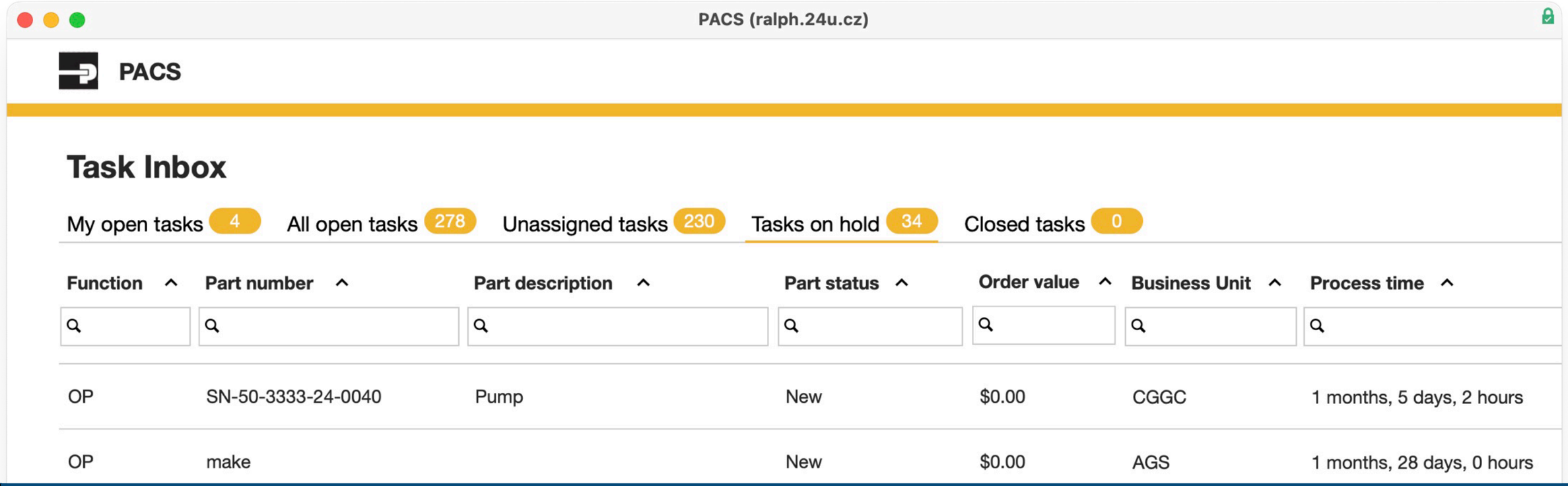
Performance Optimization Story

PACS (Part Add Change System)

A tracking system for **managing part requests, approvals, and order processes** across customer service, engineering, operations, and supply chain, ensuring each step is completed in sequence from lower-level parts to top-level parts.



Top 5 longest scripts based on total time

Script	Server		Client		All Together	
	Time	Instances	Time	Instances	Time	Instances
Startup	03:48:40	1932	00:06:56	28	03:55:36	1960
SS EMAILQUE Send e-mails	00:43:14	1864			00:43:14	1864
TASK Filter (filter)	00:02:58	1932	00:00:26	99	00:03:24	2031
TASK Go to overview and filter (filter)	00:02:11	1932	00:00:29	54	00:02:41	1986
Set statup globals	00:01:03	1932			00:01:56	17
Refresh window			00:01:46	21		
TASK Process create new task (json)			00:00:18	12		

Slow per user experience

Picked for optimization by FM Bench

Top 5 longest scripts based on average time

Script	Server		Client		All Together	
	Time	Instances	Time	Instances	Time	Instances
Startup	7.102s	1932	14.884s	28	7.213s	1960
SS EMAILQUE Send e-mails	1.392s	1864				
SS EMAILQUE Delete sent e-mails	0.092s	68				
TASK Go to overview and filter (filter)	0.068s	1932				
Set statup globals	0.033s	1932				
Refresh window			5.069s	21	5.069s	21
TASK Open choose customer			4.021s	3	4.021s	3
TASK Process create new task (json)			1.574s	12	1.574s	12
TASK Customer Filter			1.129s	3	1.129s	3

First Round of Discovery

- 28 client sessions too low - data collected from only 3 users
- Startup performing tasks not needed for server-side sessions
- Scripts perceived as slow were actually taking less than a second to execute
  - **Perceived performance** affected by **layout rendering after** the Filter scripts

Measuring Layout Rendering

Edit Script “TASK Go to overview and filter (filter)” (PACS (ralph.24u.cz))

```
1  Set Variable [ $~~~~~ ; Value: TimeLog ( "start" ; "" ) ]
2  Go to Layout [ “scpt_tasks” (TASK) ; Animation: None ]
3  Perform Script [“TASK Filter (filter)” ; Specified: From list ; Parameter: #P ( 1 ) ]
4  Go to Layout [ “Tasks Overview” (TASK) ; Animation: None ]
5  Go to Record/Request/Page [ First ]
6  Execute FileMaker Data API [ Select ; Target: $timelog_response ; TimeLogDapiJson ( "exit" ; "end" ) ]
7  Set Variable [ $timelog ; Value: "" ]
8  Set Variable [ $$layout_instance_id ; Value: TimeLogInstanceID ]
9  Set Variable [ $~~~~~ ; Value: TimeLogCustom ( "start" ; Get ( ScriptName ) ; "Layout" ; Get ( LayoutName ) ; $$layout_instance_id ) ]
10 Set Variable [ $$timelog_layout_start_time ; Value: Get ( CurrentTimeUTCMicroseconds ) ]
11 Set Variable [ $$timelog ; Value: $timelog ]
12 Install OnTimer Script [“TimeLog Layout Ontimer” ; Interval: .001 ]
```

