SYNERGY 2026 DEVPARTNER CONFERENCE

Harmony Core in 2026

What shipped this year, and early work on GantrySQL

1 **This year's releases**

8.0.117 · 8.0.118 · 10.0.43

2 **Harmony Core on GantrySQL**

early work, and a proof of concept

JEFF GREENE, CTO / SYNERGEX DEVPARTNER 2026

SYNERGY DEVPARTNER CONFERENCE 2026

A Harmony Core service today



- Customer portals
- Mobile apps
- Partner integrations
- Reporting tools

Synergy repository → CodeGen

Your services were generated from your repository.

Your DBL applications use the same ISAM files at the same time.

Three releases since October 2025



#388 · SEP 2025

Group-by and totals in $apply

answers #388

#392 · FEB 2026

Bridge processes shut down cleanly on Linux

answers #392

#396 · AUG 2026

.NET 10 and EF Core 10

answers #396

ALSO THIS YEAR

Oct 2025 JWT auth template fixes

Aug 2026 SSH.NET and MessagePack security updates

Jul 2026 Synergy 2026.05 packages

Oct 2026 Linux smoke test in CI, Traditional Bridge included

Harmony.Core on NuGet: 1.4M downloads

8.0.117: $apply totals are computed in the web server

YOUR CONSUMERS	YOUR SERVICES	HARMONY CORE	DATA ACCESS	YOUR DATA
get the answer	the same URL	in your web server	Select class	ISAM file

```
GET /odata/Customers?
$apply=groupby(
  (State),
  aggregate(
    CreditLimit
    with sum as
    TotalCredit))
```

CA	$81,000
WA	$32,000
TX	$15,000
OR	$9,000

1	CA	$25,000
2	CA	$18,000
3	WA	$12,000
4	CA	$30,000
5	OR	$9,000
6	TX	$15,000
7	CA	$8,000
8	WA	$20,000

Enumerable.Sum

8 records read

1	CA	$25,000
2	CA	$18,000
3	WA	$12,000
4	CA	$30,000
5	OR	$9,000
6	TX	$15,000
7	CA	$8,000
8	WA	$20,000

“I wouldn't recommend you use this for data files accessed remotely via xfServer…”

Our reply on #388

8.0.118: bridge processes no longer outlive the client



8.0.118 · 23 MARCH 2026

- Shutdown always kills the process, then waits for it to exit.
- Closing a pool destroys every bridge process in it.
- Grace period set by HARMONYCORE_SHUTDOWN_TIMEOUT.

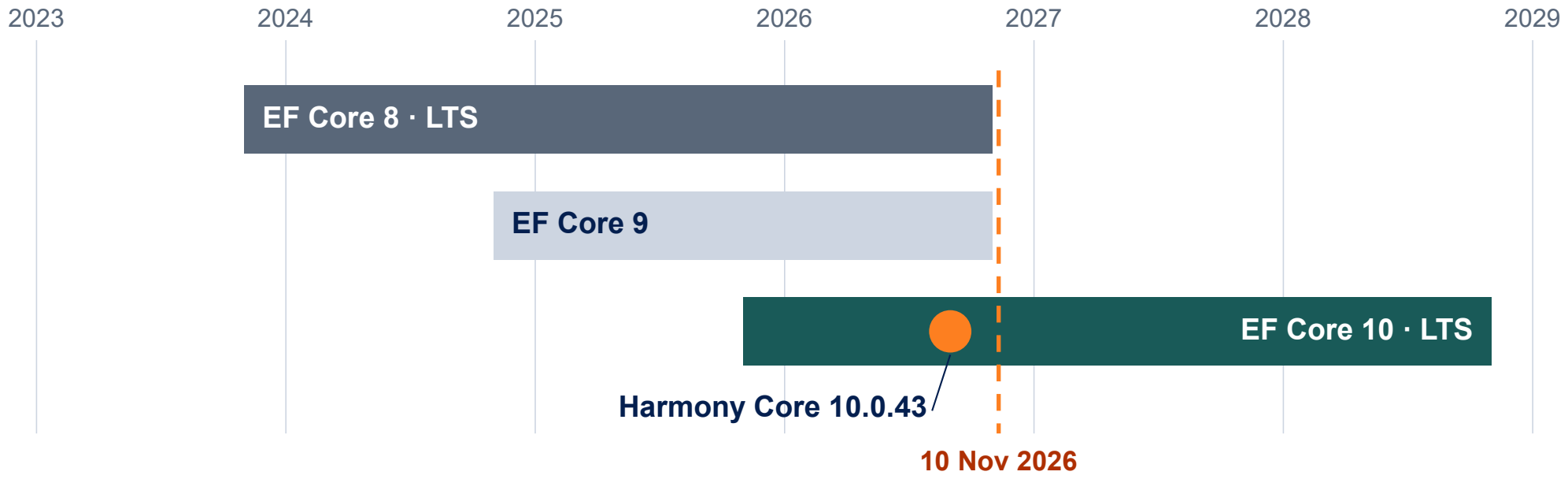
“This is all working perfectly now.”

