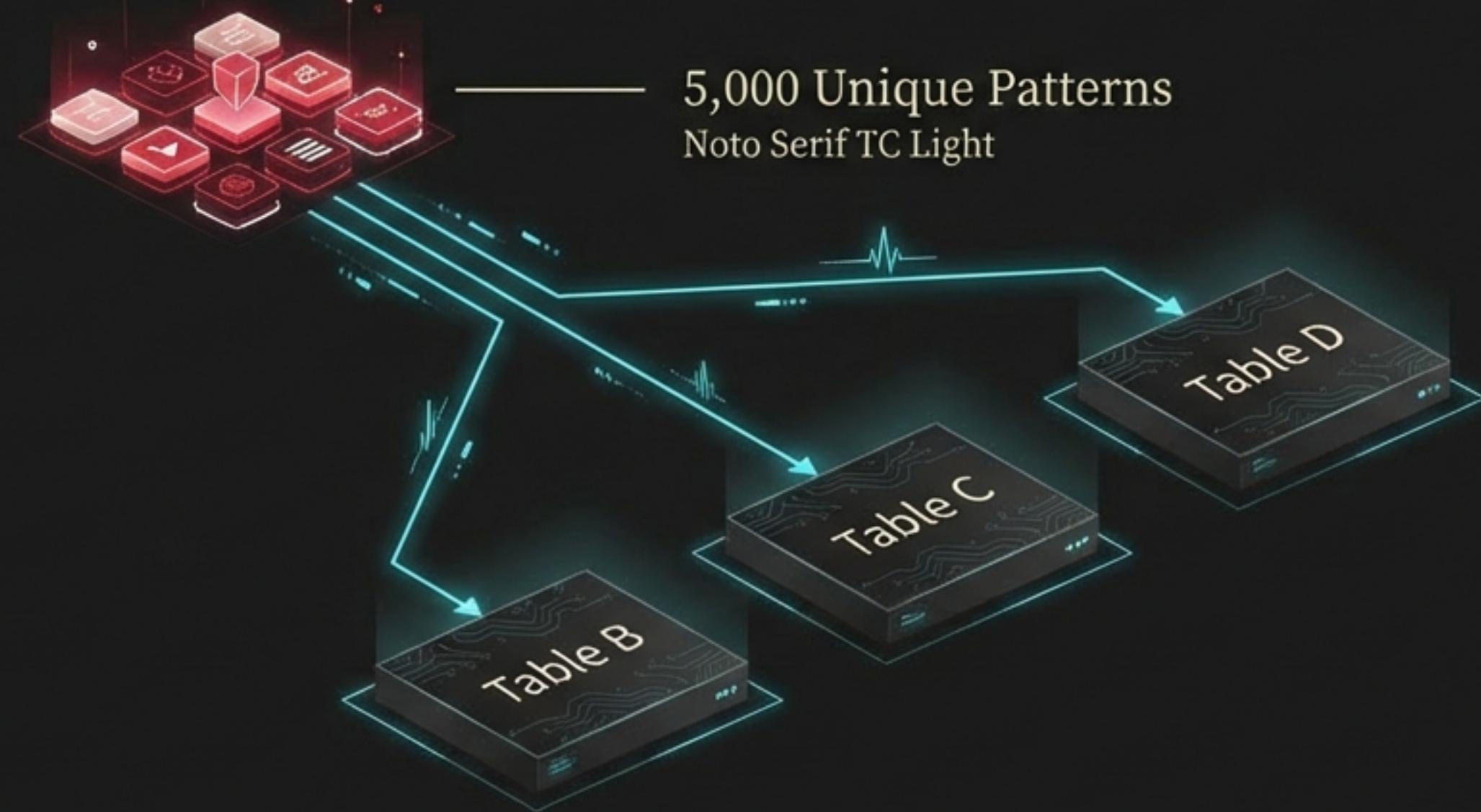
- JOIN 是非常貴的操作。
- 你浪費超多力氣在絕大多數根本沒問題的忍者身上。