Measuring Layout Rendering

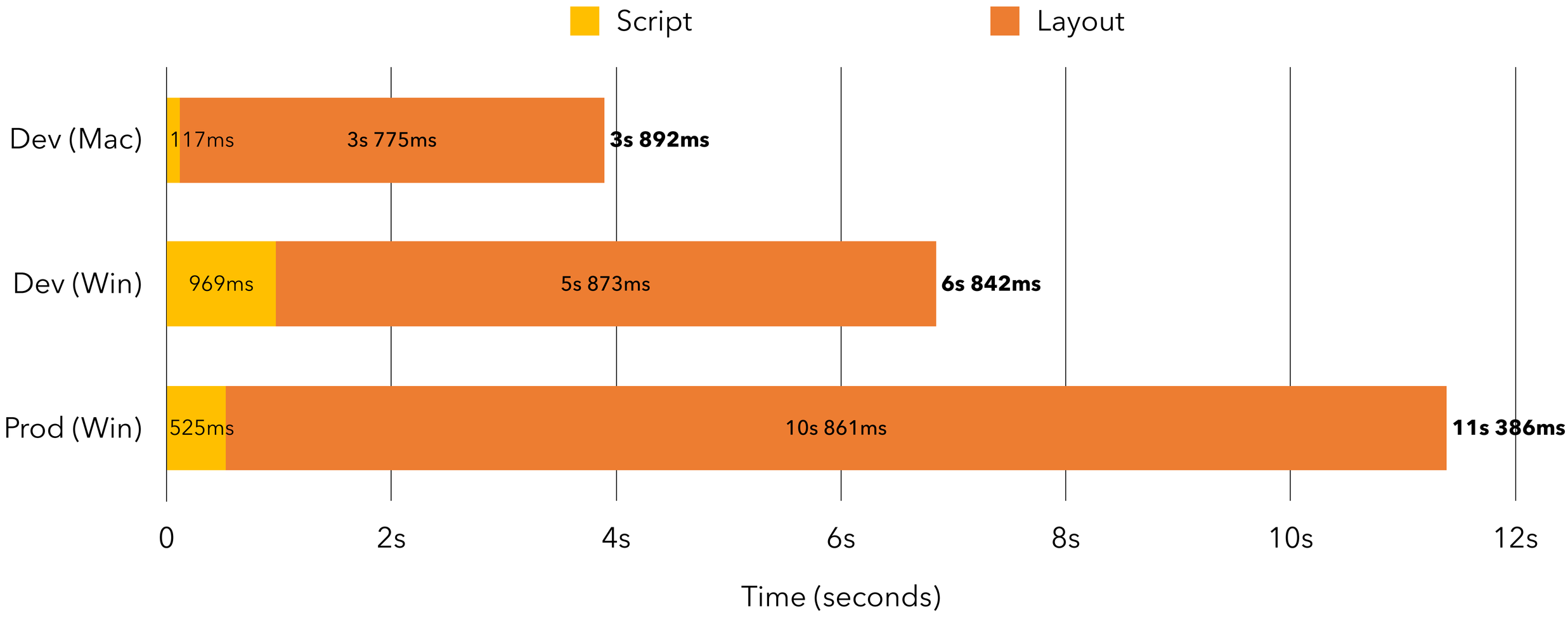
Edit Script “TASK Go to overview and filter (filter)” (PACS (ralph.24u.cz))

```
1  Set Variable [ $~~~~~ ; Value: TimeLog ( "start" ; "" ) ]
2  Go to Layout [ “scpt_tasks” (TASK) ; Animation: None ]
3  Perform Script [“TASK Filter (filter)” ; Specified: From list ; Parameter: #P ( 1 ) ]
4  Go to Layout [ “Tasks Overview” (TASK) ; Animation: None ]
5  Go to Record/Request/Page [ First ]
6  Execute FileMaker Data API [ Select ; Target: $timelog_response ; TimeLogDapiJson ( "exit" ; "end" ) ]
7
8
9
10
11
12
```

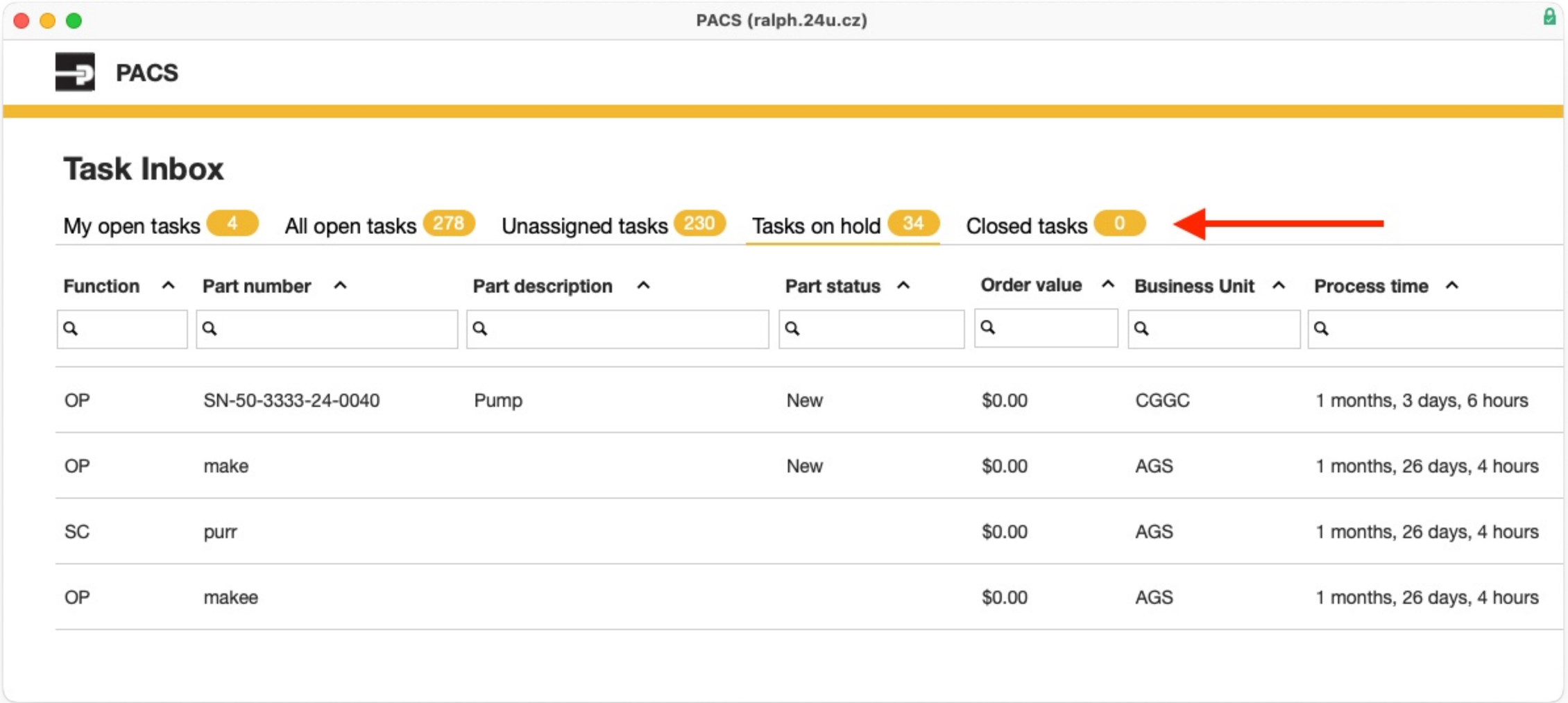
Edit Script “TimeLog Layout Ontimer” (PACS (ralph.24u.cz))

```
1  Install OnTimer Script [ ]
2  Set Variable [ $instance_id ; Value: $$layout_instance_id ]
3  Set Variable [ $$layout_instance_id ; Value: "" ]
4  Set Variable [ $timelog ; Value: $$timelog ]
5  Set Variable [ $time ; Value: Round ( ( Get ( CurrentTimeUTCMicroseconds ) - $$timelog_layout_start_time - 1000 ) / 1000 ; 0 ) & " ms" ]
6  Execute FileMaker Data API [ Select ; Target: $timelog_response ; TimeLogDapiJsonCustom ( "exit" ; $time ; "Layout" ; Get ( LayoutName ) ; $instance_id ) ]
7  // Show Custom Dialog [ $time ]
```

