

WHITE PAPER

Databricks to Excel:

Exponam.Connect vs. the Databricks Excel Add-in

A comprehensive comparison across installation, architecture, performance, cost, and capabilities

Exponam, LLC · March 2026

Executive Summary

On March 2, 2026, Databricks released its own Excel Add-in into Public Preview — a direct, purpose-built connector from the Databricks platform to Microsoft Excel. The release is a meaningful signal: it confirms that governed, business-user access to Databricks data from Excel is a strategically important capability, not an edge case.

Exponam.Connect has occupied this space for three years of production enterprise deployment. This paper provides a direct, technically accurate comparison of the two products — what each does, how each works, where each performs well, and where each falls short. The comparison is honest: this is not a claim that Databricks' product is without merit, but rather a precise assessment of how the two differ in architecture, experience, and outcome.

The core finding: Exponam.Connect and the Databricks Excel Add-in are closer in philosophy than the earlier ODBC-based approach ever was — both provide GUI-driven, no-SQL-required data access from Excel with Unity Catalog governance. The meaningful differences are in technical architecture, installation experience, data volume, performance, cost structure, and platform depth. On the dimensions that matter most to enterprise deployments at scale, Exponam.Connect holds significant advantages. On one dimension — the question of whether users must edit XML manifest files and navigate corporate IT policy restrictions to complete installation — the Databricks product fails even technically capable users in practice.

Dimension	Exponam.Connect	Databricks Excel Add-in
Product maturity	3 years in enterprise production	Public Preview — released March 2, 2026
Installation	Web add-in via installer link at exponam.com ; enhanced VSTO edition via direct installer (Windows)	Requires downloading and editing an XML manifest file, sharing a local folder, configuring Trust Center — 10+ steps; fails for many users in corporate environments
Connection architecture	User choice: Delta Sharing (direct Parquet from cloud storage — zero DBUs, maximum speed) or SQL warehouse endpoint (full SQL syntax, compute-based). Both Unity Catalog governed.	SQL warehouse endpoint only. Every data pull executes against and consumes Databricks compute.
Client framework	VSTO (Windows) — native Office integration, full performance. Office Web Add-in (Mac/browser).	Office Web Add-in only — web-based task pane on all platforms including Windows.
Data volume to Excel	Up to 10,000,000 rows across 10 sheets (licensed)	Not published; constrained by SQL warehouse query limits and Excel memory
Performance (1M rows)	~11 seconds. Compressed Parquet; all processing client-side.	Not benchmarked publicly. SQL execution + warehouse compute + wire transfer.
Databricks compute cost	Zero DBUs when using Delta Sharing path. Standard DBU consumption when using SQL endpoint path — same as any Databricks SQL query.	DBU consumption on every import. SQL warehouse must be running.

Dimension	Exponam.Connect	Databricks Excel Add-in
Add-in license cost	Volume-tiered: \$10/user/mo (100 users) → \$0.50/user/mo (100,000 users). 100-user / \$1,000/mo minimum.	Included in Databricks subscription — but DBU costs apply to every use. Trial testing observed 16 DBU consumed in a single day of casual analyst use (\$14.58 at list rate). At 5 DBU/day average, observed costs run ~\$70/user/month — 7× Exponam.Connect pricing at 100 users, growing to 70× at 10,000.
ML model execution	Yes — run Databricks ML Serving Endpoints as Excel formulas (Exponam.AI). Windows/VSTO edition only.	No
External user access	Yes — .share files issued externally; no Databricks workspace required	No — requires Databricks workspace credentials
macOS / web support	Full on all platforms	Documented as supported; installation on macOS requires manual file-system steps
Multi-cloud roadmap	Snowflake, Azure Fabric; NL/AI query with private LLM option; AI-driven auto-optimization between Delta Sharing and SQL endpoint for cost and speed	Databricks-only

1. Background and Market Context

The Excel–Databricks gap

Microsoft Excel remains the world's dominant data platform, with over one billion enterprise users. For most of them — financial analysts, operations managers, supply chain planners, compliance teams — Excel is the environment where decisions are made and results are communicated. It is not a transitional tool. It is the destination.

Databricks has become the leading enterprise lakehouse platform, consolidating data engineering, data science, and governed analytics on Delta Lake. Enterprises have invested substantially in Databricks to manage and govern their most critical data assets.

The gap between these two dominant platforms — governed data in Databricks, decisions made in Excel — has historically produced the same failure modes everywhere it occurs: manual CSV exports that age immediately, shadow data copies that bypass governance, and BI tools that most business users lack the training or access to use effectively.

Exponam.Connect was purpose-built to close this gap, and has been deployed in enterprise production for three years. On March 2, 2026, Databricks released its own Excel Add-in into Public Preview — a signal that the market opportunity is real and strategically important. This paper examines how the two products compare.

Two architectures, different philosophies

Exponam.Connect offers user choice of connection path: Delta Sharing (an open protocol developed by Databricks for sharing Delta Lake data across organizations and tools without data movement) or a SQL warehouse endpoint for cases requiring full SQL syntax. The Delta Sharing path accesses Parquet files directly from cloud object storage via short-lived presigned URLs — no Databricks compute is involved in retrieval. The SQL endpoint path follows the same compute model as Databricks' own add-in. A private preview dynamic SQL editor provides full SQL query capability via the SQL endpoint, and upcoming AI-driven fetch optimization will automatically select the most appropriate path for each query. Both paths are Unity Catalog governed.

