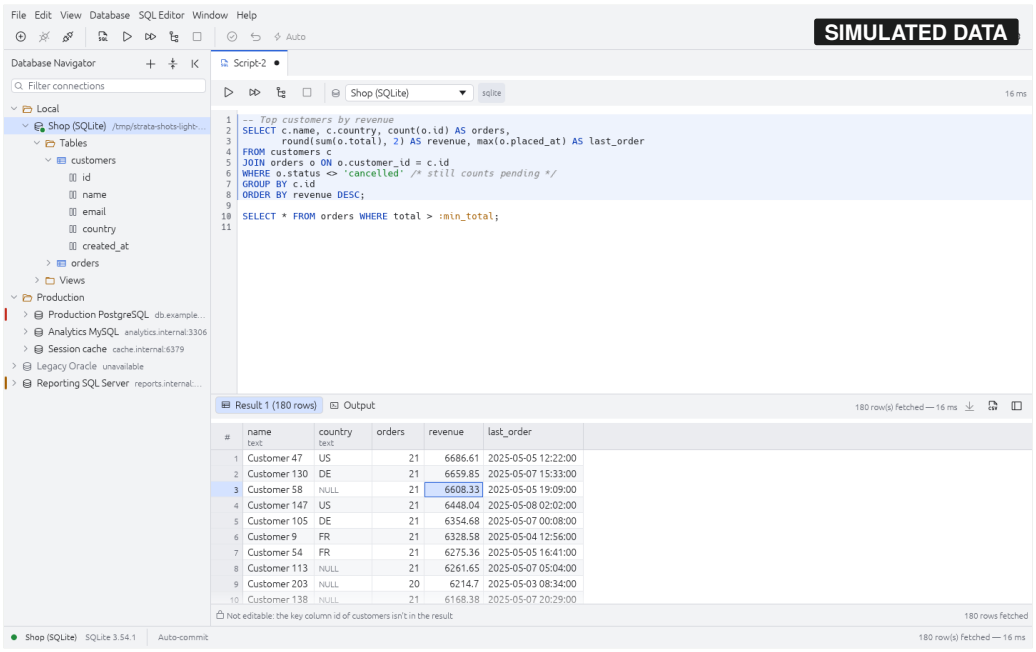
uncomplex.

strata.

User Manual

A desktop database client: connect, query, browse and edit data, with nothing else to install.



Strata's main window: the Database Navigator, an SQL editor and the results grid. Simulated data.

Guide revision	3
Date	22 November 2026
Status	Release 1.0.0
Licence	Proprietary (EULA); includes Apache-2.0 components

Read this first

Strata is a desktop database client from Uncomplex. You use it to connect to a database, browse its schemas and tables, run SQL, read and edit the results, and move data in and out as CSV. Optional feature extensions add tools for schemas, migrations, comparing and profiling data, and more. It connects to your databases directly. The program you run is called `strata`.

Four things are worth knowing before you start.

- **It asks for a licence when it starts.** Accept the terms, then activate with a licence key or start a free trial. Activation and the daily check contact Uncomplex's licensing server. Chapter 7 says exactly what is sent.
- **Which databases work depends on your edition.** The *Core* edition serves SQLite, the PostgreSQL and MySQL families, SQL Server and Redis-compatible servers. The *Complete* edition adds many more, as signed *driver extensions* installed with Strata, and the feature extensions in the **Tools** menu. **File › Extensions…** lists the extensions in your installation.
- **It reads a limited number of rows by default.** A result stops at 200 rows unless you fetch more or raise the limit. This keeps large tables from filling memory.
- **Edits are explicit.** Changes you make in the results grid are held until you choose **Save**, and then applied in one transaction.

> ⃠ **Your data** Strata runs the SQL you give it with your database account's permissions. A connection's **Read-only** option is a guard against mistakes, not a permission boundary: a session can switch it off with SQL. When it matters, connect with an account that cannot write.

Pick your path

Start at the row that matches what you want. Times are active work; downloads, installation and slow networks add to them.

I want to…	Go to	Time
Install Strata and connect for the first time	Chapter 1, p. 1	10 min
Activate a licence or start a trial	Chapter 7, p. 37	5 min
Connect through an SSH tunnel, a proxy or TLS	Chapter 2, p. 6	10 min
Protect my saved passwords	Section 2.8, p. 16	5 min
Write and run SQL	Chapter 3, p. 18	10 min
Change rows in a table	Chapter 4, p. 24	5 min
Import a CSV file, or export a result	Chapter 5, p. 28	5 min
Query CSV, JSON, Excel or Parquet files with SQL	Chapter 6, p. 32	5 min
Add a database driver extension	Section 8.2, p. 41	5 min
Diagram, design, compare or migrate schemas; compare, profile or anonymise data	Section 8.3, p. 41	10 min
Find anomalies in a result with Data Laboratory	Section 8.6, p. 46	10 min
Fix something that is not working	Chapter 9, p. 56	varies
See the keyboard shortcuts	Appendix A, p. 62	1 min
Know what Strata does not do	Appendix D, p. 68	5 min

How to use this guide

You do not need to read this guide in order. Each chapter opens with a **TL;DR** box that says what it is for. If it is what you need, carry on; if not, skip it. Each task ends with a **Done when** line, so you know when you can stop.

Markers

Marker	Means
TL;DR (black tab)	The chapter in a few lines. Read only this if you are in a hurry.
Time	Roughly how long the active work takes.
Before you begin	What you need first.
Try it	Something small to do now. Hairline frame.
Tip	Useful, but safe to skip. Hairline frame.
Careful	Something that can fail or surprise you. Solid frame with a heavy left edge.
Your data / **Security**	Data loss, exposed credentials or lost access. Heavy double frame. Read every one, before the step it precedes.
Honesty	What has not been tested, or where certainty is limited. Dashed frame.
Why this matters, **Technical detail**	Background for the curious. Grey panel.
Done when	How you know the task is finished.
Skip this if	When a section does not apply to you.

How things are written

Written as	Means
File › Extensions…	A menu, then an item in it. Labels are quoted exactly as Strata shows them, including the ellipsis.
Test connection	A button, tab, tick box or other label on screen.
`Ctrl` + `Enter`	Keys to press together. On a Mac, Strata reads `Ctrl` in most shortcuts as `Cmd` (Command); Appendix A lists the exceptions.
`strata --diagnostics`	Something to type exactly, a file name or a path.
`<data dir>`	Strata's data directory (Section 1.5). Replace it with the real path.

⚖ **Honesty** Every screenshot in this guide shows the real interface, but the names, hosts and numbers in them are invented, so each is marked *Simulated data*. Host names such as `db.example.com` are placeholders.

Contents

Read this first ii

How to use this guide iv

Strata on one page ix

1 Install Strata and make a first connection 1

1.1 Get Strata 1

1.2 Get past the licence window 3

1.3 Connect to a test database 3

1.4 Import connections from DBeaver 4

1.5 Where Strata keeps your data 4

2 Connect to your databases 6

2.1 Which databases work 6

2.2 Create a connection 9

2.3 Manage connections in the navigator 11

2.4 Reach a server through an SSH tunnel 12

2.5 Connect through a proxy 13

2.6 Encrypt and verify with TLS 14

2.7 Sign in to Amazon Web Services 15

2.8 Keep saved passwords safe 16

3 Write and run SQL 18

3.1 Open an editor 18

3.2 Run statements 18

3.3 Split, move and fold the editor 19

3.4 Statements with parameters 21

3.5 Production connections ask first 21

3.6 See how a query will run 22

3.7 Transactions 22
3.8 Tabs, history and closing 23
3.9 Non-SQL editors 23

4 Read, filter and edit results 24

4.1 The grid 24
4.2 How many rows you see 25
4.3 Copy cells 25
4.4 Look at a table’s data 25
4.5 Edit rows 26

5 Import and export data 28

5.1 Export a result to CSV 28
5.2 Copy instead of export 29
5.3 Import a CSV file into a table 29
5.4 Export your connection list 30
5.5 Time-series results 30

6 Query files with SQL 32

6.1 What you can open 32
6.2 Connect to a file or folder 33
6.3 Browse a LevelDB database 34
6.4 How types are worked out 35
6.5 Safety and limits 35

7 Activate and manage your licence 37

7.1 Activate this computer 37
7.2 The licence window’s buttons 38
7.3 What the states mean 38
7.4 What is sent, and when 39

8 Extensions 40

8.1 What an extension is 40
8.2 Driver extensions 41
8.3 Feature extensions 41

8.4 Get Data Laboratory 45
8.5 Install an extension 45
8.6 Your first analysis 46
8.7 Choose an analysis 50
8.8 Cancel, limits and messages 52
8.9 Export the results 53
8.10 What stays on your computer 53
8.11 Turn extensions on and off 53
8.12 If an extension is unavailable 54

9 When something goes wrong 56

Starting Strata 56
Connecting 57
Running SQL 58
Results and data 59
Extensions 60
9.1 Collect a diagnostics report 60

A Keyboard shortcuts 62

B Glossary 64

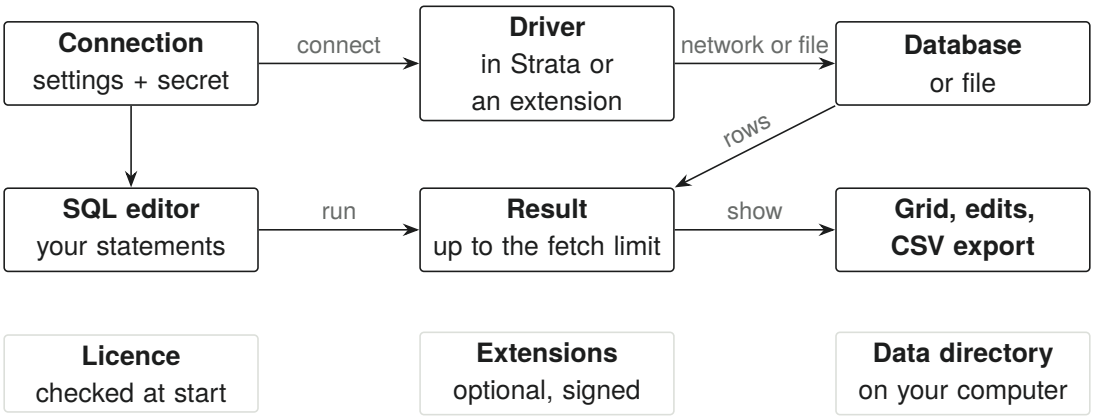
C Version and support notes 66

C.1 Which version this guide describes 66
C.2 How it was checked 66
C.3 Licence 67
C.4 Getting help 67

D Limitations 68

D.1 Honesty 68
D.2 Connections and security 69
D.3 SQL and results 69
D.4 Feature extensions 70
D.5 Files 70
D.6 Products not offered 70
D.7 Licensing 71

Strata on one page



In words: you save a **connection** (where the database is, and how to log in). Strata's **driver** for that product (part of Strata, or a driver extension) opens a session, directly or through an SSH tunnel or proxy. You type SQL in an **SQL editor**; each statement's **result** is read up to the fetch limit and shown in a **grid**, where you can copy, edit or export it. Your settings and encrypted secrets live in the **data directory**; the **licence** is checked when Strata starts; **extensions** are optional add-ons: more drivers, and tools in the **Tools** menu that work through Strata.