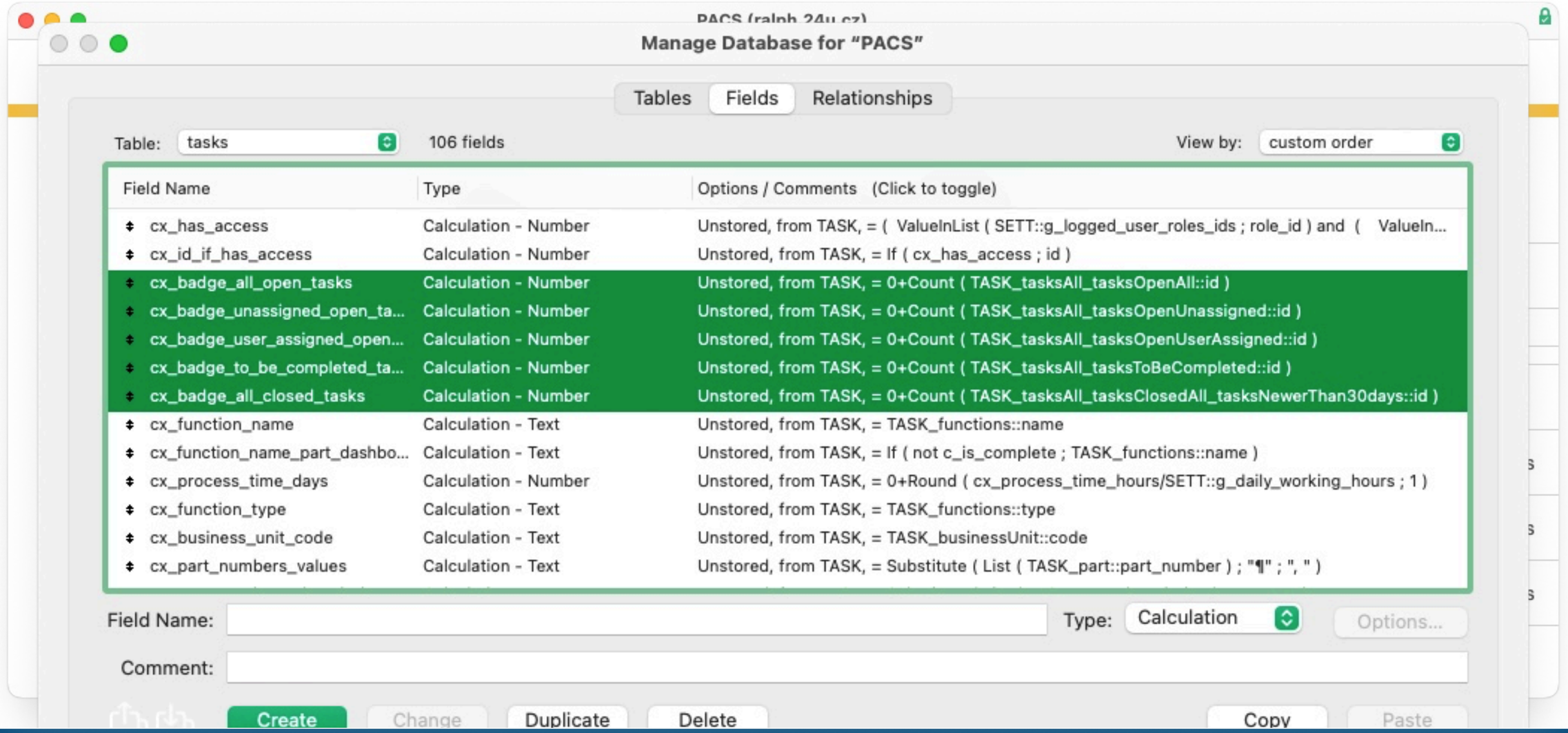
TASK Filter Script + Layout Rendering



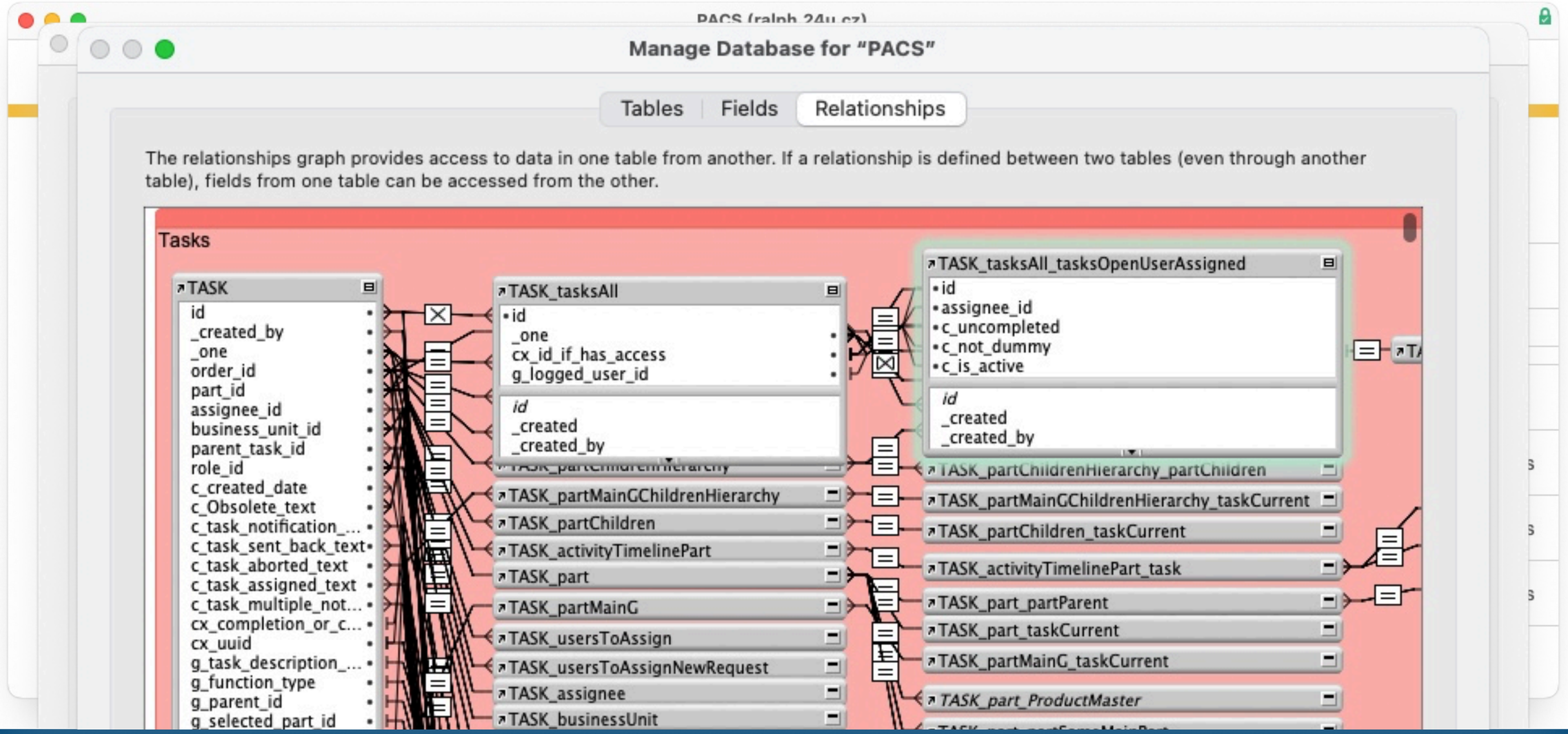
Why It Was Slow