The Databricks Excel Add-in uses a SQL warehouse endpoint only: queries are submitted to a running Databricks SQL warehouse, executed server-side, and results returned to the add-in. This is the standard Databricks SQL query path — the same compute that powers Databricks SQL queries, notebooks, and BI tool connections. Every data import consumes Databricks Units (DBUs) proportional to query size and cluster configuration.

These two architectural choices cascade through cost, performance, governance, and total cost of ownership in ways this paper explores in depth.

2. Installation and Setup

Exponam.Connect: straightforward by design

Exponam.Connect is installed via a link provided at exponam.com — a simple, direct download path that does not require navigating a third-party marketplace. The installation process is familiar to any Windows user and requires no technical configuration.

For Windows users, an enhanced VSTO edition is available via direct installer download. VSTO (Visual Studio Tools for Office) is the native Microsoft framework for deep Office integration — it provides tighter access to the Excel object model, better performance for large data operations, and a richer UI experience than the web-based Office Add-in model. This framework choice is not incidental; it is part of what enables Exponam.Connect's performance characteristics at scale.

After installation, users authenticate via one of two methods. v1 .share credential files contain embedded bearer tokens — distributed by a data administrator — and require no further configuration. v2 .share files use full OAuth 2.0 SSO, supporting Entra ID (formerly Azure AD), Okta, and other identity providers, so users authenticate with the same enterprise credentials they use everywhere else. From that point, governed data is accessible immediately.

Typical time to first data access: under five minutes from installation to live Databricks data in Excel.

The Databricks Excel Add-in: a multi-step technical process

The Databricks Excel Add-in, released March 2, 2026, uses the Office Add-in manifest approach — a standard but not simple deployment path. The Databricks documentation specifies the following self-service steps for Windows desktop installation:

1. Download the add-in XML manifest file from Databricks documentation.
2. Edit the manifest file in a text editor to insert your Databricks workspace URL(s) inside the AppDomains section — taking care to paste without formatting (Ctrl+Shift+V).
3. Create a folder named Manifest on your local drive (e.g., C:\Manifest).
4. Copy the manifest file to that folder.
5. Open folder Properties, click Sharing > Share, and add yourself as a user with read/write permissions.
6. Open Excel Desktop, navigate to File > Options > Trust Center > Trust Center Settings.
7. In the sidebar, click Trusted Add-in Catalogs.
8. Enter the UNC path to the shared folder (e.g., \\YourComputerName\Manifest) in the Catalog URL field.
9. Click Add catalog, select Show in Menu, click OK.
10. Restart Excel Desktop.
11. Navigate to Add-ins > More Add-ins > Shared Folder, select the Databricks connector, click Add.

12. Complete sign-in to your Databricks account.

Each of these steps introduces a potential failure point. Corporate IT environments commonly restrict: local folder sharing over SMB (step 5), Trust Center modifications (step 7–9), and manifest sideloading policies. These are not edge cases — they are standard enterprise endpoint management configurations. A user who has followed every documented step correctly can still find the add-in unavailable because a Group Policy Object prevents Trust Center modifications, or because SMB local shares are disabled on managed devices.

The macOS installation path is different but not simpler: it requires locating a specific hidden Library directory, copying the manifest file there, and restarting Excel. Web installation requires navigating to a non-obvious location under Add-ins > Advanced > Upload My Add-in.

A note on practical experience.

The author of this analysis followed the complete documented installation process on a standard Windows corporate machine. Installation ultimately succeeded — but only after multiple attempts and substantial troubleshooting. The failure modes encountered (IT-managed Trust Center policies and folder sharing restrictions standard in enterprise environments) are not edge cases. They are common configurations on well-managed corporate endpoints, and they are not visible to the user until after installation has already failed. Any self-service deployment scenario for the Databricks add-in must account for meaningful remediation overhead that the documentation does not acknowledge.

3. Architecture and Client Framework

VSTO vs. Office Web Add-in: why it matters

One of the most technically significant differences between the two products is the client framework — the mechanism by which each add-in interacts with Excel.

Dimension	VSTO (Exponam.Connect Windows)	Office Web Add-in (Databricks; Exponam Mac/web)
Integration model	Native COM-based integration with the Excel object model. Direct in-process access.	JavaScript running in a sandboxed browser frame. Communicates with Excel via the Office.js API layer.
Performance for large data	Direct memory access; can write millions of rows to worksheets efficiently with minimal overhead.	Office.js API calls are asynchronous and rate-limited; large data writes incur serialization overhead through the API boundary.
UI real estate	Full Excel ribbon integration; custom ribbon tabs, task panes, and dialogs.	Task pane UI within a constrained side panel. No ribbon customization.
Excel feature access	Full access to Excel object model: PivotTables, charting, cell formatting, named ranges, events.	Access limited to what Office.js API exposes. Some advanced Excel features unavailable.
Offline capability	Full — VSTO add-in runs in-process with Excel.	Requires network connectivity to load the add-in runtime.
Platform	Windows only. (Exponam uses Office Web Add-in for Mac/browser.)	Cross-platform — same framework on Windows, macOS, Excel for web.