The customer who reported #392, May 2026

SINCE 13 OCTOBER 2026

Every change is built for Linux, Traditional Bridge included, and smoke-tested in CI.

10.0.43: .NET 10, and EF Core 8 support ends 10 November



On 8.0.118 or earlier? You're on EF Core 8.
Upgrade with the Harmony Core CLI tool.

EF Core 10 is supported until 10 November 2028.

EARLY WORK

Harmony Core on the PostgreSQL EF Core provider, with GantrySQL underneath

YOUR CONSUMERS

Portals · apps
partners · reports

STAYS

YOUR SERVICES

URLs · JSON · ETags
your controllers

STAYS

HARMONY CORE

OData · EF Core 10
Traditional Bridge

STAYS

DATA ACCESS · TODAY

Harmony EF provider
Select class

YOUR DATA · TODAY

ISAM files
DBL programs

DATA ACCESS · NEXT

PostgreSQL EF Core provider (Npgsql)

GANTRYSQL · NEXT

Your ISAM files, mapped in place

Totals next to the data

no more “local files, smaller than memory”

Whole requests run as SQL

joins, $apply pipelines, any(), nested $expand

Your custom endpoints too

their LINQ runs as SQL the same way

SYNERGY DEVPARTNER CONFERENCE 2026

Proof of concept on the Harmony Core sample data



1 **The sample files, and MetaDataTool**
mapping generated from the Harmony Core repository

2 **$apply by state**
GROUP BY runs in GantrySQL; see the SQL log

3 **Revenue by customer state**
three files, two joins, one statement

4 **A $apply pipeline**
filter, group on the item file, filter the groups

5 **any(), three files deep**
becomes nested EXISTS

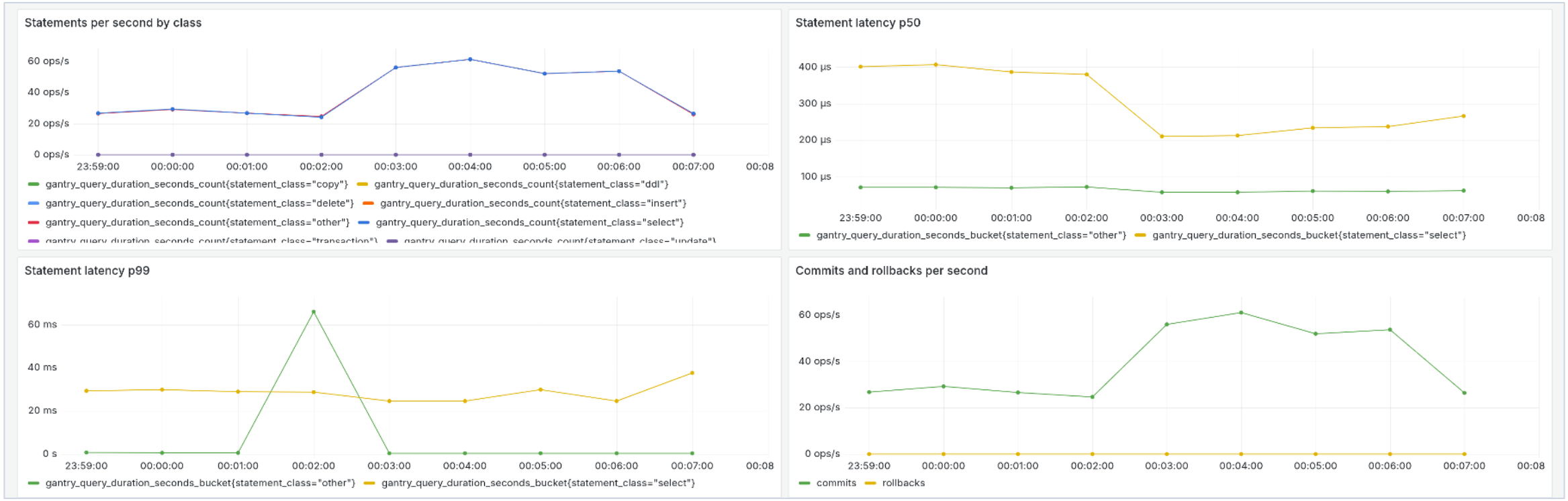
6 **One drill-down request**
nested $expand with its own $filter, $top, $count