轉折：先壓縮，再 JOIN



精準打擊



Before: 1,300,000 × Reference Tables = ⚡

After: 5,000 × Reference Tables = ⚡ (Instant)

術式：高效能版本

```
SELECT a.name, a_sum, a_avg, b.col1...
FROM (
    SELECT name, SUM(count)...
    FROM a
    GROUP BY name, bid, cid, did
) a
JOIN b ON a.bid = b.id
JOIN c ON a.cid = c.id
...
```

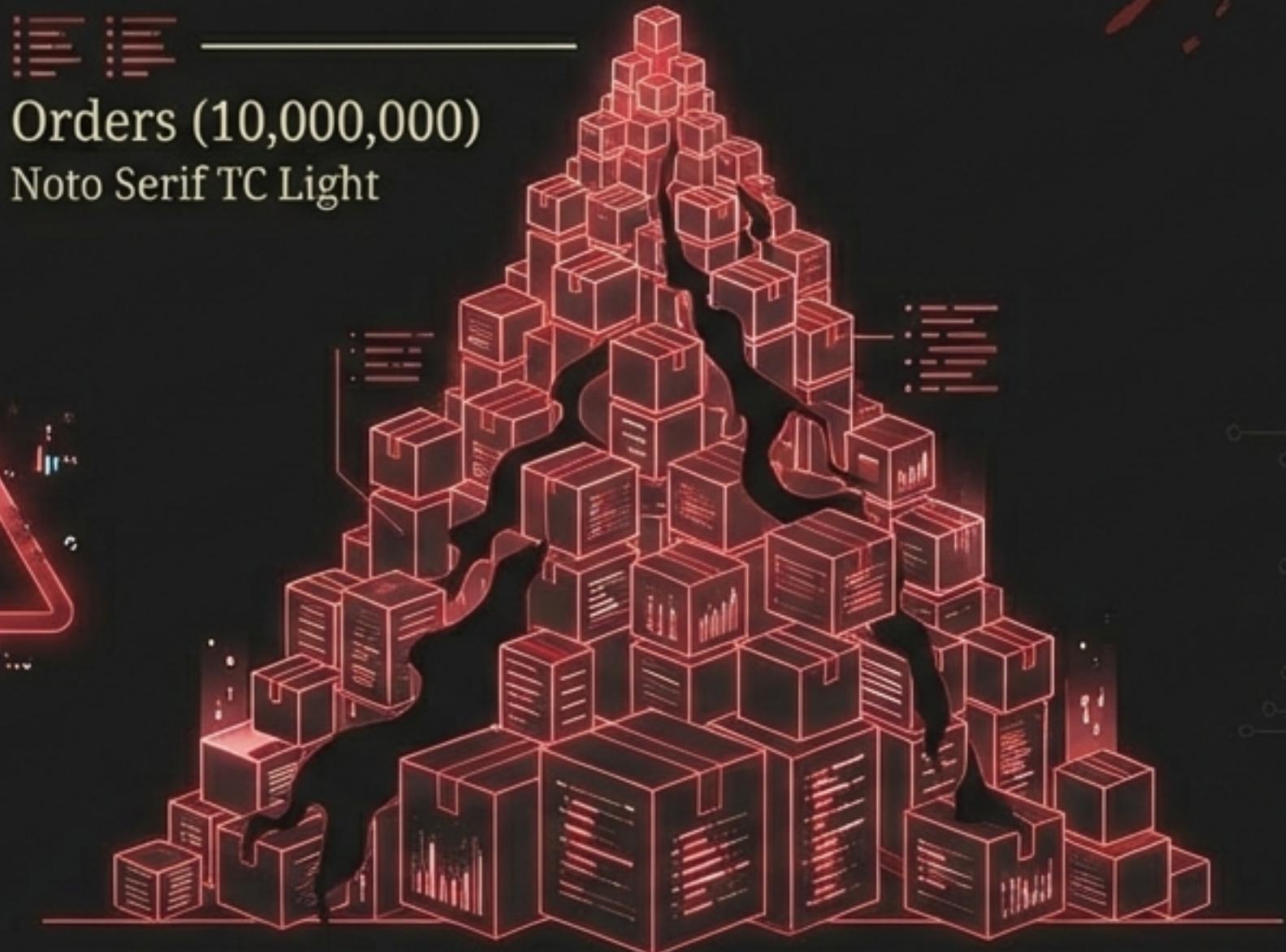
減少資料維度
(Data Dimension Reduction)