For Exponam.Connect, the VSTO framework is a deliberate choice, not a legacy constraint. The engineers behind Exponam.Connect were among the architects of the Microsoft Office COM integration layer — the technology underpinning VSTO. Choosing VSTO for the Windows edition means that bulk data writes, PivotTable creation, live data refresh, and the full Excel feature surface are all available with minimal intermediation.

The Databricks Excel Add-in uses the Office Web Add-in model on all platforms including Windows. This is the more modern, cross-platform Microsoft approach — and for many use cases it is entirely adequate. For high-volume data import operations, however, the Office.js API boundary introduces meaningful overhead. The tradeoff is one of breadth of platform support versus depth of performance on any single platform.

How Exponam.Connect retrieves data

Exponam.Connect offers two retrieval paths, and users choose based on their data needs. The Delta Sharing path is the default for most use cases; the SQL endpoint path is available when full SQL syntax is required.

Delta Sharing path (zero compute cost)

13. The user authenticates using either a v1 .share file (embedded bearer token issued by Unity Catalog) or a v2 .share file with full OAuth 2.0 SSO — Entra ID, Okta, or any compatible identity provider. No Databricks workspace account is required.
14. Exponam.Connect authenticates with the Delta Sharing REST API endpoint and requests data. The server returns a list of short-lived presigned URLs pointing directly to Parquet files in cloud object storage (AWS S3, Azure ADLS, or Google Cloud Storage).
15. Before downloading, Exponam.Connect validates query parameters against Delta Sharing metadata and leverages Delta Lake partition structure to minimize the data that needs to be fetched.
16. Compressed Parquet files transfer directly from cloud storage to the client — no Databricks cluster node is in the data path. All decompression and column selection processing occurs locally. All traffic is TLS-encrypted end to end.

The result: data retrieval is bounded by network bandwidth and local compute — not by SQL warehouse availability, DBU budget, or cluster auto-start latency.

Delta Sharing prerequisite.

The Delta Sharing path requires that Delta Sharing be enabled in the enterprise's Databricks environment. Most enterprises with current Databricks deployments have this enabled, but it is subject to organizational approval processes and is not universally active. Where Delta Sharing has not yet been approved internally, Exponam.Connect's SQL endpoint path provides an alternative that does not depend on it.

SQL endpoint path (full SQL syntax; compute consumed)

17. The user authenticates via SSO to their Databricks workspace using their standard enterprise credentials.
18. Queries — including complex SQL with joins, subqueries, and aggregations — are submitted to a SQL warehouse endpoint and executed server-side. A dynamic SQL notepad editor (in private preview) provides a full SQL authoring interface within the add-in.
19. Results are returned via the SQL warehouse API and written into Excel. Compute (DBUs) is consumed proportionally to query size and warehouse configuration, on the same basis as any Databricks SQL query.

The SQL endpoint path gives users who need advanced query syntax — arbitrary joins, window functions, subqueries — full parity with anything achievable directly in Databricks SQL.

Coming: AI-assisted query and automatic path optimization.

On Exponam's near-term roadmap: natural language / AI query capability allowing users to describe their data need in plain English and receive governed, reproducible results — including a private/BYO LLM option for regulated environments. Alongside this, AI-driven fetch optimization will automatically select between Delta Sharing and the SQL endpoint on each query, optimizing for cost and performance without requiring the user to choose. Both capabilities are in active development.

How the Databricks Excel Add-in retrieves data

The Databricks add-in routes all queries through a SQL warehouse — there is no zero-compute retrieval path:

20. The user authenticates via SSO to their Databricks workspace. Unity Catalog must be enabled and a SQL warehouse must be available and either running or able to auto-start.
21. Table browsing and data import queries are submitted to the SQL warehouse endpoint and executed server-side. Users can also write queries directly using the `DATABRICKS.SQL()` cell function.
22. Results are returned to the add-in via the SQL warehouse API, then written to Excel through the Office.js layer.

This path has two cost centers that do not exist in Exponam.Connect's Delta Sharing architecture: SQL warehouse compute consumption (DBUs) on every query, and the Office.js API serialization overhead on the write path into Excel. DBU consumption applies regardless of query size or frequency — there is no zero-compute access mode.

4. Performance and Data Volume

Exponam.Connect benchmarks

Exponam has published performance benchmarks against the Open Delta COVID-19 NYT dataset — a publicly available Delta Sharing dataset. Results on a standard Windows PC with Excel for Microsoft 365:

Record count	Time to import into Excel
100,000	1 second
500,000	5 seconds
1,000,000	11 seconds
1,111,930 (full dataset)	11 seconds

These figures reflect the combined effect of: compressed Parquet file transfer (significantly smaller on the wire than equivalent row-format data), client-side processing on VSTO's direct Excel object model access, and Delta Lake partition optimization that limits the data fetched to what is actually needed.

As a reference point: the Exponam comparison document notes that one million rows of narrow data takes approximately 11 seconds via Delta Sharing versus over 100 seconds via the legacy ODBC approach — a 10x advantage. For wider datasets (100+ columns), the gap extends to 50x or more, because Parquet's columnar compression advantage is most pronounced with wide schemas.

Head-to-head trial: observed results

In direct testing conducted during the preparation of this paper, both products were evaluated against a large enterprise-scale dataset. Both the Delta Sharing path and the SQL endpoint path of Exponam.Connect successfully retrieved the full dataset. The Databricks Excel Add-in returned a partial result followed by an un-descriptive error.