7 **dbr addcust**
a DBL program stores a customer; OData sees it

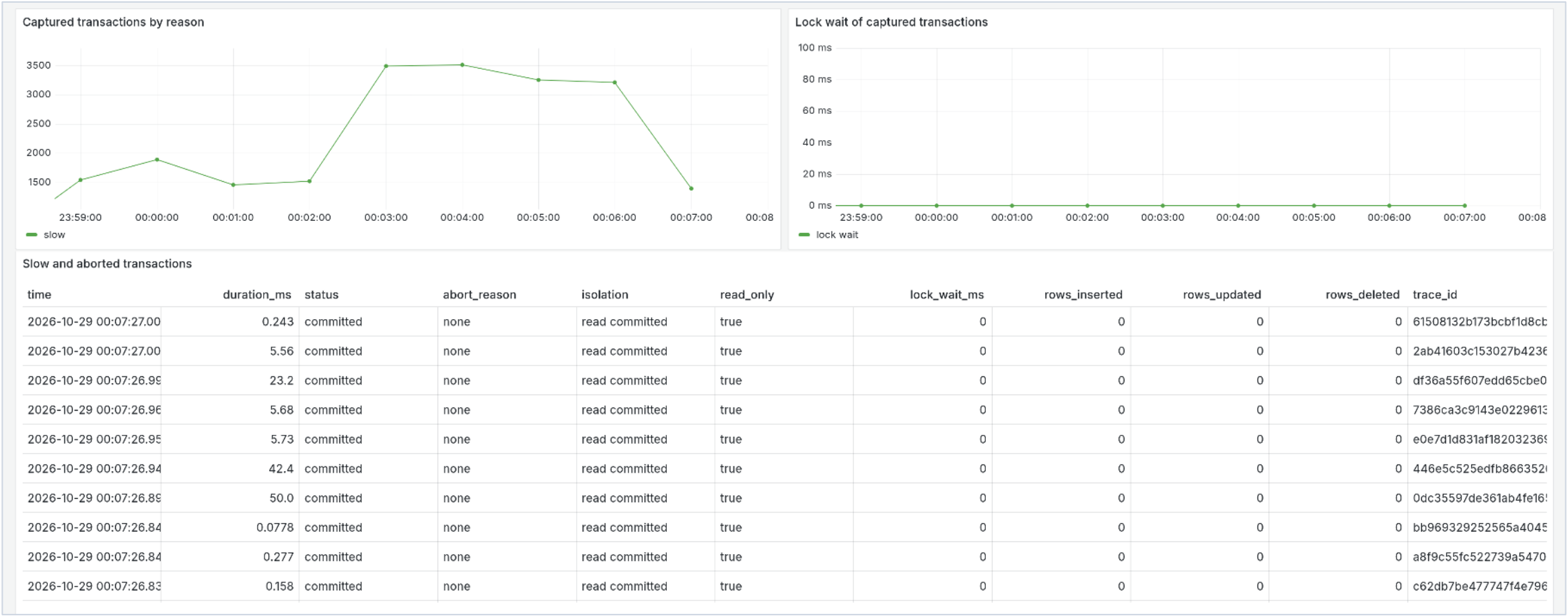
The service is hand-written C# with the sample service's entity names and route; it is not CodeGen output.