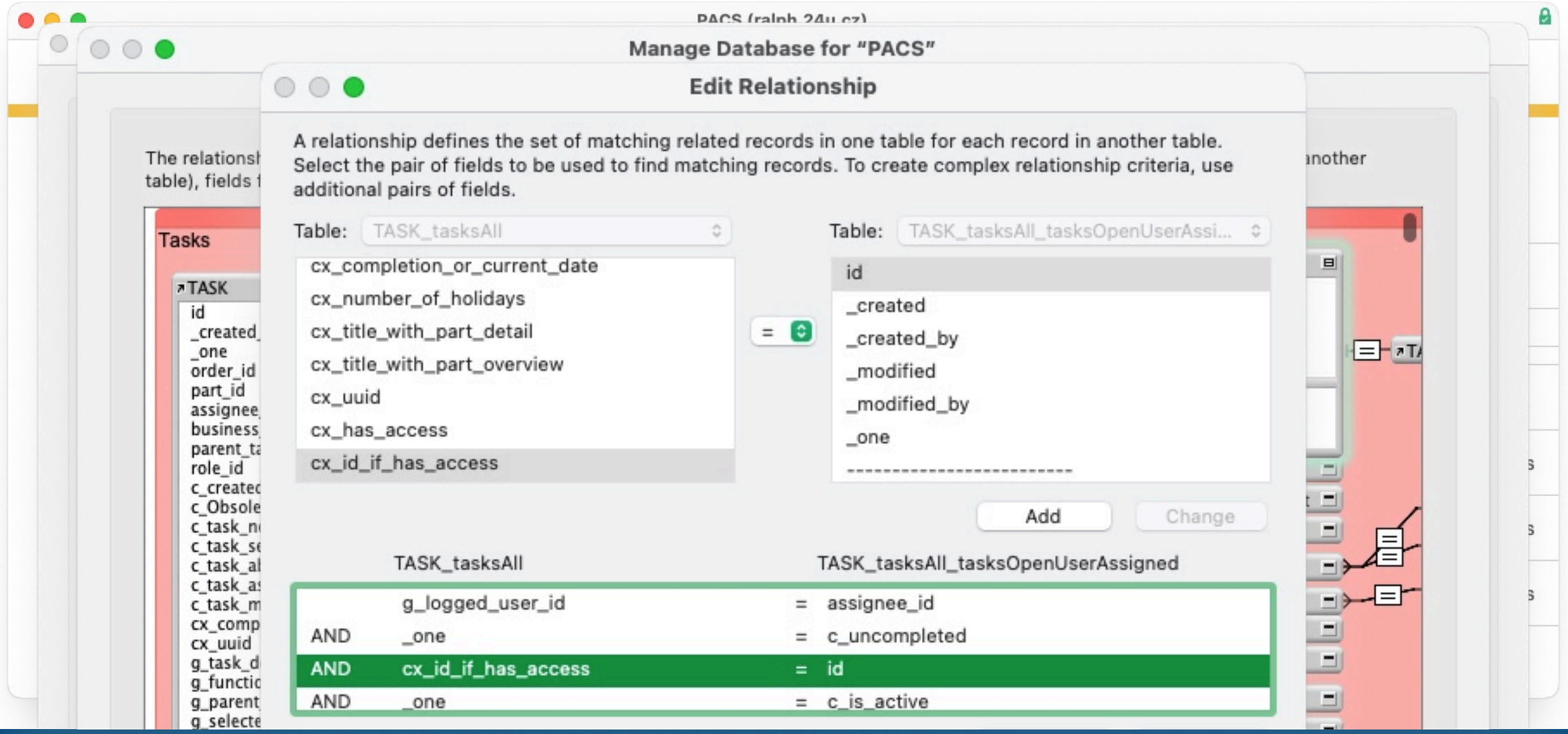
Why It Was Slow



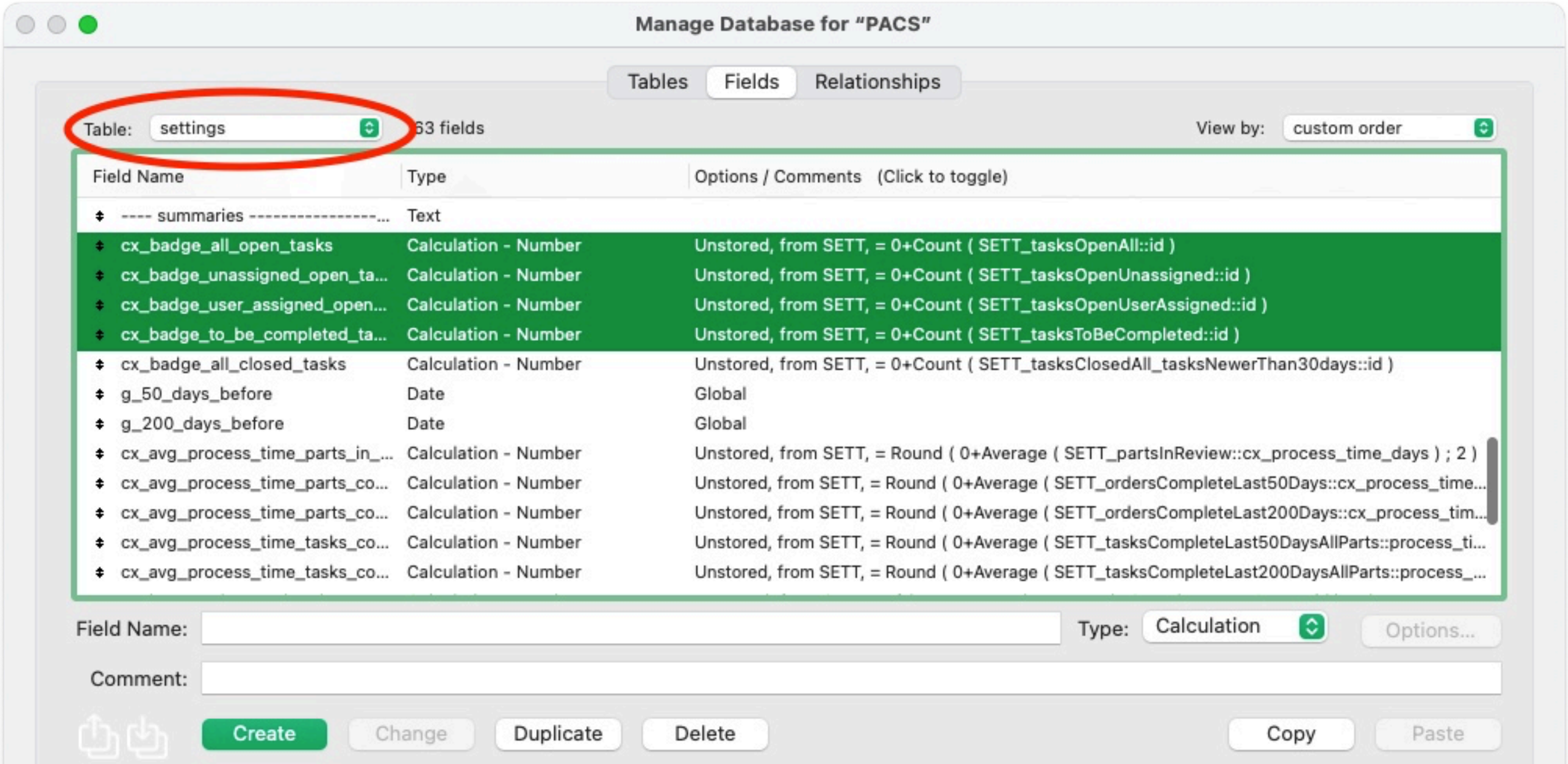
Why It Was Slow



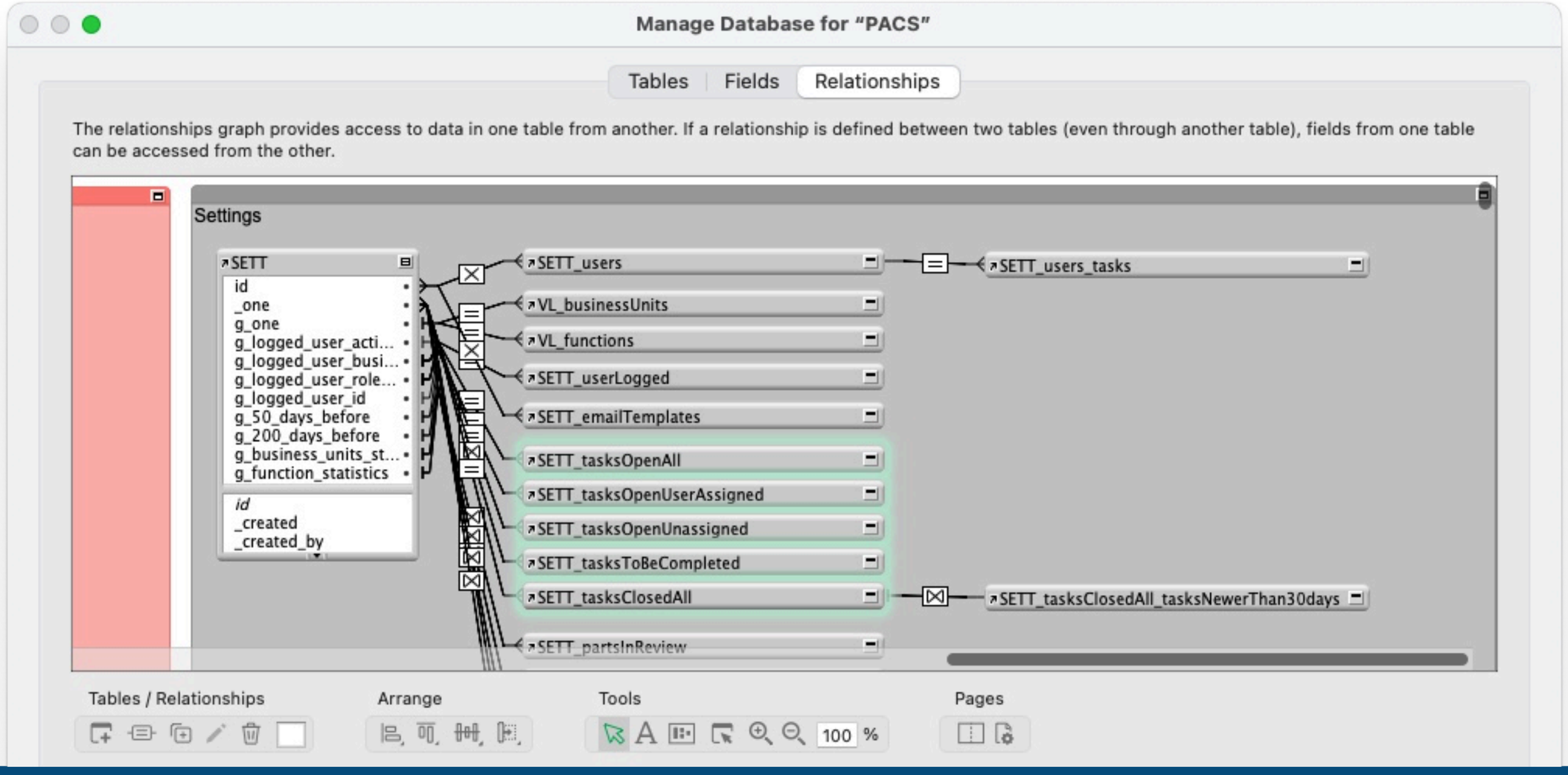