Product / path	Rows returned	Outcome
Exponam.Connect — Delta Sharing	2,879,789	Complete. Full dataset retrieved successfully.
Exponam.Connect — SQL endpoint	2,879,789	Complete. Full dataset retrieved successfully.
Databricks Excel Add-in	948,650	Incomplete. Retrieval stopped at approximately one-third of the full dataset and returned a non-descriptive error with no recovery path offered.

* Dataset used in testing: a large-scale enterprise transaction table available as a Delta Sharing trial dataset. The dataset contains approximately 2.88 million rows of sales transaction data across multiple columns. Results reflect a single test run on standard hardware; individual results may vary by network conditions, warehouse configuration, and add-in version.

The Databricks add-in error was non-descriptive and offered no guidance on whether the failure was due to a row limit, a memory constraint, a query timeout, or a network issue. This is consistent with the Office.js API's memory behavior with large result sets, but no documentation was found specifying a row limit for the add-in as of the March 2026 Public Preview release.

Databricks Excel Add-in performance

The Databricks Excel Add-in has no published performance benchmarks as of this writing. Direct trial testing during the preparation of this paper produced the results shown in the head-to-head table above. Beyond the row-count limitation, the architecture imposes additional latency characteristics not present in the Exponam.Connect Delta Sharing path:

- SQL warehouse query latency includes: cold-start time if the warehouse is stopped (typically 2–5 minutes for non-serverless warehouses, shorter for Serverless), query planning and execution time, and result serialization.
- Office.js API write overhead: transferring large result sets into Excel via the JavaScript API layer involves serialization at the add-in boundary that does not exist in VSTO's direct COM access.
- Network path: results travel from Databricks compute nodes via the warehouse API, whereas Exponam.Connect transfers Parquet directly from cloud storage — a path that is typically both faster and cheaper per byte.

For small, well-filtered queries on a warm warehouse, the performance difference may be modest. For large-volume imports the structural architecture differences are decisive — as the trial results above demonstrate.

Data volume limits

Exponam.Connect: licensed users can import up to 10,000,000 rows into Excel across 10 worksheets — the practical ceiling of what Excel can hold as a data surface. Beyond that, users can write unlimited datasets directly to CSV or Exponam's .BIG compressed format from within Excel. Both the Delta Sharing and SQL endpoint paths successfully retrieved the full 2,879,789-row trial dataset with no errors.

Databricks Excel Add-in: the documentation does not specify a row limit. Trial testing returned 948,650 rows from a 2,879,789-row dataset before producing a non-descriptive error — approximately one-third of the full dataset. Whether this reflects a hard limit, a memory constraint, or a query timeout behavior is not documented. Organizations planning to use the add-in for large-scale data retrieval should validate against their specific dataset sizes before deployment.

5. Cost Structure

Exponam.Connect licensing

Exponam.Connect uses volume-based per-user pricing with a 100-user minimum (\$1,000/month). Pricing scales down substantially with user count:

Users (from)	\$ / user / month
100	\$10.00
1,000	\$5.00
10,000	\$1.00
50,000	\$0.75
100,000	\$0.50

Minimum commitment: 100 users / \$1,000 per month. Enterprise terms available for large deployments.

Databricks compute costs: observed and projected

The Databricks Excel Add-in is included in Databricks subscriptions — there is no incremental add-in license fee. This makes the direct licensing comparison appear straightforward: Exponam.Connect has an explicit per-user fee; the Databricks add-in does not.

This framing is incomplete. Every query the add-in executes against a SQL warehouse consumes DBUs. During trial testing conducted for this paper, casual add-in use by a single analyst over the course of one working day — running four to five query imports against a SQL Serverless warehouse — generated 16 DBUs of consumption, equating to \$14.58 at the \$0.70/DBU Serverless list rate. This was not stress testing; it reflects light, representative analyst activity.

Observed trial consumption: 16 DBU / \$14.58 for one analyst, one day, 4–5 query imports.

Extrapolating conservatively: 5–10 DBU per analyst per day is a reasonable working assumption for moderate add-in use. At 5 DBU/day and \$0.70/DBU (a favorable Serverless rate), that is \$3.50/analyst/day — approximately \$70/analyst/month on a 20-working-day basis. At 10 DBU/day the figure doubles to \$140/analyst/month. These are deliberate query costs only; formula recalculation events in shared workbooks can add further unbudgeted consumption on top.

For context, Databricks reported \$5.4B ARR for 2025. Internal Databricks figures place effective blended revenue at approximately \$150 per active user per month across the full platform — a figure that encompasses data engineering, ML, and SQL analytics workloads, not the Excel add-in alone. The observed trial data suggests that even modest Excel add-in use by business analysts adds meaningfully to that meter.

DBU pricing varies by warehouse type, cloud provider, and contract terms:

- Serverless SQL warehouses: generally \$0.70+ per DBU at list price; enterprise contracts reduce this but do not eliminate it
- Pro/Classic SQL warehouses: lower per-DBU rate but require active idle-time management; warehouses left running consume DBUs continuously
- Auto-start latency: on-demand warehouses that are stopped incur startup time (2–5 minutes for non-serverless) and the first DBU consumption before any data is returned

The cost gap at scale: a worked comparison

The following comparison uses the observed trial rate of 5 DBU/analyst/day as its baseline — a figure well within the range of light-to-moderate add-in use. At \$0.70/DBU Serverless list pricing, this produces \$70/analyst/month. The table also shows a \$25/user/month scenario as a further conservative floor.