What you provide: a database to connect to (or a file), an account on it, the connection details, and a licence key or a trial request. For a Complete-edition product, the Complete edition or that product's driver extension.

	Strata does	**Strata does not**
Connections	Saves them without secrets in `connections.json`; keeps passwords encrypted in `credentials.v2`; imports DBeaver connections one way.	Store a password unless you tick **Save password** (SSH and proxy secrets are always stored when entered). Sync connections between computers.
Network	Connects directly, or through an SSH tunnel, SOCKS5 or HTTP CONNECT proxy, for the drivers that offer them. Verifies TLS server identities unless you choose otherwise.	Chain an SSH tunnel through a proxy, use jump hosts, or log in with keyboard-interactive or 2FA SSH.
SQL	Runs the statement at the cursor or the whole script; explains plans where the database supports it; asks before UPDATE or DELETE without WHERE on a Production connection.	Detect statements that commit implicitly on the server, such as DDL on MySQL.
Results	Reads up to the fetch limit (200 rows by default); edits single-table results that have a key, saving all changes in one transaction.	Read the rest of a large result unless you ask (**Fetch more rows**, or export).

	Strata does	**Strata does not**
Licence	Verifies a signed lease offline on every start; renews it online about once a day.	Run unlicensed. After seven days without a successful check, it asks you to connect.
Extensions	Loads only extensions signed by the Strata project, and checks each one before it runs. Shows a feature extension's statements, and every network request, and waits for you to confirm.	Download anything, sandbox an extension, or offer a marketplace.

CHAPTER 1

Install Strata and make a first connection

TL;DR

Get Strata, start it, deal with the licence window, then create a connection to an in-memory SQLite database and run one query. SQLite needs no server, so this works on any computer.

Time About 10 minutes.

Before you begin Linux, macOS or Windows on x86-64 or Apple Silicon. A network connection for the licence step.

1.1 Get Strata

Strata comes as an installer for each platform, in two editions. The *Core* edition is the program on its own. The *Complete* edition installs the same program with a set of signed extensions beside it: a driver extension for each further group of databases (Section 2.1), and the feature extensions, such as Schema Explorer and Migration Studio (Section 8.3).

Platform	What you install
Windows	An installer. For the Complete edition it lists each extension under **Extensions** as an optional component, so you can leave out what you do not need.
Linux	Packages for your distribution: one for Strata and one for each extension. Install only the extension packages you need.
macOS	`Strata.app`. In the Complete edition the extensions are inside the app, so there is nothing else to install.

You can add an extension to an installation later (Section 8.2).

What each platform needs at run time

Platform	Needs
Linux	A desktop session (X11 or Wayland) with OpenGL (EGL or GLX) and libxkbcommon. Most desktop distributions have them already.
macOS	Nothing extra.
Windows	Nothing beyond Windows itself. `strata.exe` is a windowed program with no console window.

> ⚠ **Careful** If macOS refuses to open `Strata.app` the first time, saying it cannot check the app or that it is from an unidentified developer, open **System Settings** › **Privacy & Security**, find the message about Strata near the bottom and choose **Open Anyway**. macOS asks once more; confirm, and it opens normally from then on. Do this only for a copy of Strata you got from Uncomplex.

👁 **You should see** a window titled Strata with a welcome page in the middle: **New connection…**, **New SQL editor**, **Open SQL script…** and **Database catalogue**.

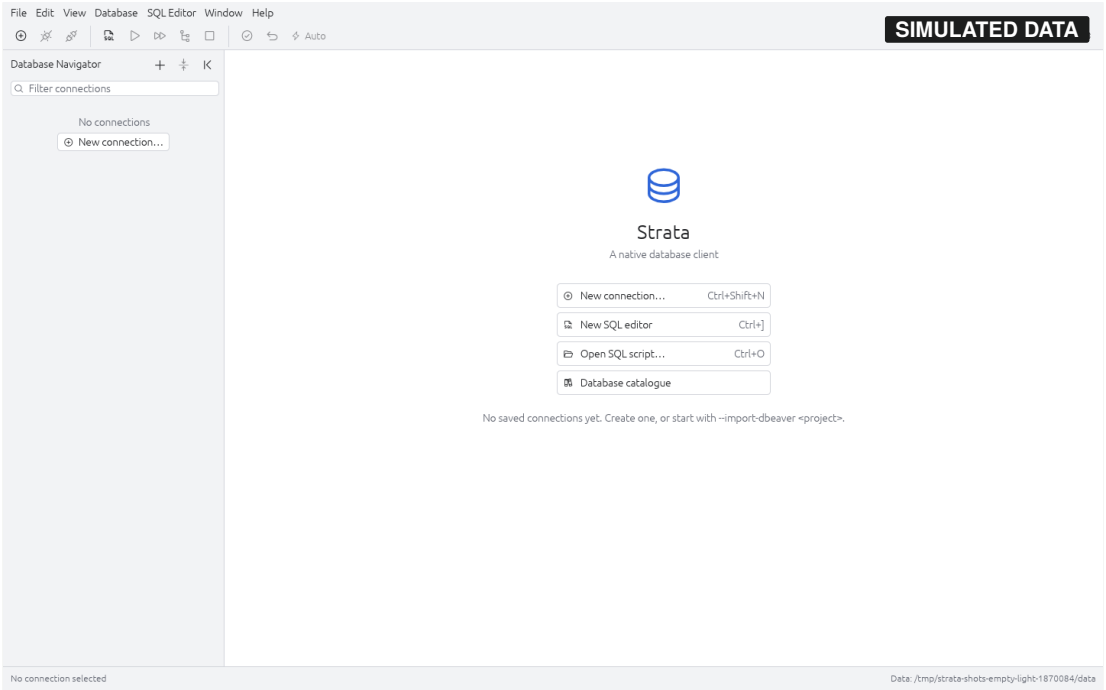


Figure 1.1. The welcome page, shown when no editor is open. *Simulated data.*

1.2 Get past the licence window

On first start, a window titled **Strata licence** covers the app until this computer is activated or on a trial. In short: tick **I agree to the licence terms**, then either paste your licence key and choose **Activate**, or choose **Start free trial**. Your browser opens to approve the request. Chapter 7 covers every state and what is sent.

1.3 Connect to a test database

1. Choose **New connection…** on the welcome page (or **File › New connection…**, or `Ctrl` + `Shift` + `N`). The **Connect to a database** window opens.
2. Type `sqlite` in **Search products**, then click the **SQLite** card.
3. On the **Main** tab, set **Name** to `Scratch` and **Database file** to `:memory:`. Leave the rest as it is.
4. Choose **Test connection**. Then choose **Save**.
5. In the **Database Navigator** on the left, double-click **Scratch** to connect.
6. Choose **File › New SQL editor** (`Ctrl` + `]`). Type:

```
SELECT sqlite_version() AS version, 6 * 7 AS answer;
```

7. Press `Ctrl` + `Enter`.

You should see a result tab under the editor, **Result 1 (1 rows)**, with `version` and `answer` columns, and `42` in the second. The status bar at the bottom names the connection and the SQLite version.

If not if the connection is greyed out in the product list, or connecting fails, see Chapter 9. An SQLite file other than `:memory:` must already exist: Strata never creates a missing database file, so a mistyped path is an error, not a new empty database.

Done when the result grid shows `42`.

Next connect to a real server (Chapter 2), or import your connections from DBeaver (below).

1.4 Import connections from DBeaver

⏩ **Skip this if** you have never used DBeaver.

Strata can read an existing DBeaver project and copy its connections, with their saved credentials, into its own store. It is a one-way import: DBeaver's files are only read.

```
strata --import-dbeaver
strata --import-dbeaver /path/to/DBeaverData/workspace6/General
```

Without a folder, Strata looks in DBeaver's default workspace. It prints a report of what it imported and what it could not.

⚠ **Careful** Connections whose driver is not in your installation are still imported. They appear in the navigator greyed, marked **unavailable**, with the reason when you hover over them. Nothing is deleted, and they work once you add the driver extension they need.

1.5 Where Strata keeps your data

Everything Strata saves is in one folder, the *data directory*:

Platform	Default data directory
Linux	`~/.local/share/strata` (or `$XDG_DATA_HOME/strata`)
macOS	`~/Library/Application Support/Strata`
Windows	`%APPDATA%\Strata`

File or folder	Holds
`connections.json`	Folders and connection settings. Never passwords or other secrets.
`credentials.v2`	Passwords and other secrets, encrypted and authenticated.
`credentials.key`	The encryption key, only when no OS keyring is available.
`settings.json`	Theme, query limits, window and panel sizes, disabled extensions.
`known_hosts`	SSH host keys you chose to trust in Strata.
`extensions/`	Extensions you installed, one folder each.
`extension-data/`	What feature extensions save, such as snippets, workspaces and schema snapshots, one folder per extension.
`licensing/`	This computer's activation and its signed lease.
`logs/strata.log`	The log, rotated at 1 MiB.

> **Tip** `--data-dir <dir>` uses another folder for one run, which is handy for a test setup. If the folder is read-only, Strata says so in a banner and applies preferences to that session only.