Why It Was Slow



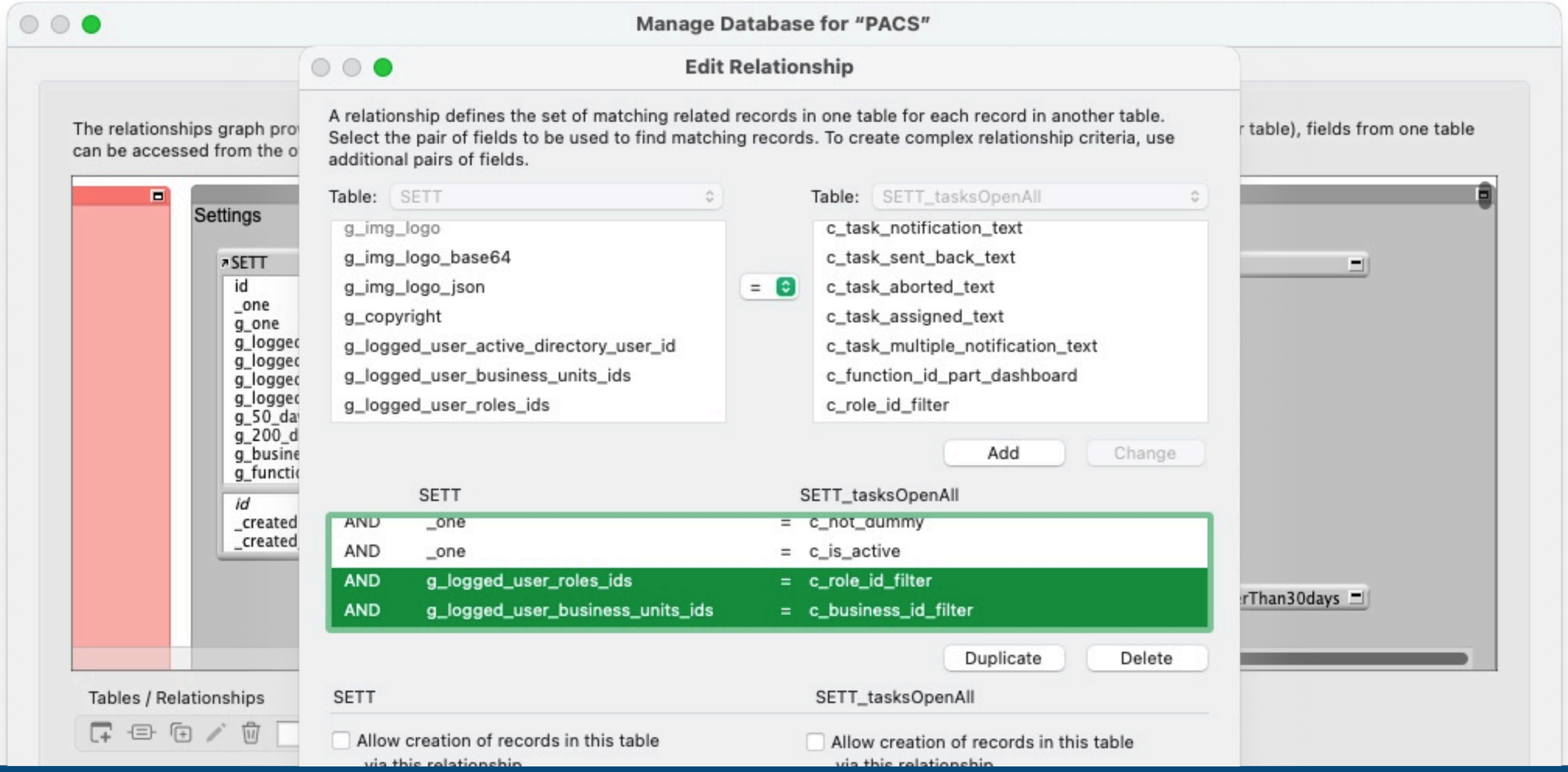
How We Optimized It



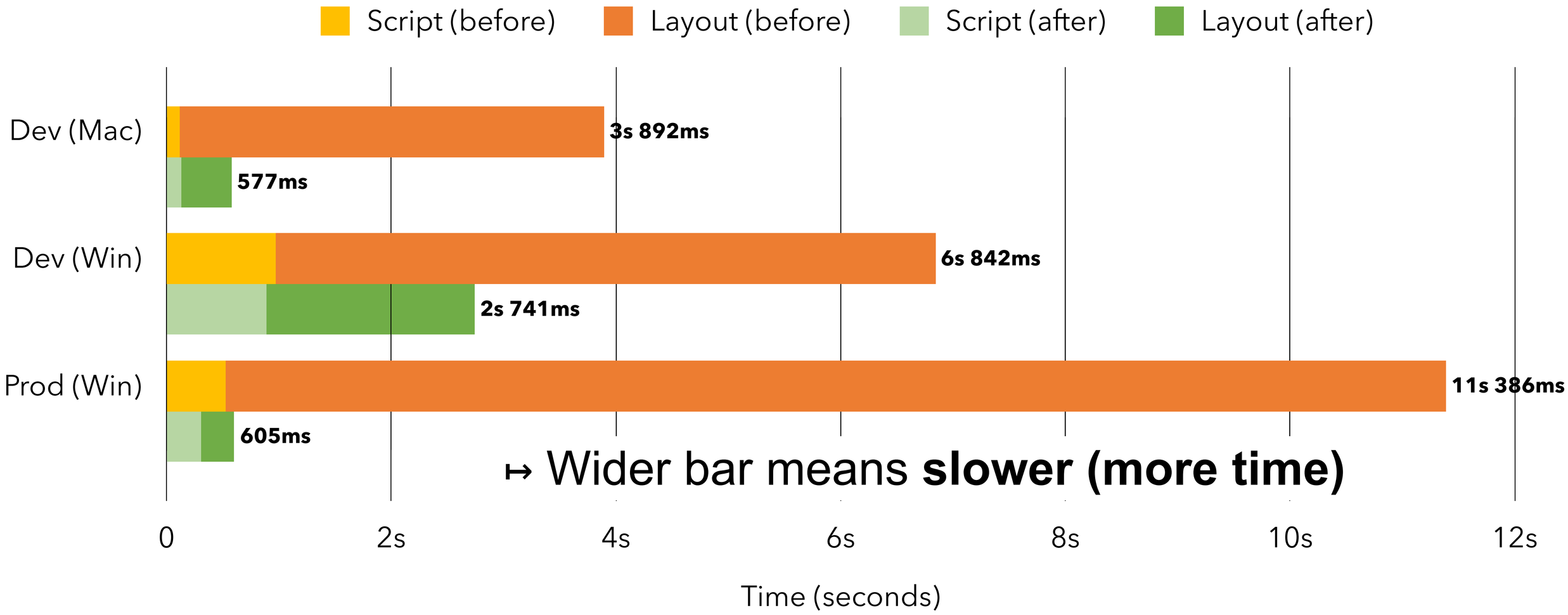
How We Optimized It