User count	Exponam.Connect (\$/user/mo)	Exponam total (\$/mo)	Databricks DBU @ \$25/user/mo †	Databricks DBU @ \$70/user/mo ‡
100 users	\$10.00	\$1,000	\$2,500	\$7,000
1,000 users	\$5.00	\$5,000	\$25,000	\$70,000
10,000 users	\$1.00	\$10,000	\$250,000	\$700,000

† Conservative floor estimate. ‡ Based on observed trial consumption of 5 DBU/analyst/day at \$0.70/DBU (SQL Serverless list rate) × 20 working days. Actual costs depend on query frequency, warehouse type, dataset size, and contracted DBU rates. Formula recalculation events in shared workbooks can add further unbudgeted consumption not reflected in these figures.

At the observed trial rate of \$70/user/month, Exponam.Connect is 7× cheaper at 100 users, 14× cheaper at 1,000 users, and 70× cheaper at 10,000. Even the conservative \$25/user floor produces multiples of 2.5×, 5×, and 25× respectively. And these figures cover only deliberate, user-initiated queries. The Databricks cell function model means formula recalculation events in shared workbooks can generate additional untracked warehouse queries that never appear in any usage report until the invoice arrives.

The real cost question.

Exponam.Connect pricing is transparent, volume-tiered, and fixed: \$10/user/month at 100 users, scaling to \$0.50/user/month at 100,000. Every dollar of that cost is visible, predictable, and budgetable in advance. The Databricks add-in has no license line item — but trial testing observed \$14.58 in DBU consumption from a single analyst's casual use in a single day. Extrapolated across a user base at scale, that consumption compounds quickly. The absence of a license fee is not the same as the absence of a cost.

Hidden costs: installation failure and IT support

The total cost picture must include the indirect costs of deployment friction. Every installation support ticket costs \$25–\$75 at typical enterprise IT rates. In a corporate environment where Trust Center modifications

and SMB local shares are restricted by policy — which describes a substantial majority of well-managed enterprise Windows deployments — a meaningful percentage of self-service Databricks add-in installations will fail and require IT intervention.

Exponam.Connect's direct installer from exponam.com eliminates this category of cost entirely. Installation requires no Trust Center configuration, no manifest files, and no folder sharing.

6. Usability and Feature Comparison

Core data access

Capability	Exponam.Connect	Databricks Excel Add-in
Browse available tables	Visual tree of shares, schemas, tables in ribbon task pane	Table browser within task pane; Unity Catalog governed
Filter rows	GUI column filters; date, value, quick right-click filter	GUI filters documented; SQL query option also available
Select / reorder columns	Checkbox + drag in visual preview	Column selection documented
SQL query option	Dynamic SQL notepad editor in private preview — full parity with Databricks SQL endpoint approach	Yes — optional SQL query editor, plus DATABRICKS.SQL() and DATABRICKS.Table() cell functions
Saved favorites	Star icon; one-click reuse from ribbon	Not documented in current preview
Refresh model	No formula exposed in cells; no unintended refreshes. Exponam manages refresh independently: auto-refresh at any frequency, manual refresh, pause/resume — all from the ribbon.	Cell functions (=DATABRICKS.Table / =DATABRICKS.SQL) written into cells. Excel formula recalculation (worksheet or workbook) can trigger re-execution and corresponding SQL warehouse charges.
PivotTable auto-creation	Checkbox at import time — one step during the import workflow	Yes — Pivot Data checkbox during table import; pivot-to-new-sheet only
Export to CSV	Unlimited size (licensed)	Not available in current product
Export to .BIG format	Yes — Exponam compressed format for large data	—
ML model execution	Exponam.AI: run Databricks ML Serving Endpoints as Excel formulas. Windows/VSTO edition only.	No
Metric Views support	Yes — Databricks Metric Views accessible via SQL endpoint path	Yes — import from Databricks Metric Views
UI / experience design	Designed by business users for business users. Controls and filters mirror Excel's own conventions — same gestures, same mental model as native Excel filtering. No new vocabulary or training required for typical business users.	Designed by technologists. Even the basic table-select workflow resembles a BI platform's report configuration UI — parameter-driven and query-centric. Comfortable for technical users; carries a meaningful learning curve for typical Excel business users.

A fair note on the Databricks add-in comparison: as a Public Preview product, its full feature set is not yet documented. Several entries above may represent features in development rather than permanent gaps. The comparison reflects documented capabilities as of the March 2026 launch.

Refresh model: a structural difference

The two products handle data refresh in fundamentally different ways, with meaningful implications for workbook governance and unexpected compute costs.

The Databricks Excel Add-in uses cell-based custom functions. Imported data can be fetched via the `=DATABRICKS.Table()` and `=DATABRICKS.SQL()` functions written directly into worksheet cells. This places live query logic inside the workbook itself. While Databricks' documentation states that data does not auto-refresh on workbook open, cell functions of this type are subject to Excel formula recalculation events — triggered by actions such as `Ctrl+Alt+F9`, certain workbook operations, and third-party add-ins that force recalculation. Each recalculation executes a SQL warehouse query and consumes DBUs. In shared workbook environments, a recipient refreshing the workbook uses their own Unity Catalog credentials, and the query fires against a running warehouse. The query logic — including the full SQL text — is also visible to anyone with edit access to the workbook, regardless of whether they have underlying data permissions.

