Benedict Corp
BENEDICT APP

PERFORMANCE REPORT · OCTOBER 2026

Benedict App — Performance, Scalability & Reliability Report

Separately verified calculations, realistic workloads, and transparent performance measurements.

Tests run 24–25 October 2026 Datasets of about 100 thousand to 5 million time entries Version 1.1 (November 2026)

All measurements in this report were taken in a **local, isolated test environment** (a single desktop workstation running SQL Server), with synthetic data and simulated users. They are not measurements of Benedict App on Microsoft Azure, and they are not a service-level commitment. Each customer installs Benedict App in their own Azure subscription, where performance depends on the service tiers they choose.

1 Executive summary

We tested Benedict App with four synthetic datasets of increasing size, the largest holding about five million time entries, with up to 100 simulated users at once. We then independently recomputed every finalized financial figure.

5,835,812
report lines of finalized reports recomputed in exact arithmetic across the four datasets: **0 differences**

0 of 6,285
acknowledged saves lost or duplicated during the load tests

213–326 ms
95th-percentile response, 10–100 simulated users, ~5 million entries

0
errors or time-outs in 54,078 requests across nine load runs

What we found

- **Everyday work stays fast as data grows.** MEASURED A team member's month view took a median 6 ms at about 100 thousand entries and 10 ms at about 5 million (single requests).
- **Performance under concurrent use.** MEASURED With 10–100 simulated users, the median response time was 8–9 ms at about 1 million entries and 19–44 ms at about 5 million. The 95th percentile was 63–459 ms and 213–326 ms, respectively.
- **Financial figures matched exactly.** CALCULATED A separate verification procedure recomputed every line of every finalized report in exact decimal arithmetic: 5,835,812 lines in five currencies, including currencies with 0 and 3 decimal places. It found no differences.
- **No data was lost.** MEASURED All 6,285 time entries that the application acknowledged during the load tests were present exactly once afterwards. Seventeen consistency checks found no violations on any dataset.
- **Month-end.** MEASURED 50 users were working while the monthly report generator ran and full-month exports were produced. The generator created 185 reports in under a second, and all 10 month exports succeeded.

Where the limits are

- **Annual, organization-wide administrator views grow with the total data.** MEASURED They take about 3.9 s at 1 million entries and about 16 seconds at 5 million. Team leads see similar delays on their annual views at 5 million. Both drive the slowest percentiles reported here.
- **Full-year free-text search across the whole organization** MEASURED with few or no matches took about 6 seconds at 1 million entries and exceeded the 30-second request limit at 5 million. From release 2.25, the user is then asked to narrow the search.
- **None of these timings were measured on Microsoft Azure.** INTERPRETATION They come from one local workstation. Network latency, sign-in and the cloud database tier will change the absolute numbers, in either direction.

2 Testing scope & methodology

Important. Every measurement in this report was taken in a **local, isolated test environment**, not on Microsoft Azure, using **synthetic data** and **simulated users**. The absolute timings describe that environment only. What carries over to other environments is the database work each operation needs, the correctness results, and how cost grows with data. Network round trips, identity-provider sign-in, cloud start-up and the cloud database's service tier were not part of these measurements.

Delivery model. Benedict App is licensed software that each customer installs and operates in their own Microsoft Azure subscription. Benedict Corp does not host customer installations. The results below describe the software, not any particular installation.

Environment

- **Hardware:** one desktop workstation with a 6-core / 12-thread processor, 32 GB of memory and NVMe solid-state storage.
- **Database:** Microsoft SQL Server 2025 Express Edition, running on the same machine, with 8 database schedulers and its memory fixed for repeatability. Customer installations use Azure SQL Database, at a service tier each customer chooses. The two share the same database engine but have different resource limits.
- **Application:** the same database code and the same compiled server code that the product ships, called in-process. A test harness took the place of identity-provider sign-in.

Data

- A deterministic generator built each dataset. Run twice, it produced byte-identical results.
- Each dataset spans one year and includes:
  - a seasonal peak, month-end and December peaks;
  - five currencies (including a 0-decimal and a 3-decimal currency) and rate changes over time;
  - every billing state;
  - one very large project (about 8 % of all time);
  - people joining and leaving, and closed projects.
- Time entries were loaded within every business rule, including the daily time limit per person. Monthly reports were then generated, finalized and corrected (cancelled and re-issued) through the application's own business logic.
- All names, clients and amounts are synthetic. No customer data was used.