Command-line options

Option	Default	Meaning
`--data-dir <dir>`	see above	Where connections, credentials and settings live.
`--import-dbeaver [<dir>]`	DBeaver's default	One-way import of DBeaver connections.
`--fetch-size <rows>`	100	Rows fetched per round trip.
`--fetch-limit <rows>`	200	Rows kept per result; the rest is not read.
`--display-limit <rows>`	200	Rows shown in the grid.
`--export-limit <rows>`	none	Rows written by CSV export.
`--query-timeout <s>`	none	Seconds before a statement is cancelled.
`--diagnostics`		Prints a support report and exits. Writes nothing.
`--with-hosts`		With `--diagnostics`: include host names. Passwords written into a connection URL are still left out.

The limits can also be set in **File › Settings…** (`Ctrl` + `,`), on the **Query limits** page. An option given on the command line wins for that run.

CHAPTER 2

Connect to your databases

TL;DR

Choose **New connection…**, pick the product, fill the **Main** tab and choose **Test connection**, then **Save**. Use the **SSH** or **Proxy** tab to reach a server behind a bastion, and the **SSL / TLS** tab to encrypt and verify the connection. Passwords are encrypted on disk; you can protect them with a password of your own.

Time 5 minutes per connection; longer the first time you set up a tunnel or certificates.

Before you begin the server's host name and port, a database account, and, for TLS, any CA or client certificate files.

2.1 Which databases work

Strata knows about more products than it can connect to. Two things decide whether a product works for you: whether your installation has its driver, and how far that driver has been *tested*.

Editions and driver extensions

Edition	Includes
Core	SQLite, the PostgreSQL and MySQL wire families, Microsoft SQL Server, Redis-compatible servers, and the extension manager. These drivers are part of Strata itself.
Complete	Everything in Core, plus a signed *driver extension* for each further group of databases, installed with Strata (listed below).

Driver extension	Identifier	Databases
DuckDB	`cx.dixon.duckdb`	DuckDB
Analytics engines	`cx.dixon.analytics`	ClickHouse, Cassandra, ScyllaDB, Druid, Pinot, Trino, Hive, Spark SQL and other analytical engines
Document databases	`cx.dixon.documents`	MongoDB, CouchDB and other document stores
Time series	`cx.dixon.timeseries`	Prometheus-compatible, InfluxDB and other time-series servers
Graph databases	`cx.dixon.graph`	Neo4j, Gremlin servers, RDF stores and other graph databases

Driver extension	Identifier	Databases
Search engines	`cx.dixon.search`	Elasticsearch, OpenSearch, Solr and other search engines
Vector databases	`cx.dixon.vector`	Qdrant, Milvus, Weaviate, Chroma
Key-value stores	`cx.dixon.keyvalue`	Memcached, etcd, Aerospike, Riak KV
Proprietary databases	`cx.dixon.proprietary`	Oracle, Db2, SAP HANA and other proprietary-protocol databases
Message brokers	`cx.dixon.messaging`	Kafka, Pulsar, RabbitMQ, NATS and other messaging systems
Amazon Web Services	`cx.dixon.aws`	Amazon DynamoDB, Athena, Timestream, Kinesis, SQS and SNS
Cloud services	`cx.dixon.cloud`	libSQL and Turso, Microsoft Fabric, Snowflake, BigQuery, Databricks SQL, Azure Data Explorer, Cosmos DB, Pinecone, Pub/Sub, Service Bus
Files	`cx.dixon.files`	CSV, JSON, Excel, Parquet and other files (Chapter 6)
SQL Server, Redis	`cx.dixon.sqlserver`, `cx.dixon.redis`	Only for installations that do not already include these drivers

To see what you have, open **File › Extensions…**: each installed driver extension has a card there. A product whose extension is missing, disabled or cannot load stays in the lists, greyed, with the reason (Section 8.2).

Test status

Each product in Strata's catalogue has a status that records evidence, not intent:

Status	Means
Exploration tested	Connecting, running queries and browsing in the navigator passed live tests against a real server of the version named in the catalogue.
Query tested	Connecting and running queries passed live tests; browsing was not part of the test.
Available	A driver exists, but it has been tested against fake servers at most. It may work; nobody has shown that it does.
Catalogued	Known and planned. Nothing is implemented; you cannot connect.

This guide calls a product *supported* only when it is **exploration tested** (or, for one product, query tested). They are:

Category	Edition	Exploration-tested products
Relational	Core	PostgreSQL, TimescaleDB, CockroachDB, YugabyteDB, Materialize, RisingWave, CrateDB, QuestDB PostgreSQL endpoint, MySQL, MariaDB, Percona Server, TiDB, Vitess, MatrixOne, Microsoft SQL Server, SQLite, SQLCipher (read-only)
Analytical	Core	Apache Doris, StarRocks
Key-value	Core	Redis, Valkey, KeyDB, Dragonfly, Garnet
Vector	Core	Redis vector search, PostgreSQL pgvector
Search	Core	Manticore Search (query tested)
Relational	Complete	Oracle Database, IBM Db2 LUW, Firebird, SAP HANA, Informix, InterSystems IRIS, MonetDB, Vertica, Exasol, CUBRID, DuckDB, Microsoft Access MDB, libSQL
Analytical	Complete	ClickHouse, Apache Druid, Apache Pinot, Apache Spark SQL, Apache Hive, Apache Drill, Trino, PrestoDB, Dremio, Apache Calcite Avatica, HeavyDB (as OmniSci 5.10), Apache Arrow Flight SQL
Document	Complete	MongoDB, FerretDB, CouchDB, Couchbase, RavenDB, ArangoDB, SurrealDB, OrientDB, RethinkDB, MarkLogic, Azure Cosmos DB (emulator)
Key-value	Complete	Memcached, etcd, Aerospike, Riak KV
Wide-column	Complete	Apache Cassandra, ScyllaDB
Time series	Complete	VictoriaMetrics, Prometheus, Grafana Mimir, Thanos, Cortex, M3 Query, InfluxDB 1.x, InfluxDB 2.x, InfluxDB 3.x, QuestDB, TDengine, Apache IoTDB, GreptimeDB, Apache OpenTSDB, Graphite, Warp 10, Azure Data Explorer (emulator), Machbase, DolphinDB
Search	Complete	Elasticsearch, OpenSearch, Apache Solr, Meilisearch, Typesense, Quickwit, Vespa, ZincSearch, VictoriaLogs, VictoriaTraces
Graph	Complete	Neo4j, Memgraph, JanusGraph, Apache TinkerPop Gremlin Server, Dgraph, TigerGraph, NebulaGraph, TypeDB, TerminusDB, Virtuoso RDF, Ontotext GraphDB, Apache Jena Fuseki
Vector	Complete	Qdrant, Milvus, Weaviate, Chroma, Pinecone (emulator)
Messaging	Complete	Apache Kafka, Redpanda, ksqlDB, Apache Pulsar, RabbitMQ, NATS, NATS JetStream, ActiveMQ Artemis, ActiveMQ Classic, Google Pub/Sub (emulator)
Cloud (AWS)	Complete	Amazon DynamoDB, Amazon Kinesis, Amazon SQS, Amazon SNS (tested against the LocalStack emulator, not AWS itself)
Files	Complete	CSV, TSV, JSON, JSON Lines, XML, YAML, XLSX, XLS, DBF, Apache Parquet, Apache Arrow IPC, Apache Avro, Apache ORC, Protocol Buffers, Delta Lake, Apache Iceberg, Apache Hudi, LevelDB, RocksDB, LMDB, bbolt (Chapter 6)

⚖ **Honesty** "Exploration tested" means one version of each product passed Strata's live suite, almost always against a test server on Linux. Other versions, managed cloud services and other operating systems were not tested. Many products have documented limits: no TLS test, insecure mode only, one statement per line, and so on. Read them in **View › Database catalogue** before you rely on a product.

These products connect with an existing driver but are **not tested against a real server** (status *Available*): Greenplum, Apache Cloudberry, Greengage, KingbaseES, EnterpriseDB, Amazon Aurora, Amazon RDS, Google Cloud SQL and AlloyDB, Azure Database for PostgreSQL and MySQL, Neon, Supabase, Yellowbrick, Amazon Redshift, SingleStore, PlanetScale, OceanBase (MySQL mode), MySQL NDB Cluster, Azure SQL Database and Managed Instance, Azure Synapse Analytics, Redis Streams, Denodo and Sphinx (its SphinxQL path is tested only through Manticore Search) (Core); Microsoft Access ACCDB, ClickHouse Cloud, Apache Kylin, Cloudera Impala, Amazon DocumentDB, Azure Cosmos DB's MongoDB API, Amazon Keyspaces, YugabyteDB YCQL, Amazon Neptune, Stardog, Apache Phoenix (Phoenix Query Server with JSON serialization), Confluent Cloud, Azure Event Hubs, Oracle Autonomous Database, IBM Db2 for z/OS, IBM Db2 for i, Amazon Athena, Amazon Timestream, Turso, Microsoft Fabric, Snowflake, Google BigQuery, Databricks SQL and Azure Service Bus (Complete). Products marked *emulator* above were tested against the vendor's local emulator, not the hosted service. Try them on data you can afford to lose first.

The database catalogue

View › Database catalogue lists every product Strata targets, with a search box, category, tier and status filters, the tested versions and the known limitations. Products your installation can serve have a **New connection** button. A product it cannot serve says why.

2.2 Create a connection

1. Choose **File › New connection…** (`Ctrl` + `Shift` + `N`), or the **+** button at the top of the **Database Navigator**.
2. In **Connect to a database**, type part of the product's name in **Search products** and click its card. A product whose driver extension is missing, disabled or cannot load is greyed and badged **needs extension**; hover over it for the reason.
3. Fill the **Main** tab. Fields marked ***** are required.
4. If you need them, fill the **SSL / TLS**, **SSH**, **Proxy** and **Driver properties** tabs. A tab appears only when the product has settings on it.
5. Choose **Test connection**. Wait for the result under the form.
6. Choose **Save**.

👁 **You should see** the connection in the **Database Navigator**. Double-click it to connect.