SYNERGY 2026 DEVPARTNER CONFERENCE

The demo run in GantrySQL's Overview dashboard



Every request, captured as a transaction



Slow and aborted transactions

time	duration_ms	status	abort_reason	isolation	read_only	lock_wait_ms	rows_inserted	rows_updated	rows_deleted	trace_id
2026-10-29 00:07:27.00	0.243	committed	none	read committed	true	0	0	0	0	61508132b173bcbf1d8cb
2026-10-29 00:07:27.00	5.56	committed	none	read committed	true	0	0	0	0	2ab41603c153027b4236
2026-10-29 00:07:26.99	23.2	committed	none	read committed	true	0	0	0	0	df36a55f607edd65cbe0
2026-10-29 00:07:26.96	5.68	committed	none	read committed	true	0	0	0	0	7386ca3c9143e0229613
2026-10-29 00:07:26.95	5.73	committed	none	read committed	true	0	0	0	0	e0e7d1d831af18203236
2026-10-29 00:07:26.94	42.4	committed	none	read committed	true	0	0	0	0	446e5c525edfb866352
2026-10-29 00:07:26.89	50.0	committed	none	read committed	true	0	0	0	0	0dc35597de361ab4fe16
2026-10-29 00:07:26.84	0.0778	committed	none	read committed	true	0	0	0	0	bb969329252565a4045
2026-10-29 00:07:26.84	0.277	committed	none	read committed	true	0	0	0	0	a8f9c55fc522739a5470
2026-10-29 00:07:26.83	0.158	committed	none	read committed	true	0	0	0	0	c62db7be477747f4e796

What those requests became in SQL

ODATA REQUEST	IN THE SQL GANTRYSQL RAN	ANSWER
Revenue by customer state REL_Order/REL_Customer/State	**2 INNER JOINs · GROUP BY · ORDER BY**	5 states from 14,188 lines; CA 114,367.77
filter / groupby on REL_Item / filter best sellers shipped in 2018	**INNER JOIN · WHERE · GROUP BY · HAVING · ORDER BY · date_part**	5 items over 700 units; top: Scotch Heather, 933
any() inside any() 10+ of a red-flowering item in 2018	**INNER JOIN · nested EXISTS · WHERE · date_part**	6 of 38 customers
Nested $expand with $filter, $orderby, $top=2 and $count	**INNER JOIN · LEFT JOIN LATERAL · WHERE · ORDER BY · LIMIT · date_part**	83 orders counted, 2 returned with lines and items
compute(month(DateOrdered)) /groupby((Month))	**WHERE · GROUP BY · ORDER BY · date_part**	10 months of 2018
$compute=UnitPrice sub CostPrice as Margin, in $filter and $orderby	**WHERE · ORDER BY**	4 items with margin over 9.00