Dataset	Time entries	People	Projects	Finalized reports	Report lines	Entries per person per week CALCULATED	Database size
~100K	99,280	120	118	942	87,517	16	0.5 GB
~500K	499,041	300	258	2,173	445,388	32	2.5 GB
~1M	999,437	500	422	3,575	884,309	38	5.1 GB
~5M	4,994,224	500	422	3,586	4,418,598	192	24.9 GB

The ~5M dataset keeps the same 500 people as ~1M, so it is five times as dense: about 192 entries per person per week. It is a deliberate stress case, well beyond typical use. It represents the volume of a much larger organization, or several years of history, concentrated into one year.

Simulated users

- **Role mix:** about 85 % team members, 10 % team leads and 5 % administrators.
- **Behavior:** each virtual user opens application pages in a realistic mix (logging and editing time, dashboards, reports, reviews) and waits 3–8 seconds between pages. Sequences are seeded, so every run repeats the same work.
- **Timing:** each run had a 60-second warm-up, followed by a 300-second measurement window.
- **The 1-user level** is a single administrator opening organization-wide views. It shows the heaviest single-user case, not a typical user.
- **Month-end run:** it added report approval, finalization and export, the scheduled monthly report generator, and back-to-back full-month exports.

Single-request measurements

For each dataset, 106 read operations (across roles, date ranges, dashboards, reports and exports) were timed three times on a warm system; the median is reported. The same run checked that each role received only the data its role permits, and that the totals were correct.

How to read the labels

- MEASURED means a value observed directly in a test.
- CALCULATED means a value derived from measured or stored data, for example by recomputation.
- INTERPRETATION means our assessment of what the results imply.

"p50", "p95" and "p99" are the response times that 50 %, 95 % and 99 % of requests completed within.

Not covered by this report

- performance on Azure SQL Database at any service tier, including network and sign-in (Benedict Corp's Azure measurements are summarized in the Benedict App Technical Review);
- load tests at the ~100K and ~500K sizes, and at 25 users;
- a repeat of the earlier version-to-version comparison on Azure (repeated locally, it showed no reproducible difference in database work);
- security testing, which is covered separately.

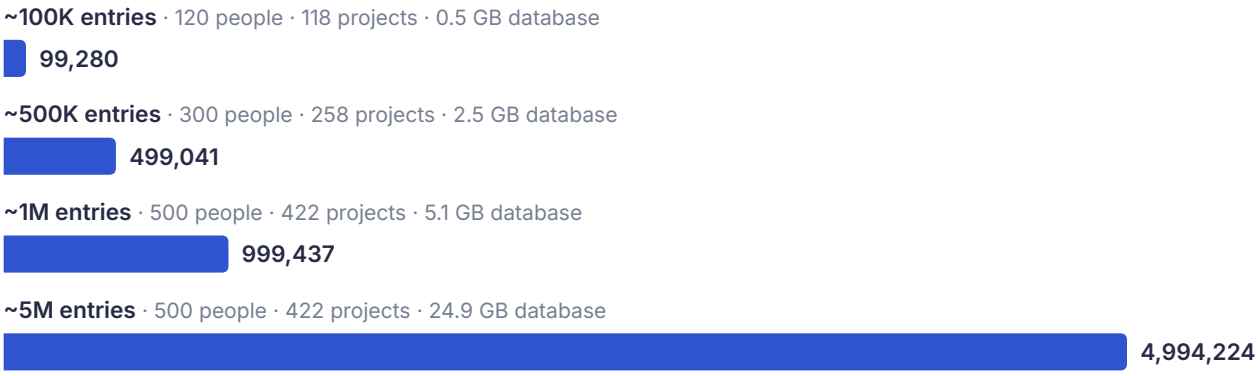
This report is not a certification, an external audit or a service-level agreement.

3 Performance at scale

Response times for individual users depend mostly on how much data *that user* works with. Only organization-wide annual views grow with the total.