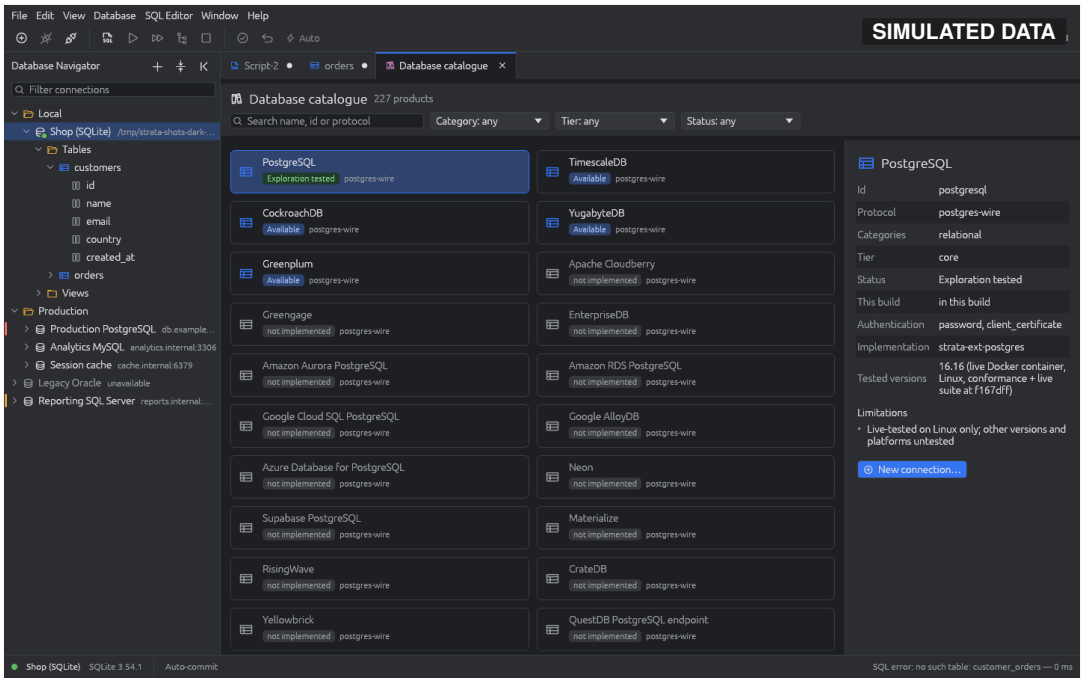


Figure 2.1. The database catalogue: search and filters on the left, a product's tested versions and limitations on the right. *Simulated data.*

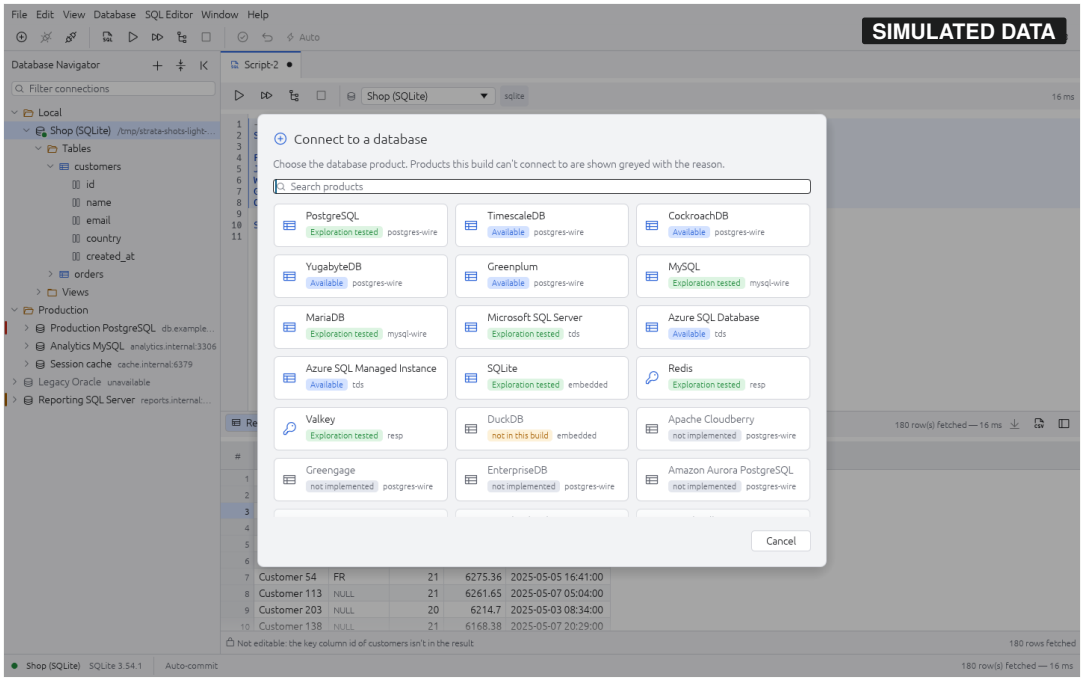


Figure 2.2. **Connect to a database**: products your installation can serve come first. *Simulated data.*

The Main tab

Field	What it does
Name	What the navigator shows.
Folder	Groups connections in the navigator. Use `/` for subfolders: `Prod/EU`.
Connection type	**Development**, **Test** or **Production**. Test connections get an amber bar in the navigator, Production a red one, and both show a badge in their editors and the status bar. On Production, an UPDATE or DELETE without WHERE asks before it runs.
Database type	The product. Changing it changes the fields below.
Host, **Port**, **Database**	Where the server is. Empty host and port use the product's defaults, shown greyed in the field.
User, **Password**	The database account.
Read-only	Asks the database to refuse writes for this session (below).
Save password	Keeps the password in the encrypted store. Without it, Strata asks each time you connect.

> ⚠ **Careful** **Read-only** is a guard against slips, not a security control. SQLite opens the file read-only; PostgreSQL and MySQL start each session in read-only mode, but a statement can switch it back. The SQL Server driver refuses to open a read-only connection at all, because SQL Server has no such session setting. If writes must be impossible, use an account without write privileges.

2.3 Manage connections in the navigator

Right-click a connection for **Connect** or **Disconnect**, **New SQL editor**, **Refresh**, **Edit connection…**, **Duplicate**, **Delete…** and **Copy name**. The filter box at the top searches names, database types, hosts and folders. Right-click a folder to rename or delete it; its connections and subfolders move up a level.

- **File › Recent connections** and the welcome page reopen the connections you used last.
- **File › Export connections…** writes your connections to a JSON file, without passwords or other secrets.
- Editing a connection whose query is still running asks you to cancel the query first, because the new settings need a new session.

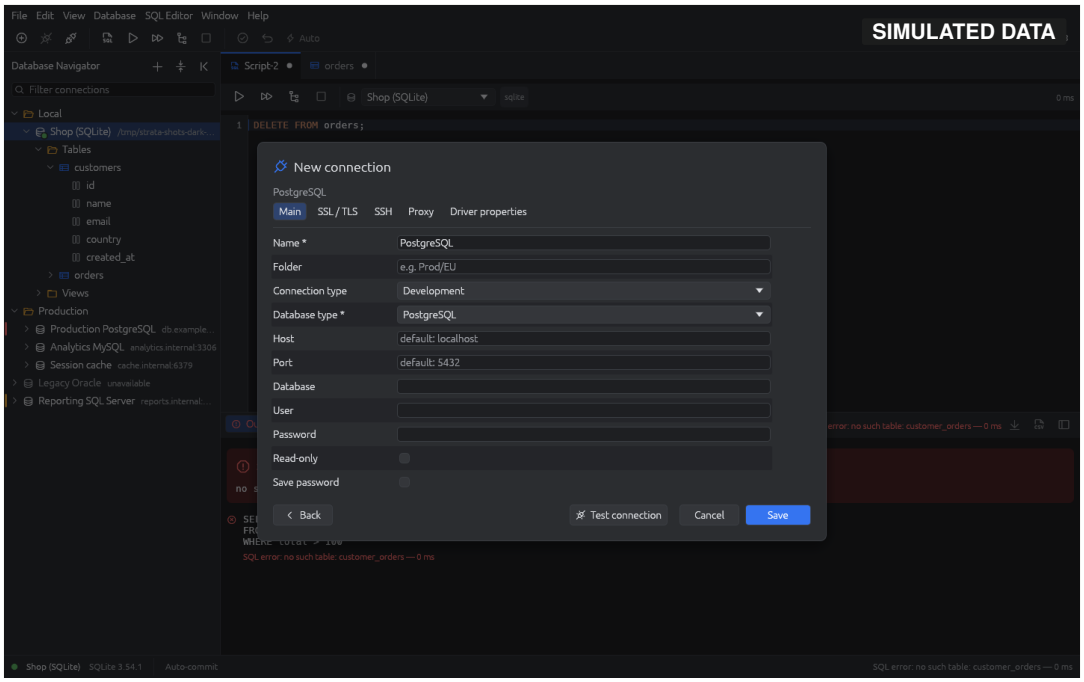


Figure 2.3. The connection form for PostgreSQL: the **Main** tab, with **Test connection**, **Cancel** and **Save**. *Simulated data.*

2.4 Reach a server through an SSH tunnel

⏩ **Skip this if** you can reach the database server directly.

An SSH tunnel carries the database connection through an SSH server you can log in to. On the **Main** tab set **Network** to **SSH tunnel**; then fill the **SSH** tab.

Field	What it does
SSH host, **SSH port**	The SSH server (port 22 if empty).
SSH user	Your login on the SSH server.
SSH authentication	**Password**, **Public key** or **SSH agent** (agent: Unix only, through SSH_AUTH_SOCK).
SSH password	For **Password**.
Private key file, **Key passphrase**	For **Public key**: an OpenSSH, PKCS#8 PEM or PuTTY file.
Keep-alive interval (ms)	How often to check the session is alive.
SSH connect timeout (ms)	How long to wait for the connection and login.

The **Host** on the **Main** tab is still the database server, as the SSH server sees it. TLS, if you use it, still checks the database server's real name, not the tunnel.

The first connection: trusting the host key

The first time Strata meets an SSH server, it shows **Unknown SSH host key** with the server, key type and `SHA256:` fingerprint.

> **Security** Compare the fingerprint with one you got from the server's administrator, for example from `ssh-keygen -lf` run on the server's public key, *before* you choose **Trust and connect**. Trusting adds the key to `<data dir>/known_hosts`. Strata also reads your own `~/.ssh/known_hosts`, but never writes to it.

If a server's key has changed since you trusted it, Strata shows **SSH host key CHANGED: not connecting** and refuses, with no way to override. A server that now offers a key of a different type from the one you trusted counts as changed too, not as a new server. Someone may be intercepting the connection. If the key was genuinely replaced, remove the line the message names from the `known_hosts` file and connect again.