Revenue by customer state: the plan GantrySQL ran

EXPLAIN (ANALYZE, VERBOSE, BUFFERS, KERNELS, SETTINGS, WAL, TIMING, SUMMARY, COSTS)

```
Unified Execution (actual rows=5 time=54.819 ms)
  Compute k16 (project) (actual rows=5 est=553 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
    Compute k15 (project) (actual rows=5 est=553 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
      Sort k14 (actual rows=5 est=553 time=0.0ms loops=3 yields=2 yield_time=0.0ms)
        Compute k13 (project) (actual rows=5 est=553 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
          Compute k12 (project) (actual rows=5 est=553 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
            HashAggregate k11 (actual rows=5 in=14188 est=553 time=13.7ms loops=235 yields=234 yield_time=0.2ms)
              Compute k10 (project) (actual rows=14188 est=5533 time=2.1ms loops=299 yields=67 yield_time=0.0ms)
                HashProbe k9 (adaptive-join-region:3legs) (actual rows=14188 in=18951 est=5533 time=28.1ms loops=82 yields=15 yield_time=0.1ms
region_probes=0 region_raw_rows=4763 region_qualifying_rows=4762 region_cache_hits=35589 region_expansions=39859 region_bindings=26540
region_order_changes=0 region_output_rows=14188 region_pilots_started=4 region_pilots_completed=3 region_pilots_censored=4
region_committed_work=110082250000 region_reused_exploratory_work=0 region_exploratory_work=85532000000 region_peak_debt=1622250000
region_abandoned_requests=0 region_qualifications=0 region_build_attempts=0 region_builds_completed=0 region_builds_abandoned=0 region_entry_changes=0)
                  HashBuild k2 (actual rows=4724 time=0.2ms loops=3 yields=2 yield_time=0.0ms)
                    TableScan k1 public.orders (actual rows=4724 est=4724 time=0.3ms loops=38 yields=37 yield_time=0.6ms)
                  Compute k3 (bloom_filter) (actual rows=14188 time=1.7ms loops=17 yields=9 yield_time=0.0ms)
                    TableScan k0 public.order_items (actual rows=14188 est=14188 time=1.2ms loops=119 yields=111 yield_time=2.1ms)
                    HashBuild k2 (actual rows=4724 time=0.2ms loops=3 yields=2 yield_time=0.0ms)
                      TableScan k1 public.orders (actual rows=4724 est=4724 time=0.3ms loops=38 yields=37 yield_time=0.6ms)
                  HashBuild k7 (actual rows=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                    TableScan k6 public.customers (actual rows=39 est=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
Buffers: shared hit=0 read=0 dirtied=0 written=0
WAL: records=0 fpi=0 bytes=0 buffers_full=0
Settings: enable_async_append = "off", enable_distinct_reordering = "off", enable_gathermerge = "off", enable_group_by_reordering = "off",
enable_incremental_sort = "off", enable_material = "off", enable_memoize = "off", enable_parallel_append = "off", enable_presorted_aggregate = "off",
enable_self_join_elimination = "off", enable_tidscan = "off", geqo = "off", jit_inline_above_cost = "-1", jit_tuple_deforming = "off",
synchronize_seqscans = "off"
Planning Time: 2.38 ms (cache=miss(not_cacheable) optimize=2.38ms lower=0.00ms)
Compilation Time: 0.09 ms (artifact_cache=hit)
Execution Time: 52.22 ms (txn=0.00ms setup=0.00ms drive=52.22ms finalize=0.01ms)
```

Three ISAM files

order_items 14,188 rows · orders 4,724 · customers 39

One adaptive hash join

Three legs, REL_Order and REL_Customer, with a bloom filter in front of order_items.

14,188 rows in, 5 out

HashAggregate groups by state; only the five totals leave the database.