Dataset sizes tested

Time entries per synthetic dataset (linear scale)



Four datasets, each spanning one year. Database size includes the indexes and the full change history.

3.1 Response time under concurrent load

Response time by number of simulated users

All requests in a 300-second measurement window · median (p50), 95th and 99th percentile · logarithmic time axis · label = p95



MEASURED 54,078 requests across the eight runs shown and the month-end run. Every run had 0 errors and 0 time-outs. The 1-user level is a single administrator opening annual organization-wide views, so it is slower than the mixed-role levels, where most requests come from team members. At ~5M, the p99 (4.1–10.5 s) comes from the annual lead and administrator views described in §6.

Dataset · users	Requests / s	p50	p95	p99	Slowest	Errors
~1M · 1 (admin)	0.6	118 ms	1.1 s	3.5 s	3.6 s	0
~1M · 10	5.1	9 ms	459 ms	1.0 s	1.4 s	0
~1M · 50	25.2	9 ms	63 ms	788 ms	2.0 s	0
~1M · 100	50.4	8 ms	65 ms	724 ms	3.3 s	0
~5M · 1 (admin)	0.4	692 ms	10.9 s	13.3 s	15.6 s	0
~5M · 10	4.6	19 ms	213 ms	10.5 s	17.4 s	0
~5M · 50	23.6	32 ms	255 ms	4.1 s	26.1 s	0
~5M · 100	47.1	44 ms	326 ms	4.3 s	25.7 s	0

3.2 Response time by role

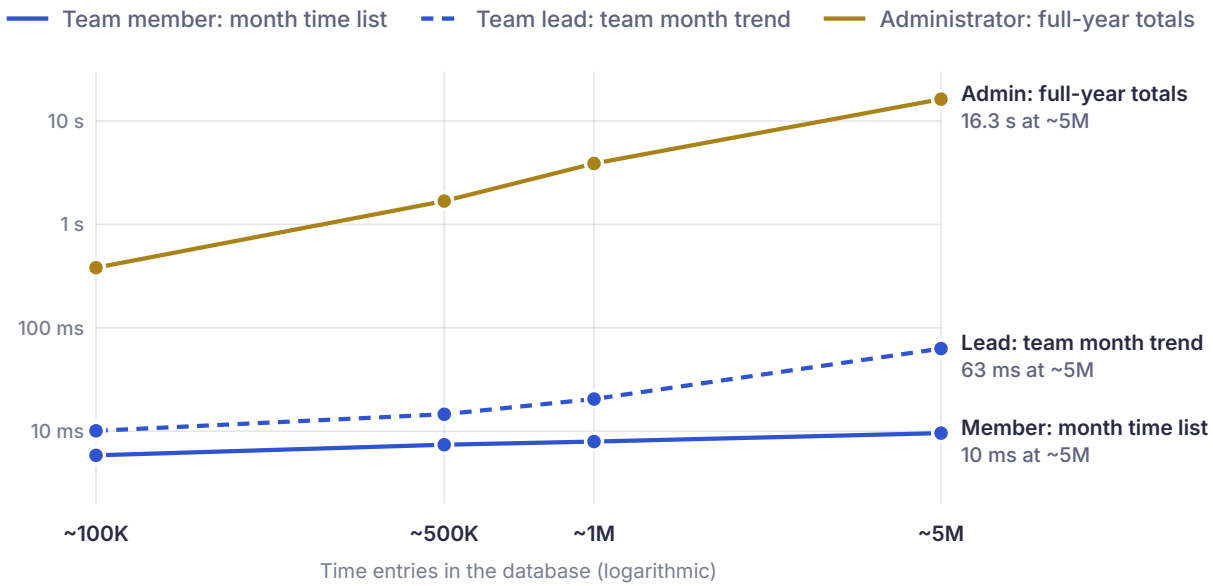
MEASURED The overall percentiles combine roles with very different work. Team members, who make most requests, stay well under half a second at the 95th percentile even at ~5M. Team leads' and administrators' annual views take seconds.