> **Technical detail** The tunnel listens on `127.0.0.1` at a random port while the connection is open. As with `ssh -L`, any local process, including other users on the same computer, could connect to that port; the database's own login still applies. Not supported: jump hosts, an SSH tunnel through a proxy, keyboard-interactive or two-factor SSH logins, and SSH certificates. RSA keys work, but their signing has a known timing weakness that a well-placed attacker could in principle use to learn the key; Ed25519 and ECDSA keys avoid it, so prefer them.

Done when **Test connection** succeeds with **Network** set to **SSH tunnel**.

2.5 Connect through a proxy

Skip this if you do not use a SOCKS or HTTP proxy.

Set **Network** to **SOCKS5 proxy** or **HTTP CONNECT proxy**, then fill the **Proxy** tab: **Proxy host**, **Proxy port**, and **Proxy user** and **Proxy password** if the proxy needs them. A SOCKS5 proxy resolves the database host name itself.

> **Careful** An HTTP CONNECT proxy receives its user name and password in clear text (Basic authentication). Use it only on a network you trust, or put TLS on the database connection too.

2.6 Encrypt and verify with TLS

TLS encrypts the connection and, in its verifying modes, proves you reached the real server. Set it on the **SSL / TLS** tab.

TLS mode	What happens
`disabled`	Never use TLS. Everything, including the password, crosses the network readable.
`preferred`	Use TLS if the server offers it, otherwise plaintext. **An attacker on the network can force plaintext.**
`required`	Encrypted, but the server's identity is **not** checked. Any certificate is accepted.
`verify-ca`	The certificate chain is checked; the host name is not.
`verify-full`	Chain, validity dates and host name are all checked. Use this.

Other fields on the tab: **CA certificate** (when your server's certificate comes from a private CA), **Client certificate** and **Client key** (for mutual TLS; give both or neither).

HTTP or HTTPS

Products that Strata reaches over HTTP, such as the search engines, have a **Protocol** list: **HTTP**, **HTTPS** or **Driver default**. **HTTPS** sets TLS mode `verify-full`; **HTTP** sets `disabled`. Put only the host name in **Host**, without `https://`; otherwise Strata says **Enter the hostname in Host and choose HTTP or HTTPS under Protocol**. The list is hidden when the connection is given as a URL, which says the protocol itself.

> 🚫 **Security** Choosing `required`, or SQL Server's **Trust server certificate (insecure)**, shows an **INSECURE** warning, and saving asks you to confirm with **Save insecure connection**. In that mode a machine in the middle can impersonate the server and capture the password and every row. Prefer `verify-full` with a **CA certificate**.

Certificate pins

PostgreSQL connections (and SQL Server, and Cassandra or ScyllaDB in the Complete edition) can also check the server's public key against **Certificate pins (SHA-256)**: one or more SHA-256 digests of the server certificate's public key, in hex or base64, separated by commas. Pins need TLS mode `required`, `verify-ca` or `verify-full`. MySQL cannot check pins, so the field is disabled for MySQL with the reason.

Microsoft SQL Server

SQL Server's form puts the named instance on **Main**, and `encrypt` and **Trust server certificate (insecure)** on **SSL / TLS**. A SQL Server connection with no TLS setting is encrypted and verified by default. With TLS mode `disabled`, the login packet (including the password, which SQL Server only obfuscates) crosses the network readable.

> ⚖ **Honesty** These TLS modes are documented in detail for MySQL and MariaDB, PostgreSQL, SQL Server and Cassandra or ScyllaDB. Other drivers support TLS to different degrees; the catalogue's limitations say which. Strata does not use your operating system's trust store: a private CA goes in **CA certificate**.

✅ **Done when** **Test connection** succeeds with TLS mode `verify-full`.

2.7 Sign in to Amazon Web Services

⏩ **Skip this if** you do not use the Amazon Web Services driver extension (DynamoDB, Athena, Timestream, Kinesis, SQS, SNS).

An AWS connection has no password of its own. Strata takes the first credentials it finds, in this order:

1. **Access key ID** and **Secret access key** on the **Main** tab, with **Session token** for temporary keys.
2. The profile named in **Profile (~/.aws/config, ~/.aws/credentials)** on the **Driver properties** tab.
3. The environment variables `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` and `AWS_SESSION_TOKEN`.
4. The profile in `AWS_PROFILE`, otherwise the `default` profile.

The region comes from the connection's **Region** field, then the profile, then `AWS_REGION` or `AWS_DEFAULT_REGION`, and otherwise `us-east-1`. Strata reads the same files as the AWS command-line tools, including `AWS_CONFIG_FILE` and `AWS_SHARED_CREDENTIALS_FILE` when set.

IAM Identity Center (SSO). For a profile set up with `aws configure sso`, sign in with the AWS command-line tools first:

```
aws sso login --profile <name>
```

Strata uses that sign-in to fetch temporary credentials and renews them while it lasts. When it expires, Strata's error tells you to run the command again.

⚖ **Honesty** Strata does not support profiles that assume a role (`role_arn`, `source_profile`), run a `credential_process` or use a web identity token, and it does not use EC2 or container instance roles. For those, run `aws configure export-credentials --profile <name>` and enter the temporary keys it prints. SSO sign-in was tested against a simulated AWS portal, not AWS itself.

2.8 Keep saved passwords safe

Strata stores connection settings and secrets apart. `connections.json` never holds a password. Secrets go into `credentials.v2`, encrypted with XChaCha20-Poly1305 and authenticated, so a changed or damaged file is refused rather than half-read.

What protects the key depends on your computer:

Key source	Protects against	Does not protect against
OS keyring (default when available: macOS Keychain, Windows Credential Manager, Linux Secret Service)	Someone copying your data directory without your unlocked keyring	Malware running as you while the keyring is unlocked
Key file `credentials.key` (when no keyring is available, for example on a headless Linux machine)	Other users of the computer who cannot read your files	**Anyone who can read the data directory**: backups, a stolen disk, malware running as you
Your own password (opt-in; Argon2id)	Theft of the data directory, as far as your password is strong	Malware running as you while Strata is unlocked

Protect the store with a password

1. Open **File** › **Settings…** (`Ctrl` + `,`) and choose the **Security** page.
2. Under **Saved passwords**, type a password of 8 or more characters in **Password** and again in **Repeat**.
3. Choose **Set password**.

From then on, Strata asks for that password at start-up (**Unlock saved passwords**). If you choose **Continue without**, saved passwords stay locked: connections ask for their password, and nothing new is saved until you choose **File** › **Unlock saved passwords…**.

The same page has **Change password**, **Remove password** and **Rotate key**, which re-encrypts the saved secrets under a new key and removes the old keyring entry or key file.

> **Your data** There is no way to recover a forgotten store password. Strata cannot decrypt the store without it, and nobody else can either. Keep it in a password manager. If it is lost, you will need to re-enter every saved database, SSH and proxy password.

> **Why this matters** Strata only stores a database password if you tick **Save password**. SSH and proxy passwords and key passphrases are always stored when you enter them, encrypted in the same way. Passwords are kept out of the log: `strata.log` never contains passwords, SQL text or result data.

Done when Strata asks for the store password at its next start.

→ **Next** write and run SQL (Chapter 3).

CHAPTER 3

Write and run SQL

TL;DR

Open an SQL editor on a connection with `Ctrl` + `]`. `Ctrl` + `Enter` runs the statement at the cursor, `Alt` + `X` the whole script, `Ctrl` + `Shift` + `E` explains the plan and `Esc` cancels. Statements with parameters ask for values first. On a Production connection, an UPDATE or DELETE without WHERE asks before it runs. An editor can show two views of its text side by side, or move to a window of its own.

Time 10 minutes to try everything here.

Before you begin a saved connection (Chapter 1 or Chapter 2).

3.1 Open an editor

Select a connection in the **Database Navigator**, then choose **File › New SQL editor** (`Ctrl` + `]`). The new tab is bound to that connection. Its header has a connection list (change it and the text stays) and, where the database has them, a default schema. Right-click a schema or database in the navigator for **New SQL editor** there. On a connection that does not take SQL, such as Redis, the same item is called **New query editor**.

Syntax highlighting follows the connection's SQL dialect, and the statement that `Ctrl` + `Enter` would run is shaded, so you can see what will run before you run it.

3.2 Run statements

Each command has a keyboard shortcut, and most are in a menu as well.

To	Do this
Run the statement at the cursor	`Ctrl` + `Enter`, or **SQL Editor › Execute statement**. With text selected, the selection runs.
Run the whole script	`Alt` + `X` or `Ctrl` + `Shift` + `Enter`, or **SQL Editor › Execute script**. Each statement gets its own result tab.
Stop a running statement	`Esc`, or **SQL Editor › Cancel**. If the completion list, the find bar or a cell editor is open, `Esc` closes that first; press it again to cancel.
Comment lines out, or back in	`Ctrl` + `/`, or **Edit › Toggle comment**.

To	Do this
Complete a name	`Ctrl` + `Space`. Suggestions come from SQL keywords and the objects already loaded in the navigator. Arrows choose; `Enter` or `Tab` inserts; `Esc` closes. A single match is inserted at once.
Find text	`Ctrl` + `F`. Case-insensitive, wraps at the end; `Enter` finds the next match, `Esc` closes.
Open or save a script	`Ctrl` + `O` and `Ctrl` + `S` (`.sql` files).

👁 **You should see** a result tab per statement, named like **Result 1 (180 rows)**, and an **Output** tab with what ran and how long it took. A failed statement turns **Output** into an error panel with the database's message.

3.3 Split, move and fold the editor

Three buttons in the editor's toolbar change how you see the text. None of them changes the text or the connection.

To	Do this
See two parts of a long script at once	Choose the split button (**Split editor: a second view of the same text, side by side**). Both views edit the same text. Choose it again to close the second view.
Give the editor a window of its own	Choose **Open in its own window**. The editor, its connection and its results move to a new window, which you can put on another screen. The tab stays in the main window and says **This editor is open in its own window.**
Bring it back	Choose **Return to the main window** in the new window, **Return it here** in the tab, or close the new window. Text, results and connection come back unchanged.

Folding

With the SQL Intelligence extension (Section 8.3), statements and blocks that span several lines get a fold toggle in the margin. Click it to hide the lines; click it again to show them. Above the text, ***n* folded** counts the hidden blocks, and **Unfold all** shows everything.

You can keep typing while blocks are folded. Folds move with your edits, and putting the cursor inside a fold, or editing it, opens it.