One drill-down request: the plan GantrySQL ran

EXPLAIN (ANALYZE, VERBOSE, BUFFERS, KERNELS, SETTINGS, WAL, TIMING, SUMMARY, COSTS)

```
Unified Execution (actual rows=9 time=40.461 ms)
  Sort k30 (actual rows=9 est=1 time=0.0ms loops=3 yields=2 yield_time=0.0ms)
    Compute k29 (project) (actual rows=9 est=1 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
      HashProbe k28 (hash_join) (actual rows=9 in=47 est=1 time=0.0ms loops=3 yields=2 yield_time=0.0ms)
        Compute k23 (project) (actual rows=9 est=1 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
          HashProbe k22 (hash_join) (actual rows=9 in=228 est=1 time=0.0ms loops=3 yields=2 yield_time=0.0ms)
            TableScan k0 public.customers using CUSTOMER_NUMBER (actual rows=1 in=39 est=1 time=0.0ms loops=1)
              Key: col[0] = 8
            HashBuild k21 (actual rows=227 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
              Compute k20 (project) (actual rows=227 est=3 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                HashProbe k19 (hash_join) (actual rows=227 in=14264 est=3 time=0.9ms loops=3 yields=2 yield_time=0.0ms)
                  Compute k11 (project) (actual rows=76 est=3 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                    Compute k10 (filter) (actual rows=76 in=2584 est=3 time=0.1ms loops=2 yields=2 yield_time=0.0ms)
                      Compute k9 (window) (actual rows=2584 est=17 time=1.2ms loops=5 yields=3 yield_time=0.0ms)
                        Sort k8 (actual rows=2584 est=17 time=0.5ms loops=5 yields=3 yield_time=0.0ms)
                          Sort k7 (actual rows=2584 est=17 time=0.4ms loops=5 yields=3 yield_time=0.0ms)
                            Compute k6 (project) (actual rows=2584 est=17 time=0.1ms loops=3 yields=2 yield_time=0.0ms)
                              HashProbe k5 (hash_join) (actual rows=2584 in=2623 est=17 time=0.5ms loops=5 yields=3 yield_time=0.0ms)
                                HashBuild k3 (actual rows=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                                  TableScan k2 public.customers (actual rows=39 est=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                                Compute k4 (bloom_filter) (actual rows=2584 time=0.3ms loops=3 yields=2 yield_time=0.0ms)
                                  TableScan k1 public.orders (actual rows=2584 in=4724 est=47 time=1.0ms loops=9486 yields=9485 yield_time=3.7ms)
                                  HashBuild k3 (actual rows=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                                    TableScan k2 public.customers (actual rows=39 est=39 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                  HashBuild k18 (actual rows=14188 time=1.3ms loops=37 yields=19 yield_time=0.6ms)
                    Compute k17 (project) (actual rows=14188 est=14188 time=1.3ms loops=37 yields=19 yield_time=0.5ms)
                      HashProbe k16 (hash_join) (actual rows=14188 in=14309 est=14188 time=3.1ms loops=33 yields=14 yield_time=0.4ms)
                        HashBuild k14 (actual rows=121 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                          TableScan k13 public.items (actual rows=121 est=121 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                        Compute k15 (bloom_filter) (actual rows=14188 time=2.0ms loops=25 yields=13 yield_time=0.4ms)
                          TableScan k12 public.order_items (actual rows=14188 est=14188 time=1.2ms loops=123 yields=111 yield_time=2.2ms)
                          HashBuild k14 (actual rows=121 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
                            TableScan k13 public.items (actual rows=121 est=121 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
        HashBuild k27 (actual rows=38 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
          Compute k26 (project) (actual rows=38 est=4 time=0.0ms loops=1 yields=1 yield_time=0.0ms)
            HashAggregate k25 (actual rows=38 in=2584 est=4 time=0.2ms loops=3 yields=2 yield_time=0.0ms)
              TableScan k24 public.orders (actual rows=2584 in=4724 est=47 time=1.0ms loops=9485 yields=9485 yield_time=3.4ms)
Buffers: shared hit=0 read=0 dirtied=0 written=0
WAL: records=0 fpi=0 bytes=0 buffers_full=0
Settings: enable_async_append = "off", enable_distinct_reordering = "off", enable_gathermerge = "off", enable_group_by_reordering = "off", enable_incremental_sort = "off",
enable_material = "off", enable_memoize = "off", enable_parallel_append = "off", enable_presorted_aggregate = "off", enable_self_join_elimination = "off", enable_tidscan =
"off", geqo = "off", jit_inline_above_cost = "-1", jit_tuple_deforming = "off", synchronize_seqscans = "off"
Planning Time: 8.33 ms (cache=miss(not_cacheable) optimize=8.33ms lower=0.00ms)
Compilation Time: 0.17 ms (artifact_cache=hit)
Execution Time: 30.70 ms (txn=0.00ms setup=0.00ms drive=30.70ms finalize=0.03ms)
```

Customer 8 by its ISAM key

customers using CUSTOMER_NUMBER, Key col[0] = 8: one record read.

$top=2 became a window

2,584 2017 orders numbered per customer; the filter keeps 76.

Lines, items, count

Two hash joins for lines and item names, a HashAggregate for $count; 9 rows back.

What we expect would change in a generated project

GENERATED STARTUP (EXPECTED, SIMPLIFIED)

```
- UseHarmonyDatabase(opts, dataObjectProvider)
```

```
+ UseNpgsql(connectionString)
```

Generated by CodeGen from your repository, as today.

EXPECTED TO STAY THE SAME

- URLs and JSON
- Controllers and custom endpoints
- Traditional Bridge calls
- Authentication
- Generated tests