實戰：客戶與訂單

Customers (1,000)
Noto Serif TC Light



Orders (10,000,000)
Noto Serif TC Light



Don't let the order history drown the customer list.
不要讓訂單歷史淹沒客戶名單。

成本的數學

JOIN Cost \approx Left Rows \times Right Rows

Naive Approach

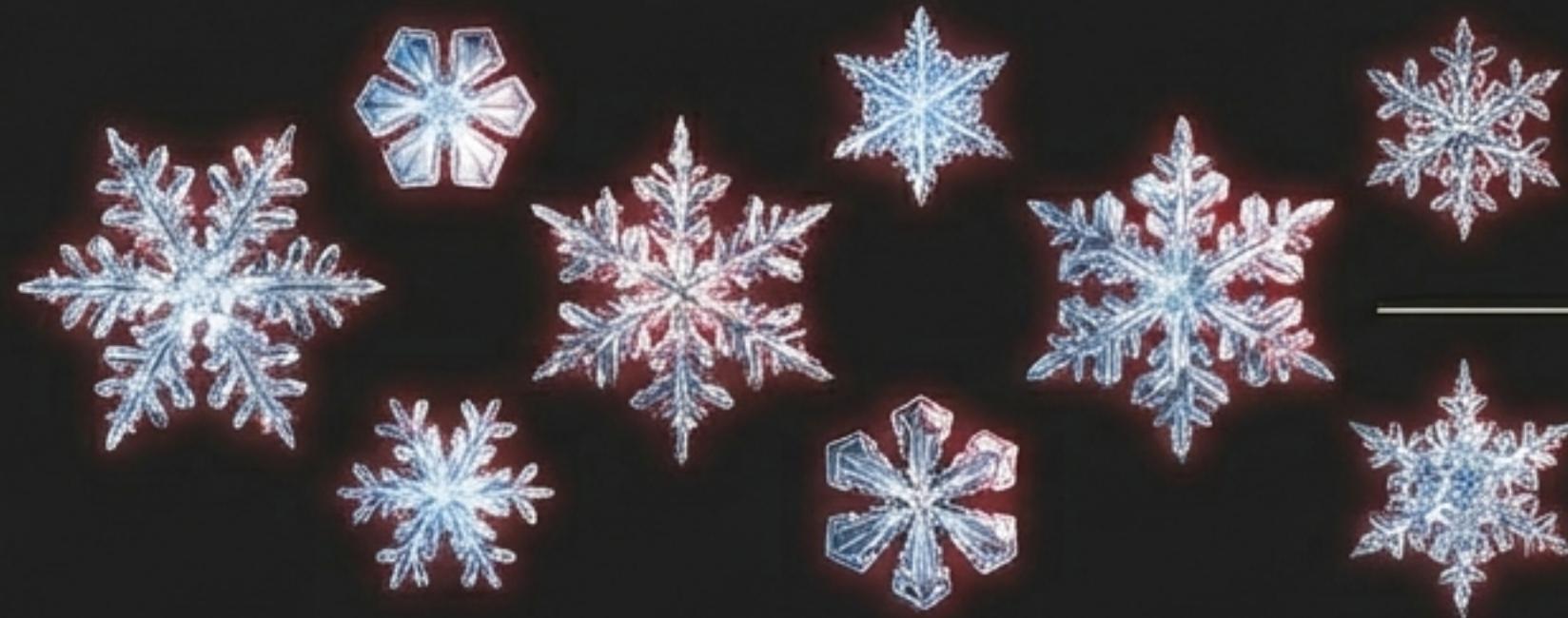
$$\begin{aligned} & 1,000 \text{ Customers} \\ & \times \\ & 10,000,000 \text{ Orders} \\ = & 10,000,000,000 \text{ Operations} \end{aligned}$$