ℹ **Why this matters** Folding only hides lines on screen. `Ctrl` + `Enter`, `Alt` + `X`, saving, and copying or cutting a selection that spans a fold all use the full text, hidden lines

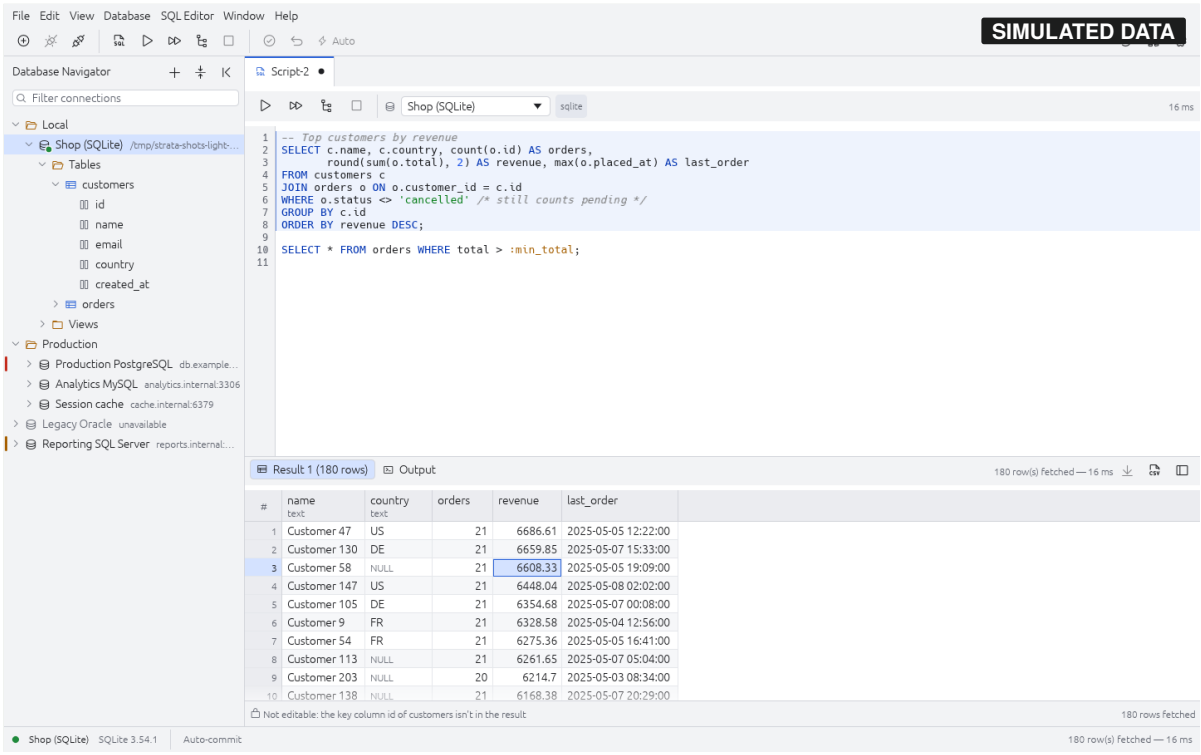


Figure 3.1. An SQL editor bound to **Shop (SQLite)**. The shaded block is the statement `Ctrl` + `Enter` would run; its result is underneath. *Simulated data.*

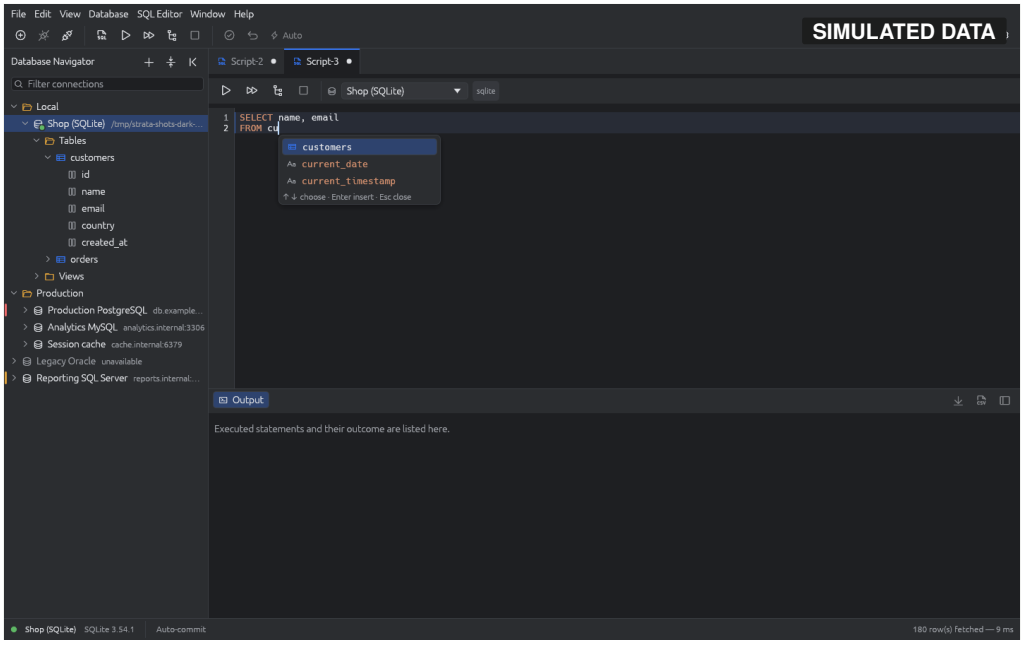


Figure 3.2. Completion after `Ctrl` + `Space`: table and column names come from what the navigator has loaded. *Simulated data.*

included.

3.4 Statements with parameters

A statement that contains `:name` or `?` placeholders opens **Bind parameters** before it runs. Type a value for each, or tick **NULL**, then choose **Run**.

```
SELECT * FROM orders WHERE total > :min_total;
```

Values are *bound*, never pasted into the SQL text, so a value cannot change the statement. Strata remembers them for the rest of the session.

3.5 Production connections ask first

Your data On a connection whose **Connection type** is **Production**, an UPDATE or DELETE with no WHERE clause stops at **Change every row on a production connection?**. Read the connection name in the question. **Run anyway** runs it; **Cancel** does not. `Enter` does not answer this question, or any other marked as dangerous: you have to choose the button. On Development and Test connections there is no such question, so check the shaded statement before you press `Ctrl` + `Enter`.

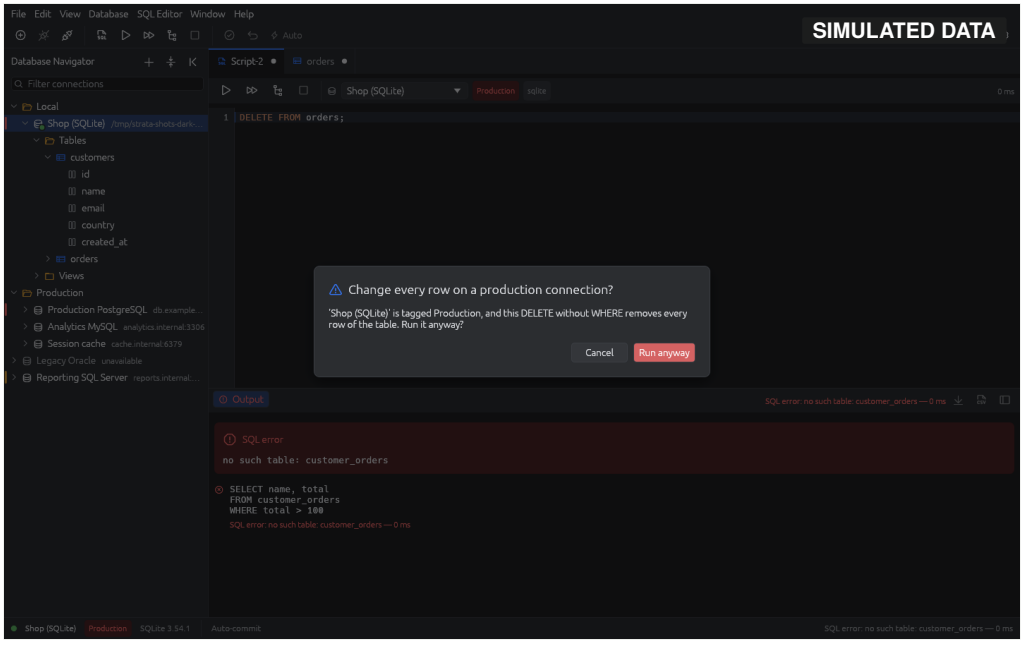


Figure 3.3. The production guard, before a DELETE without WHERE runs. *Simulated data.*

3.6 See how a query will run

SQL Editor › Explain plan (`Ctrl` + `Shift` + `E`) shows the database's execution plan for the statement at the cursor in a new tab, as a tree. It is available only when the connection's database can explain; otherwise the toolbar button says **Explain plan: not supported by this connection**.

> ⚠ **Careful** **SQL Editor › Explain analyze…** *runs* the statement to measure it, and asks first (**Run the statement to measure it?**). PostgreSQL rolls back its data changes (but not effects outside the transaction, such as sequences); MySQL and MariaDB keep them. Use plain **Explain plan** for statements that change data.