Dataset · users	Team member p50	Team member p95	Team lead p50	Team lead p95	Administrator p50	Administrator p95
~1M · 10	8 ms	16 ms	23 ms	394 ms	314 ms	1.1 s
~1M · 50	8 ms	26 ms	27 ms	80 ms	43 ms	1.1 s
~1M · 100	8 ms	31 ms	26 ms	77 ms	175 ms	966 ms
~5M · 10	17 ms	119 ms	80 ms	10.8 s	264 ms	11.7 s
~5M · 50	29 ms	132 ms	109 ms	3.3 s	459 ms	10.8 s
~5M · 100	40 ms	203 ms	141 ms	4.2 s	654 ms	12.5 s

3.3 How cost grows with data

Single-request time as the dataset grows

Median of three timed requests per dataset · both axes logarithmic



MEASURED Interactive views (cobalt) stay in the tens of milliseconds from ~100K to ~5M entries. The administrator's full-year, organization-wide totals (brass) grow roughly in proportion to the total data: 381 ms at ~100K, 1.7 s at ~500K, 3.9 s at ~1M, 16.3 s at ~5M. These single-request runs were taken before the correction in §3.4. That correction does not affect the operations shown: their database work was identical in both builds wherever both were measured.

Operation	~100K	~500K	~1M	~5M
Team member: month time list	6 ms	7 ms	8 ms	10 ms
Team lead: team month trend	10 ms	15 ms	20 ms	63 ms
Administrator: full-year totals	381 ms	1.7 s	3.9 s	16.3 s

3.4 A scaling defect found and corrected

MEASURED The tests revealed an efficiency defect. Some team-member and team-lead views did work in proportion to the whole organization's data, not the user's own. The results were always correct; only the cost was wrong. At ~5M entries, a team member's full-year text search exceeded the request limit.

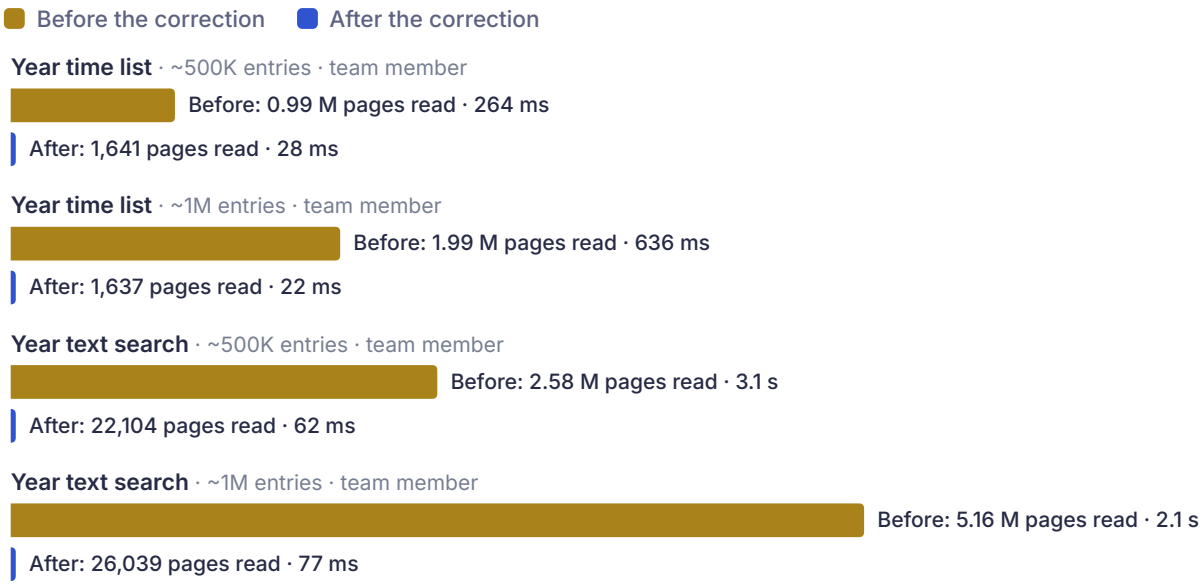
The correction changed only how the database locates the data:

- All 198 compared responses were byte-for-byte identical before and after the change.
- The full automated database test suite passed.

The correction is part of release 2.25; customers receive it when they install or upgrade to that release.