Pre-Aggregation

$$\begin{aligned} & 1,000 \text{ Customers} \\ & \times \\ & 1,000 \text{ (Aggregated Orders)} \\ = & 1,000,000 \text{ Operations} \end{aligned}$$

99.99% Reduction

極限：基數的陷阱



High Cardinality (高基數)
Noto Serif TC Light

GROUP BY 是否一定能減少資料量？

如果每筆資料都是獨一無二的
(Unique Snowflake)，那麼：



Optimization Fails. 

工具一：暫存表 (Temporary Tables)

```
CREATE TABLE #temp_order_counts ...
INSERT INTO #temp_order_counts
    SELECT ... GROUP BY ...
...
JOIN #temp_order_counts ...
```

Best for complex
steps & debugging.

工具二：CTEs (通用資料表運算式)

```
WITH temp_order_counts AS (
    SELECT ... GROUP BY customer_id
)
```

```
SELECT ... FROM customers
JOIN temp_order_counts ...
```

Readable,
Logical Flow.



工具三：資料表變數 (Table Variables)

@variable



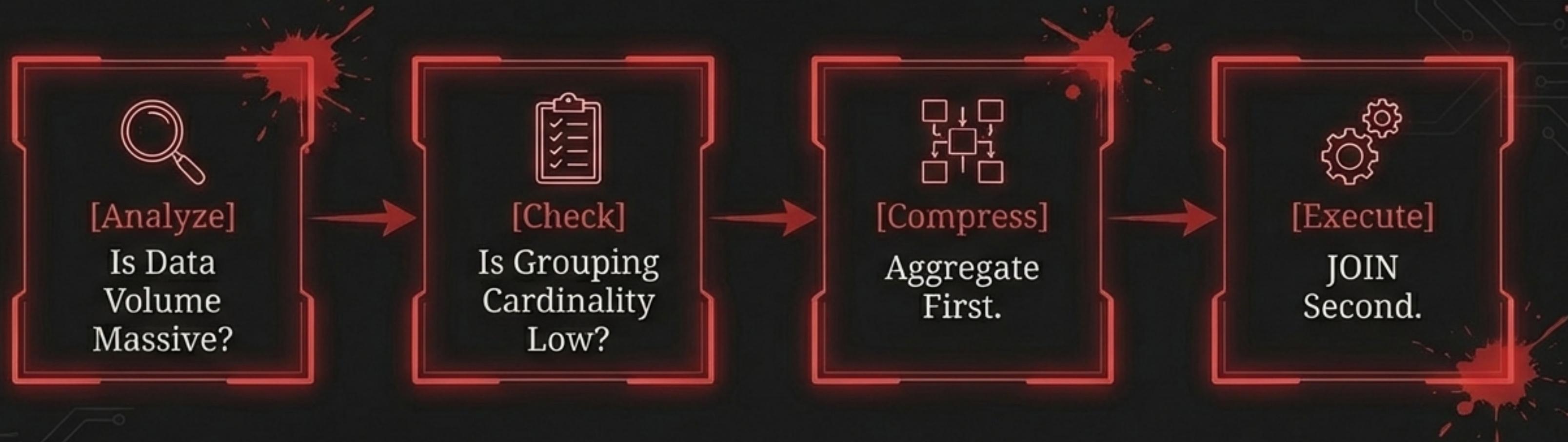
```
DECLARE @temp_order_counts TABLE ...  
  
INSERT INTO @temp_order_counts ...  
  
SELECT ...  
... FROM ... JOIN @temp_order_counts
```

Strictly scoped
storage for
compressed
datasets.





最終奧義 (Final Wisdom)



The smartest ninja isn't the one with the most chakra,
but the one who wastes the least.

(最聰明的忍者不是查克拉最多的，而是浪費最少的。)

Review your slow queries today. Find the Big Join.