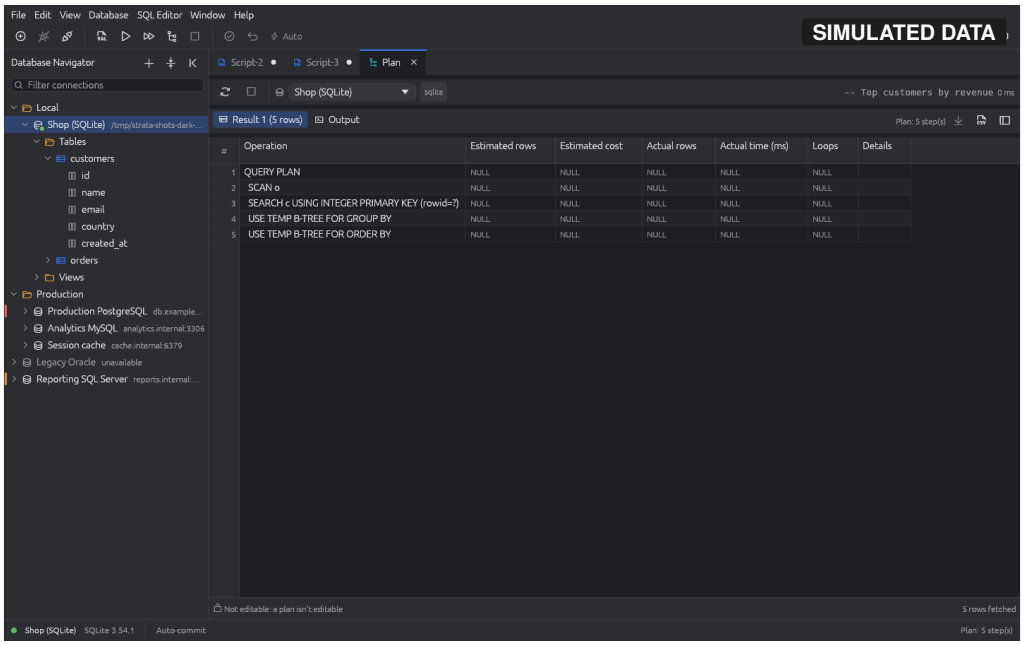


Figure 3.4. An explain plan tab. *Simulated data.*

3.7 Transactions

By default each statement commits on its own (auto-commit). The toolbar button shows **Auto**; click it to switch the connection to manual commit (it then shows **Manual**). Strata remembers the choice per connection, and **Database › Auto-commit** does the same.

In manual mode the toolbar shows **Transaction: 2 pending** (or however many statements the open transaction holds), with **Commit** and **Rollback** beside it. Disconnecting or quitting with an open transaction asks **Commit** / **Roll back** / **Cancel**, and a connection cannot be edited or deleted until its transaction ends.

> ⚠ **Careful** If the connection drops in the middle of a transaction, Strata says the transaction is lost and stops the script. Statements that commit implicitly on the server, such as DDL on MySQL, are not detected: the pending count can be wrong after one.

3.8 Tabs, history and closing

- `Ctrl` + `Tab` and `Ctrl` + `Shift` + `Tab` move between tabs; drag a tab to reorder it; **Window** lists them all. A dot after a title means unsaved changes.
- `Ctrl` + `W` closes the tab. An editor with unsaved text asks **Save changes?** (**Save** or **Don't save**); unsaved result edits ask **Close without saving?**.
- **View** › **Query history** lists the statements you ran in this session, with **Open in editor** and **Copy**. **Open in editor** opens the statement on the connection it ran on, ready to run again. History is not kept after Strata closes.
- A query timeout (**File** › **Settings…**, **Query limits**) cancels any statement that runs longer.

> **Tip** Feature extensions add more to the editor: checks, formatting and snippets with SQL Intelligence, and a review of risky statements before they run with Operation Preflight (Section 8.3).

3.9 Non-SQL editors

Skip this if you only use SQL databases.

Some products take their own language in the editor. Redis and Valkey editors take one command per line (`GET key`); `Ctrl` + `Enter` sends the current line, exactly as written. Opening a key in the navigator shows a viewer that pages through its value and its TTL, and lets you set the TTL where the server supports expiry. Other Complete-edition products (Prometheus' PromQL, for example) have their own editors; their catalogue entries describe what they accept.

Done when you have run a statement, a script and an explain plan, and cancelled a slow query with `Esc`.

Next work with the results (Chapter 4).

CHAPTER 4

Read, filter and edit results

TL;DR

A result shows up to 200 rows unless you fetch more. Click, Shift+click or drag to select; `Ctrl` + `C` copies. Double-click a table to see its data, with a WHERE filter. Results from one table with a key can be edited: changes are held until **Save** applies them all in one transaction, or **Discard** drops them.

Time 5 minutes.

Before you begin a connection with a table you are allowed to change, if you want to try editing. Use a test database.

4.1 The grid

- **Select:** click a cell; Shift+click or drag for a block; click row numbers for rows; the corner cell or `Ctrl` + `A` for everything.
- **Move:** arrow keys, `Page Up` and `Page Down`, `Home` and `End`; `Ctrl` + `Home` and `Ctrl` + `End` jump to the first and last cell.
- **Columns:** they start fitted to their contents. Click a header to sort; drag its edge to resize; double-click the edge to fit.
- **NULL** shows as a dimmed *NULL*, so it never looks like the text `'NULL'`.
- **Value viewer: View › Value viewer panel** shows the selected value in full at the side: JSON pretty-printed, binary as hex.

To see more	Do this

4.2 How many rows you see

Strata reads a result up to the *fetch limit* (200 rows by default) and stops. A tab title with a plus sign, such as **Result 1 (200+ rows)**, means there is more that was not read.

To see more	Do this
This result	**SQL Editor** › **Fetch more rows**: the statement runs again with a higher fetch limit. Save or discard any pending edits first (Edit rows).
Every result from now on	Raise **Fetch limit** and **Display limit** in **File** › **Settings…**, **Query limits**, then **Save limits**.
All of it, in a file	Export it (Chapter 5). Export reads the whole result.

> ⚠ **Careful** **Fetch more rows** runs the statement *again*. For a SELECT that is harmless; for a statement with side effects it repeats them. Data may also have changed between the two runs.

4.3 Copy cells

- `Ctrl` + `C` copies the selection as tab-separated text; `Ctrl` + `Shift` + `C` adds the column names. Values are copied in full, not as displayed.
- Right-click **Copy as** offers **Tab-separated (with header)**, **CSV**, **JSON** and **SQL INSERT statements**.

4.4 Look at a table's data

Double-click a table or view in the navigator, or right-click it and choose **View data**. The data tab has a **WHERE** box: type one condition, such as `country = 'GB'`, and press `Enter` to run the table's SELECT again with it. **Count** runs `SELECT COUNT(*)` with the same filter. Right-click a table for **Generate SELECT** too.

On a connection that does not take SQL, such as Redis or a document database, the data tab has no **WHERE** box and the navigator offers no **Generate SELECT**, because both are SQL.

> ℹ **Why this matters** The filter accepts one condition and never a second statement, so a slip such as a stray semicolon cannot turn it into something else.

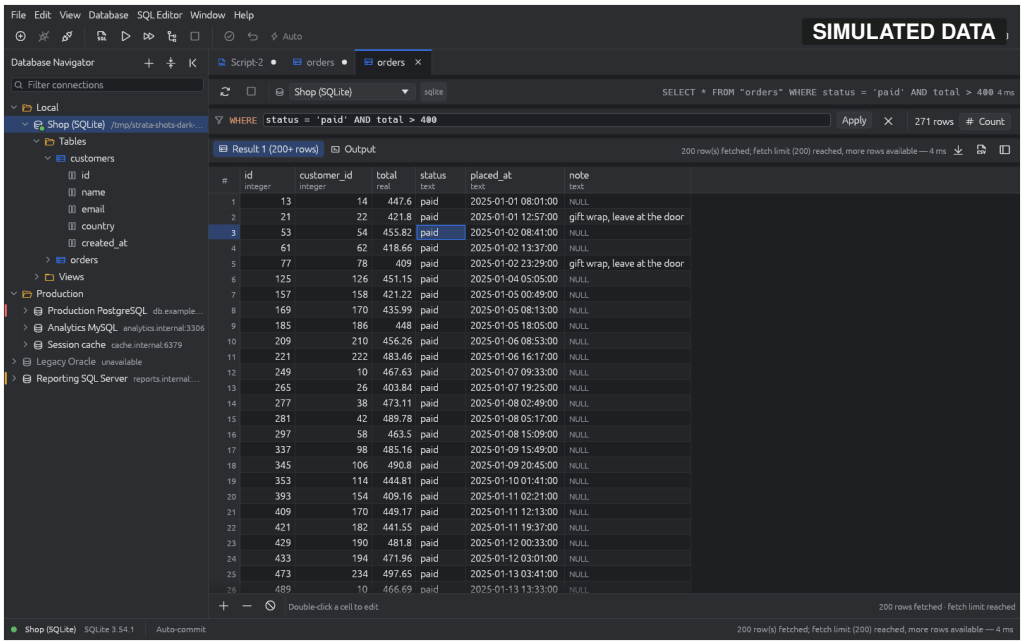


Figure 4.1. A table's data tab, filtered with **WHERE**. *Simulated data.*

4.5 Edit rows

A result can be edited when it comes from *one table* that has a primary key or unique key in the result, and the connection is not read-only. Otherwise the bar under the grid shows a lock and the reason, for example **Not editable: the key column id of customers isn't in the result**.

> 🚫 **Your data Save** writes to the database for real, all changes at once. Check the coloured cells before you save. On a shared or production database, use a test copy first. There is no undo once a save has committed; until then, **Discard** drops every pending change.

1. Double-click a cell, or select it and press `F2` or `Enter`. Type the new value. `Enter` keeps it; `Esc` abandons it.
2. To add a row, choose **Add row** in the bar under the grid (or **Edit › Add row**). To delete rows, select them and press `Del`. To set cells to NULL, choose **Set to NULL**.
3. Changed cells stay coloured. Review them.
4. Choose **Save** to apply them all in one transaction, or **Discard** to drop them.

👁 **You should see** the colours clear after **Save**. If any change fails, none is applied: the transaction is all or nothing, and the database's error is shown.

While a result has pending edits, Strata will not run its statement again, fetch more rows or apply a **WHERE** filter, because each of those would replace the grid and lose the edits. It says, for example, **2 pending change(s) in this result: save or discard them before running again.** Choose **Save** or **Discard**, then try again.