Exponam.Connect does not expose any formula or query logic in worksheet cells. Data is imported into cells as static values. Refresh is managed entirely through Exponam's own mechanism, accessed from the Excel ribbon: users can configure automatic refresh at any desired frequency, trigger an immediate manual refresh, or pause and resume refresh for any dataset. A workbook refresh, formula recalculation, or any other standard Excel event has no effect on whether data is re-fetched from Databricks. There are no unexpected compute charges, no accidental re-executions, and no query logic visible in the workbook.

This difference matters most in shared workbook scenarios — the common case in enterprise finance, operations, and planning environments where a workbook is built by one person and used by many. With Exponam.Connect, the data in the workbook refreshes only when the creator or an authorized user explicitly requests it. With the Databricks cell function approach, any user with the workbook and a valid Databricks account can trigger a warehouse query.

Governance and access control

Both products support Unity Catalog governance. This is a meaningful area of parity — both products integrate with Databricks' enterprise data governance framework, enabling per-user and per-group access control, credential revocation, and audit trails.

Governance dimension	Exponam.Connect	Databricks Excel Add-in
Governance framework	Unity Catalog — via Delta Sharing recipients (DS path) or directly via workspace credentials (SQL endpoint path). Both paths fully governed.	Unity Catalog directly
User credentials required	DS path: v1 .share file (embedded token) or v2 .share file with SSO — Entra ID, Okta, OAuth 2.0. No workspace account needed. SQL endpoint path: standard Databricks workspace SSO credentials.	Databricks workspace account with Unity Catalog permissions.
External user access	Yes — .share files can be issued to users outside the organization	No — requires Databricks workspace credentials

Governance dimension	Exponam.Connect	Databricks Excel Add-in
Credential revocation	Bearer token revocation via Unity Catalog immediately terminates access	Workspace account deactivation
IP-based restrictions	Supported via Unity Catalog credential properties	Workspace-level network policies
Encryption in transit	End-to-end TLS: Delta Sharing to client, cloud storage to client	TLS via SQL warehouse endpoint

Unity Catalog governance is an area of genuine parity: both products integrate with Databricks' enterprise governance framework, and neither holds an advantage on audit trails, per-user access control, or credential revocation. The meaningful distinction in governance is credential scope for users accessing via the Delta Sharing path. The Databricks add-in requires every user to have a Databricks workspace account — a meaningful provisioning overhead for large enterprise deployments, and an insurmountable barrier for external data sharing. Exponam.Connect's .share file model allows data access to be granted to any user, including partners and clients, without creating Databricks workspace accounts. For users accessing Exponam.Connect via the SQL endpoint path, the authentication model is identical to the Databricks add-in.

Exponam.AI: ML model execution from Excel

Exponam.Connect includes a capability with no equivalent in the Databricks add-in: the ability to run Databricks Machine Learning Model Serving Endpoints directly from Excel formulas. This capability is currently available in the Windows VSTO edition only and is not yet available in the Mac or Excel for the web editions.

The workflow requires no data science knowledge:

- An admin configures the Model Serving Endpoint and bearer token once in Exponam.Connect settings.
- A user-guided formula builder identifies required model input parameters.
- Users specify Excel cell references or direct values for each input.
- The model executes via Databricks Serverless clusters and returns results as a standard Excel formula output.
- Advanced users can bypass the GUI and write the model function directly into cells.

This transforms Databricks ML models — historically accessible only to data scientists with Python or notebook environments — into tools any Excel user can apply directly to their data. Credit scoring, churn prediction, demand forecasting, fraud detection: any trained model deployed on Databricks can be invoked from a spreadsheet cell.

User interface and experience design

The two products reveal their intended audiences the moment they are opened. This is a meaningful differentiator for enterprise deployments, and one that rarely appears in feature comparison matrices.

Exponam.Connect was designed by business users for business users. The ribbon integration, task pane layout, and interaction model are built to mirror what Excel users already know. Filters are applied exactly as they are on a native Excel sheet — same gestures, same visual conventions, same mental model, no new vocabulary. A financial analyst, operations manager, or supply chain planner can be productive within minutes of installation, without training or documentation. The design assumption is that Excel is the user's native environment, and the add-in should feel like a natural extension of it rather than a new application embedded within it.

The Databricks Excel Add-in was designed by technologists, and the difference is apparent. Even using the basic table-select option — the simplest path through the product — the interface looks and feels like a BI platform's report modification UI. It is parameter-driven and query-centric, structured around concepts that are second nature to a data engineer or SQL analyst but unfamiliar to the typical Excel business user. This is not a failure of the product for its intended audience: a Databricks-credentialed analyst comfortable in notebooks and SQL editors will feel at home. But that is a narrower population than the business user base that both products nominally target.

The deployment implication.