Effect of the correction on a team member's full-year views

Database pages read per request (linear scale), with the measured time · same data, same requests, before and after



MEASURED Database work fell by more than 99 % on every operation shown. Database work is deterministic; times on a single workstation vary from run to run. All load tests in §3.1 and §5 used the corrected version.

Operation	Before: pages read	After: pages read	Before: time	After: time
Year time list · ~500K entries · team member	992,991	1,641	264 ms	28 ms
Year time list · ~1M entries · team member	1,991,884	1,637	636 ms	22 ms
Year text search · ~500K entries · team member	2,578,818	22,104	3.1 s	62 ms
Year text search · ~1M entries · team member	5,158,969	26,039	2.1 s	77 ms

4 Financial accuracy & data integrity

Speed only matters if the numbers are right. We checked correctness independently of the application's own calculations.

5,835,812
report lines recomputed, including superseded versions

0
differences in quantity, rate or amount

5
currencies, including 0- and 3-decimal currencies

4 × 17
consistency checks run on each dataset: 0 violations

Independent recomputation of every report line

CALCULATED For each dataset, a separate verification procedure, written independently of the application's billing code, recomputed every line of every finalized report. That includes earlier versions that were cancelled and re-issued, whose lines are kept. It used exact decimal arithmetic (no floating point) and compared:

- billable quantity;
- the rate in effect on the work date;
- the amount, rounded to the currency's decimal places;
- report totals.

It found 0 differences across 5,835,812 lines (~100K: 87,517; ~500K: 445,388; ~1M: 884,309; ~5M: 4,418,598).

Consistency checks

MEASURED After each dataset was built, and again after the load tests, 17 automated checks ran against the stored data. They covered, among other things:

- the daily time limit per person;
- report headers equal to the sum of their lines;
- amounts equal to the rounded quantity × rate;
- finalized entries matching their report lines;
- exactly one live version per project and month;
- no duplicate submissions;
- no transactions or locks left behind;
- the database's own constraint verification.

All checks reported 0 violations on every dataset.

Access scope and totals

MEASURED In the 106 read operations per dataset, the test compared what each role received with what that role is permitted to see, and compared the totals shown with totals computed directly from the data. Across the four datasets, it found 0 scope violations and 0 total mismatches.

At ~5M, 100 of the 106 operations completed. The other six were stopped or refused, so nothing was returned to check:

- 3 full-year text searches exceeded the request limit (two of them before the correction in §3.4);
- 3 exports were refused as too large (§6).

No lost or duplicated saves

MEASURED Every time entry the application acknowledged during the nine load runs (6,285 in total) was afterwards present exactly once: 0 lost, 0 duplicated. No open transactions or held locks remained after any run.

---

5 Concurrent workloads & month-end

Month-end is the busiest moment for a time-and-billing system. People catch up on time entries while leads review, administrators close and finalize reports, the monthly report generator runs, and exports are produced. We ran all of that at once on the ~5M dataset.

50
simulated users working through the 300-second window

14 ms / 287 ms
median / 95th-percentile response

185 in 0.8 s

missing monthly reports created by the scheduled generator · 0 failed

10 / 10

full-month exports succeeded (374 reports each, median 39 s)

What happened during the run

- MEASURED 7,013 requests: median 14 ms, p95 287 ms, p99 2.8 s. The slowest single request took 9.3 s. There were 0 errors and 0 time-outs.
- MEASURED Administrators approved, finalized and exported 33 monthly reports. Finalization took a median of 211 ms, and every one succeeded.
- MEASURED The scheduled monthly report generator created the 185 missing reports for the closed month in 0.77 s. It skipped the 189 that already existed, and nothing failed.
- MEASURED An administrator exported a full finalized month (374 reports, delivered in batches as the product does) ten times back to back. All ten succeeded, with a median of 38.6 s and a slowest of 43.7 s.
- MEASURED All 798 time entries acknowledged during the run were present exactly once afterwards.

INTERPRETATION The month-end workload ran alongside the scheduled generator and exports without errors or lost work. Its response percentiles were in line with ordinary 50-user use on the same dataset (p95 255 ms).

6 Known limitations

These are the limits we observed or could not test. None caused incorrect results, but some affect how the product feels at very large scale.