How We Optimized It



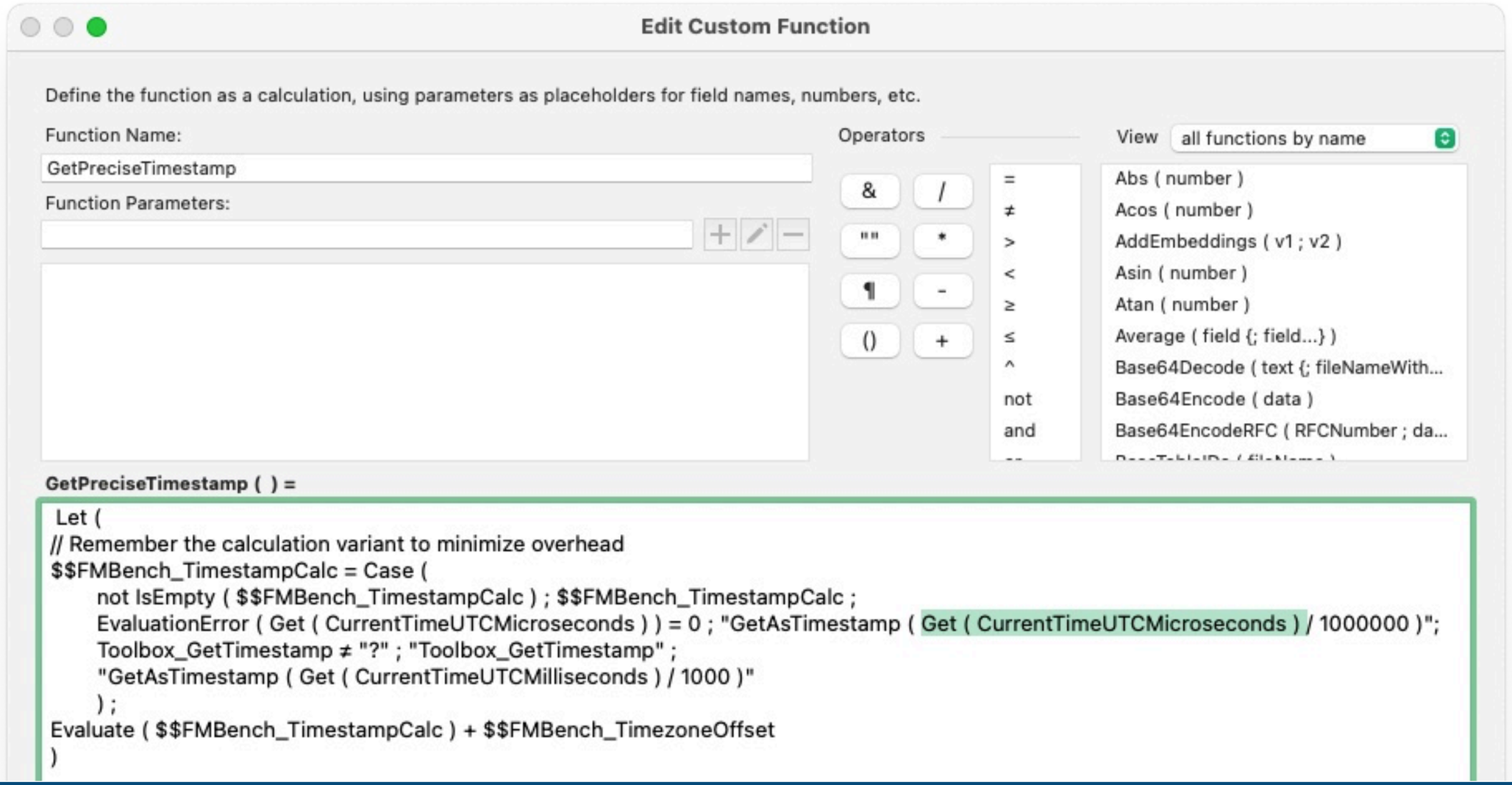
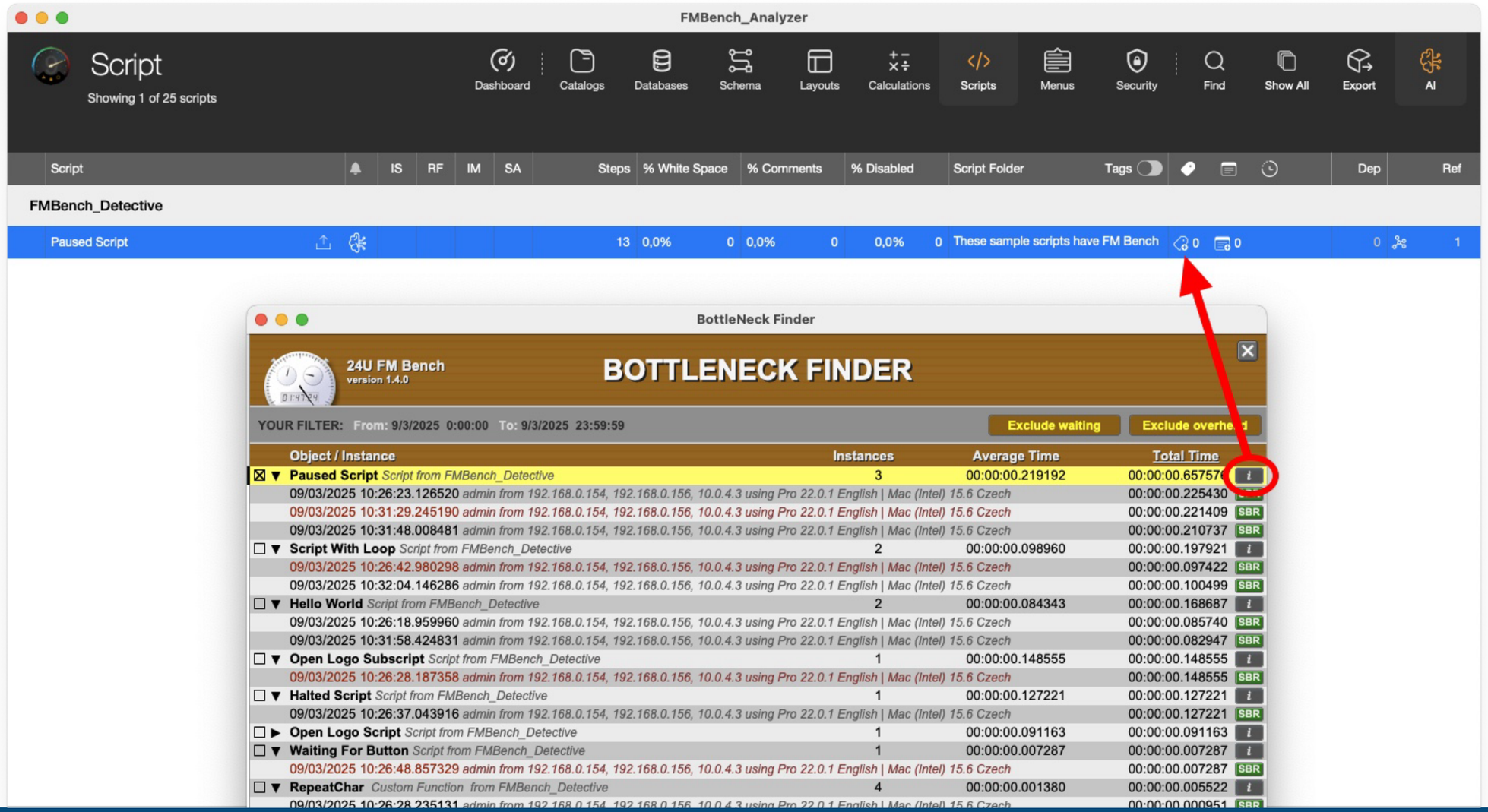
TASK Filter Script + Layout Rendering (Optimized)



Takeaways

- Always consider both measured and perceived performance
- FileMaker Pro still mostly single-threaded
  - Layout rendering time can be measured accurately using OnTimer script trigger
- No need to over-optimize

New Version of FM Bench



New Version of FM Bench