In most enterprises, the population of Excel business users outnumbers the technical Databricks user base by a wide margin. Both products target that broader population in principle. In practice, Exponam.Connect is the product those users will adopt without hand-holding. The Databricks add-in will require change management and training investment that its absence of a license fee does not account for — a hidden cost that belongs in any honest total cost of ownership analysis.

7. Decision Framework

The two products are now genuinely competitive in philosophy — both target non-technical Excel users who need governed Databricks data without SQL. The right choice depends on the specific deployment context.

Scenario	Better fit	Primary reason
Large enterprise deployment, non-technical users, Windows primary	Exponam.Connect	Simple installer from exponam.com ; VSTO performance; zero DBU cost on Delta Sharing path
Large enterprise, Mac-heavy workforce	More balanced — evaluate carefully	Both use the Office Web Add-in framework on Mac. Exponam.Connect installation is simpler than Databricks add-in but somewhat more involved than Windows. Exponam.AI and advanced SQL query (private preview) are not yet available on Mac edition. Delta Sharing cost advantage and external access remain Exponam strengths.
Large-volume data imports (>1M rows regularly)	Exponam.Connect	10M row ceiling; compressed Parquet; VSTO write performance on Windows
External data sharing (partners, clients)	Exponam.Connect	.share files work without Databricks workspace accounts
Running ML models from Excel	Exponam.Connect (Windows)	Exponam.AI (Windows/VSTO only) has no equivalent in Databricks add-in. Not available on Mac or web editions of Exponam.Connect.
Cost-sensitive; minimizing Databricks compute spend	Exponam.Connect	Zero DBUs on Delta Sharing path; fixed per-user cost; no formula recalculation risk
Small team (fewer than 100 users); Databricks-native users	Databricks add-in or evaluate carefully	Exponam.Connect's 100-user / \$1,000/month minimum is a meaningful commitment for small teams. Databricks add-in avoids additional license cost, though DBU consumption should be monitored.
Complex ad-hoc SQL required (arbitrary joins, subqueries)	Either — parity achieved	Exponam.Connect dynamic SQL editor (private preview) achieves full SQL parity via SQL endpoint. Databricks add-in uses SQL query editor and <code>DATABRICKS.SQL()</code> cell function.
Metric Views access	Either — parity	Both products support Databricks Metric Views access via SQL endpoint
Managed IT deployment, no user self-service	Either, via M365 admin deployment	Both support M365 admin-managed deployment

8. Strategic Roadmap and Positioning

Exponam's multi-cloud vision

Exponam.Connect currently serves the Databricks-to-Excel market. The company's strategic roadmap extends substantially beyond this:

Roadmap item	Description	Strategic significance
Direct SQL query (private preview)	Dynamic SQL notepad editor with full SQL support via SQL endpoint — in private preview, achieving full parity with Databricks' SQL approach	SQL parity achieved. Removes the prior capability distinction.
Natural language / AI query	LLM-assisted query: describe your data need in plain English; governed, reproducible results returned	Transforms data access for non-technical users; eliminates the SQL barrier entirely
AI-driven fetch optimization	LLM capabilities will auto-select between Delta Sharing and SQL endpoint on each query, optimizing for cost and performance	Intelligent routing means users never need to think about which path is faster or cheaper — the system decides
Private / BYO LLM	AI query processing within the enterprise security perimeter; support for user-selected LLM models	Critical for regulated industries (financial services, healthcare, government) where data cannot leave controlled environments
Snowflake expansion	Governed Excel connectivity to Snowflake data platform	Addresses the second largest cloud data platform; completes the multi-cloud analyst layer
Azure Fabric expansion	Extension to Microsoft Azure Fabric / OneLake	Targets Microsoft's own modern data platform; positions for Microsoft-first enterprise accounts

The combined roadmap — governed, multi-cloud, AI-powered Excel data access — positions Exponam.Connect not as a Databricks tool but as the enterprise analyst access layer: the neutral bridge between all major cloud data platforms and the Excel-based work that drives enterprise decisions.

This is a position no individual platform vendor can occupy without compromising their neutrality. Databricks will build a good Excel connector for Databricks data. They will not build one for Snowflake. Snowflake's connector will not cover Databricks. Microsoft Copilot assists with formulas but does not connect to either at the data layer. The independent analyst layer is structurally the only viable neutral solution — and three simultaneous market signals in late 2025 and early 2026 confirm that enterprises are actively seeking it.

The Databricks add-in as market validation

Databricks' decision to build its own Excel connector — rather than partnering with Exponam despite Exponam's Validated Technology Partner status — should be read as strong market validation, not as a competitive threat that changes the fundamental analysis.

Databricks is confirming that Excel access is strategically important to their platform story. Their connector, by architectural necessity, will be optimized for Databricks compute consumption and Databricks-only data. Its performance is bounded by the Office Web Add-in framework they have chosen. Its installation requires technical steps that fail in standard enterprise IT environments. And it will never serve Snowflake or Fabric data.

For organizations whose Databricks investment is one part of a multi-platform data environment — which describes most enterprises — the case for a platform-neutral Excel access layer is strengthened, not weakened, by the Databricks announcement.

9. Conclusions

The Databricks Excel Add-in is a meaningful new entrant. It brings a genuinely simpler experience than the legacy ODBC approach it appears to replace, and it will serve some use cases — particularly small teams of Databricks-credentialed users with modest data volume needs — reasonably well.