Annual, organization-wide aggregations

MEASURED Year-long totals, trends and breakdowns over the whole organization read all of that year's data. Their cost grows in proportion to the total, as the table shows. At the 5-million density, team leads' annual views of their teams can also take more than ten seconds (team-lead p95 of about 11 s with 10 users).

Administrator operation (single request, median)	~100K	~500K	~1M	~5M
Full-year totals	381 ms	1.7 s	3.9 s	16.3 s
Full-year trend by month	324 ms	1.5 s	3.1 s	14.1 s
Full-year breakdown by project	340 ms	1.5 s	3.2 s	14.3 s

INTERPRETATION Possible mitigations, if customers of this size appear:

- pre-computed monthly totals (finalized months do not change);
- caching of finalized-month sums;
- shorter default ranges for organization-wide views.

These are options, not commitments.

Full-year free-text search

MEASURED A search that finds matches early returns its first page quickly: 41–50 ms for an administrator at every dataset size. A full-year search of the whole organization with few or no matches has to read every description in that range:

- about 6.0 s at ~1M entries;
- longer than the 30-second request limit at ~5M.

The request is then stopped. From release 2.25, the user is told to narrow the date range or add a filter (previously the message only said to try again). A dedicated text index, or a required filter for year-wide searches, would remove this limit.

Export limits

By design:

- a single time export holds at most 100,000 entries;
- a month export is delivered in batches of at most 300 reports;
- a single report of more than 50,000 lines is exported on its own rather than inside the month archive.

MEASURED At ~5M, these limits were reached and handled as designed:

- An administrator's month-wide time export (more than 100,000 entries) was refused with a clear "too large" message.
- So was the peak-month export batch that contained a report of more than 50,000 lines.
- That report (65,269 lines) exported on its own in 5.2 s.

INTERPRETATION Customers of this size would need higher or split limits, or per-project batch export.

Storage

CALCULATED The database used about 5.0 GB per million time entries, including the complete change history.
MEASURED The change history was about two-thirds of the total at ~5M, because every change is kept in full.

INTERPRETATION Large deployments will need a retention or archiving policy for history, and a database tier sized accordingly.

Infrastructure dependence

- INTERPRETATION All timings come from one local workstation. On Azure, absolute times will differ, in either direction, with:
  - the database service tier;
  - network distance;
  - sign-in;
  - cold starts: a paused serverless database resuming, or an application instance starting.
- These local results were not repeated on Azure. On Azure, Benedict Corp measured release 2.21 with up to 75 simultaneous simulated users (a synthetic 300-person organization) and release 2.26 with up to 20, with no errors in any run; the Benedict App Technical Review gives the details. Customers choose their own database tier; the sizing guidance supplied with the software explains how.
- The workload was synthetic. Real organizations will have different data distributions and usage patterns.
- The tail percentiles (p99) at ~5M were several seconds, driven by annual lead and administrator views.

---

7 Conclusion

INTERPRETATION In these tests, Benedict App did three things at once from about 100 thousand to about 5 million time entries:

- it kept everyday work fast;
- it kept every finalized financial figure exactly right;
- it lost or duplicated no acknowledged work under concurrent load, including a full month-end close.

The costs that grow with data are concentrated in a few organization-wide annual views and in full-year text search. We have documented them above, together with the mitigations available.

The tests also found and corrected one efficiency defect, which now has an automated safeguard.

Two things remain open:

- These results describe a local test environment. Performance in a customer's Azure installation depends on the tiers the customer chooses; Benedict Corp's Azure measurements are summarized in the Benedict App Technical Review.
- This report is not a service-level commitment.

Glossary

- **p50 / p95 / p99**: the time within which 50 %, 95 % or 99 % of requests completed.
- **Simulated user**: a program that uses the application like a person, with pauses between pages.
- **Pages read**: units of database work (8 KB data pages read from memory or disk). For identical data, this is a deterministic measure of the work a request needs.
- **Finalized report line**: a time entry fixed into a monthly client report with its quantity, rate and amount.

© 2026 Benedict Corp. Benedict App performance report, version 1.1 (November 2026; measurements of October 2026). Measurements from a local, isolated test environment with synthetic data; not a service-level agreement. Microsoft, Azure and SQL Server are trademarks of the Microsoft group of companies.