For enterprise deployments at scale, the structural differences in architecture, installation reliability, performance, and cost remain decisive. The Delta Sharing approach — no compute consumption, direct Parquet transfer, VSTO-accelerated writing into Excel — provides performance and cost characteristics that a SQL warehouse-backed web add-in cannot match. Trial testing observed \$14.58 in DBU consumption from a single analyst's casual daily use; extrapolated at 5 DBU/analyst/day, the Databricks add-in runs approximately \$70/user/month against Exponam.Connect's \$10/user/month at the 100-user tier — before accounting for unbudgeted formula recalculation events in shared workbooks.

The right framing for most organizations is not either/or. Exponam.Connect serves the broad enterprise population: the analysts, finance teams, and operations staff who need live Databricks data in Excel without technical barriers. The Databricks add-in may serve the narrower population of data engineers who are already operating within the Databricks workspace and occasionally need SQL-queried data in a spreadsheet.

What Databricks' entry does not change is the multi-cloud imperative. An organization with data in both Databricks and Snowflake — a common and growing configuration — still needs a single Excel access layer that works across platforms. That is not a product Databricks will build, because it cannot be in Databricks' interest to do so. It is precisely the space Exponam.Connect is built to occupy.

Appendix: Full Technical Specification Comparison

Specification	Exponam.Connect	Databricks Excel Add-in
Release status	Generally available; 3 years production	Public Preview — March 2, 2026
Connection protocol	User choice: Delta Sharing REST API (zero compute) or SQL warehouse endpoint (full SQL syntax). Both Unity Catalog governed.	SQL warehouse endpoint only (Databricks SQL API)
Data in transit	Delta Sharing path: compressed Parquet (columnar) from cloud object storage. SQL endpoint path: SQL query results via warehouse API.	SQL query results via warehouse API
Data source	Delta Sharing path: AWS S3, Azure ADLS, GCS — directly via presigned URLs. SQL endpoint path: Databricks SQL warehouse cluster.	Databricks SQL warehouse cluster

Specification	Exponam.Connect	Databricks Excel Add-in
DBU consumption	Zero for Delta Sharing path. Standard DBU consumption on SQL endpoint path.	Yes — every query consumes compute
Authentication	Delta Sharing path: v1 .share files (embedded bearer token) or v2 .share files with full SSO — Entra ID, Okta, OAuth 2.0. SQL endpoint path: Databricks workspace SSO credentials.	SSO via Databricks workspace account (Unity Catalog)
Client framework (Windows)	VSTO — native COM Office integration	Office Web Add-in (JavaScript / Office.js)
Client framework (Mac / web)	Office Web Add-in + helper application	Office Web Add-in
Max Excel rows	10,000,000 (licensed; 10 sheets)	Not published; ~1M practical ceiling
Unlimited data export	CSV and .BIG format (licensed)	Not documented
Windows installation	Installer link at exponam.com; enhanced VSTO edition via direct download	Download manifest XML → edit → create shared folder → Trust Center config → restart Excel → 10+ steps
macOS installation	Installer link at exponam.com	Locate hidden Library directory; copy manifest file; restart Excel
Corporate IT compatibility	Direct installer, no Trust Center or SMB configuration required	Requires Trust Center modification and local folder sharing — restricted by common IT policy
External user access	Yes — .share files work without workspace accounts	No — requires Databricks workspace credentials
Unity Catalog governance	✓	✓
SQL query option	Dynamic SQL notepad editor (private preview) — full parity with Databricks SQL approach; UC Views always available	SQL query editor in task pane; DATABRICKS.SQL() and DATABRICKS.Table() cell functions
Metric Views support	Yes — accessible via SQL endpoint path	✓
ML model execution	Yes — Exponam.AI (Databricks Serving Endpoints as Excel formulas). Windows/VSTO only.	No
Refresh model	Independent of Excel recalculation: auto-refresh at any frequency, manual, pause/resume from ribbon. No formula exposed in cells.	DATABRICKS.Table/SQL cell functions can be re-executed by Excel formula recalculation (worksheet or workbook), triggering SQL warehouse queries and DBU consumption.
PivotTable creation	Checkbox at import time	Yes — Pivot Data checkbox at import; new sheet only

Specification	Exponam.Connect	Databricks Excel Add-in
License cost	Volume-tiered: \$10/user/mo (100 users) → \$0.50/user/mo (100,000 users). 100-user / \$1,000/mo minimum.	Included in Databricks subscription
Compute cost	Zero on Delta Sharing path. Standard DBU consumption on SQL endpoint path.	DBUs on every import query. Blended platform revenue ~\$150/user/mo; add-in consumption is a fraction of that but non-zero and variable.
Multi-cloud roadmap	Snowflake, Azure Fabric	Databricks only
NL / AI query roadmap	Yes — with private LLM option	Not announced
Databricks partner status	Validated Technology Partner (2024)	First-party Databricks product

About Exponam

Exponam, LLC is the world's foremost firm at the intersection of modern cloud data platforms and Microsoft Office. Our engineers include the architects who built core technology upon which Microsoft Office still relies, and technologists with decades of experience spanning relational databases, data lakes, encryption, and financial services data delivery. Exponam holds Validated Technology Partner status from Databricks since 2024.

exponam.com · info@exponam.com · +1.646.360.0110

© 2026 Exponam, LLC. All rights reserved.