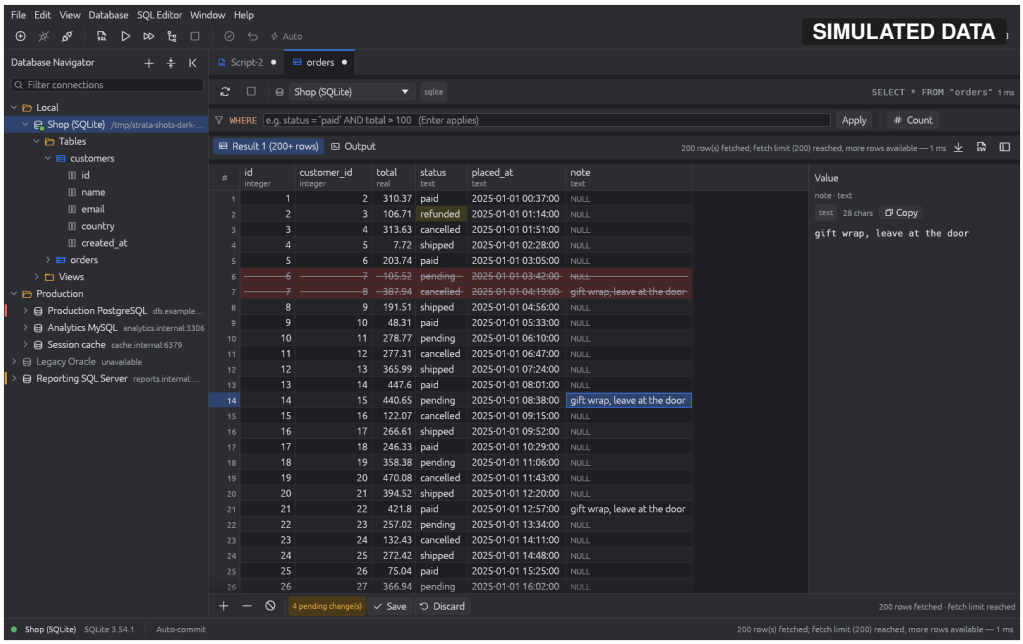


Figure 4.2. Pending edits: changed cells are coloured until **Save** or **Discard**. *Simulated data.*

Done when the edited rows show their new values after you run the SELECT again.

Next export the result, or import a CSV file (Chapter 5).

CHAPTER 5

Import and export data

TL;DR

Export: right-click a result and choose **Export to CSV…**; if the fetch limit cut the result short, Strata reads the whole thing again into the file. **Import:** right-click a table and choose **Import CSV…**; map the file's fields to the table's columns and choose **Import**. The import is all or nothing.

Time 5 minutes for each.

Before you begin for import, a table that already exists, on a connection that is not read-only.

5.1 Export a result to CSV

1. Run the query whose result you want.
2. Right-click the grid and choose **Export to CSV…** (or **SQL Editor › Export results to CSV…**).
3. In **Export to CSV**, type the file path. Leave **Guard against spreadsheet formulas (prefix cells starting with = + - @)** ticked unless you know why you need it off.
4. Choose **Export**.

If the result was complete, Strata writes what it has. If the fetch limit cut it short, the statement runs again on a separate connection and every row (up to the export limit, if you set one) streams to the file. The status bar shows progress and a **Cancel export** button.

You should see the file, with a header row of column names.

Why this matters A spreadsheet treats a cell that starts with =, +, − or @ as a formula, and a formula in data you did not write can run commands or leak data when someone opens the file. The guard puts a ' in front of such text so it stays text. It changes those values, so turn it off for files meant for another program rather than a spreadsheet.

Careful Because a cut-short result is read again, the file can differ from what you saw if the data changed in between, and a statement with side effects runs a second time. Export SELECT statements.

Done when the file opens and its row count matches `SELECT COUNT(*)` for the same query.

5.2 Copy instead of export

For a few rows, copying is quicker: select cells, then right-click **Copy as** and choose **CSV**, **JSON**, **SQL INSERT statements** or **Tab-separated (with header)**. Copying uses the rows already fetched; it does not run the statement again.

5.3 Import a CSV file into a table

Your data Import adds rows to a real table. Check the table name in the dialog's title before you choose **Import**. Try a new import on a test copy of the table first.

1. In the navigator, right-click the table and choose **Import CSV…**. The window is titled **Import CSV into** followed by the table's name.
2. Type the file's path. Set **Delimiter** and **Quote**, tick **First line is a header** if it has one, and fill **NULL as** with the text that means NULL in your file, if any.
3. Under **Mapping and preview**, check which field goes into which column.
4. Choose **Import**. Progress shows in the window; **Cancel import** stops it.

You should see **Imported 120 row(s) into orders.**, with your row count and table name.

If not if **Import CSV…** is greyed out, hover over it: it says **Can't import: the connection is read-only or its driver can't import**.

Technical detail Rows go in as batched INSERT statements with bound values, inside one transaction. If any row fails, or you cancel, the transaction rolls back and the table is left as it was.

Honesty Import has been tested end to end on SQLite. For MySQL and PostgreSQL, the parts it shares with other features are tested, but import itself has not been run end to end against real servers. Check the row count after importing.

Done when `SELECT COUNT(*)` on the table has gone up by the number of data rows in the file.

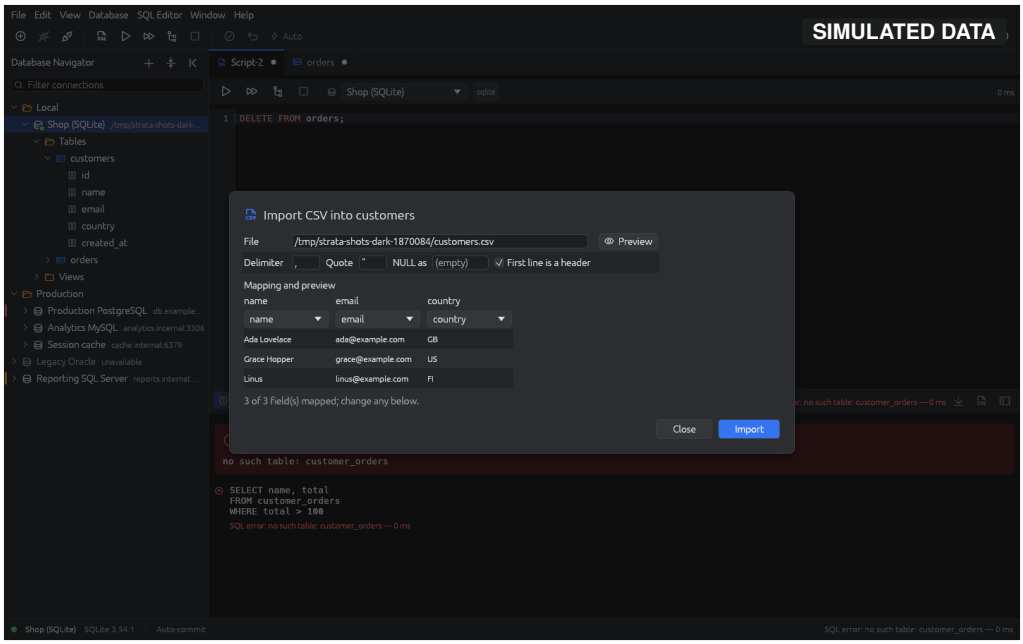


Figure 5.1. **Import CSV into** a table: delimiter, quote, header and NULL options, with the mapping and preview. *Simulated data.*

5.4 Export your connection list

File › Export connections… writes every connection to a JSON file. Passwords and other secrets are left out, so the file is safe to share with a colleague, who still needs their own credentials.

5.5 Time-series results

⏩ **Skip this if** you do not use Prometheus-compatible servers (Complete edition).

A PromQL result has a **Chart** and a **Grid** view and its own exports: **Export to CSV (a column per series)** and **Export as Prometheus JSON**, which writes every sample in the Prometheus API's matrix (or vector) format.

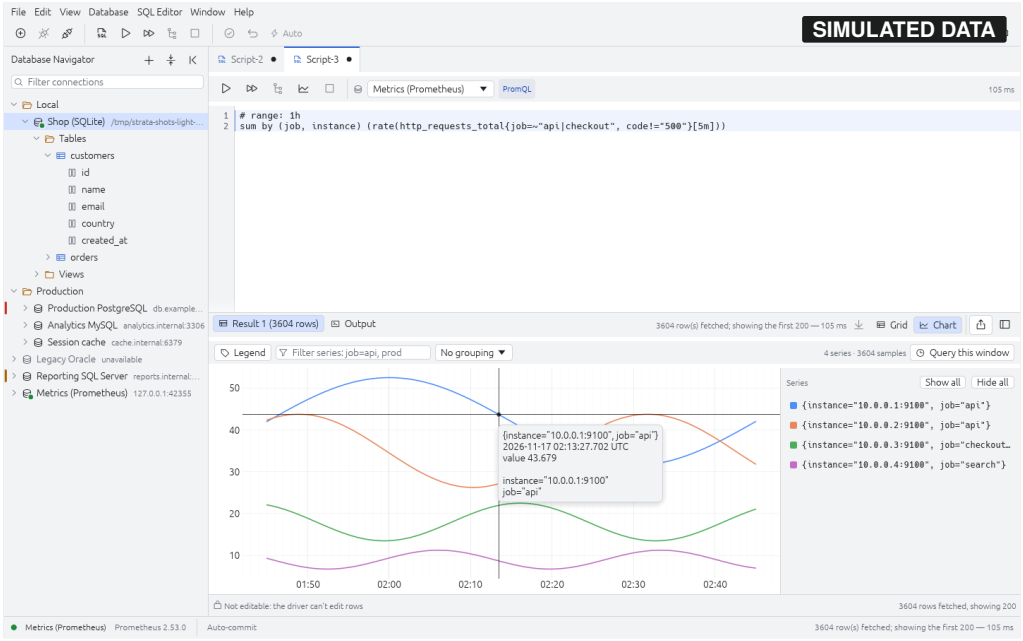


Figure 5.2. A PromQL result as a chart, from a fake Prometheus server. *Simulated data.*

CHAPTER 6

Query files with SQL

TL;DR

With the Files driver extension, a file or a folder of files can be a connection. Each file (or worksheet, or Access table, table folder or key-value database) becomes a read-only table you query with SQLite's SQL, joins included. Strata only ever reads the files.

Time 5 minutes.

Before you begin the Files driver extension, `cx.dixon.files` (part of the Complete edition; **File › Extensions…** shows it), and the files on this computer.

Skip this if you only query database servers.

6.1 What you can open

Format	Notes
CSV, TSV	Delimiter, quote and header row detected or set; any text encoding.
JSON, JSON Lines	An array of objects, or one value per line. Nested values become JSON text you can query with `json_extract` or `->>`.
XML, YAML	Repeated records become rows; any other document becomes a `path, value` table of its leaves.
XLSX, XLS	One table per worksheet. Formulas give their saved values; macros are never read or run. Excel 97–2003 workbooks are checked before they are opened, and one that would need too much memory is refused.
DBF	dBase, FoxPro and Visual FoxPro tables, with memo files beside them.
Apache Parquet, Arrow IPC, Apache Avro, Apache ORC	Column types come from the file's schema.
Protocol Buffers	Give the schema in **Protobuf schema**: a `.proto` file or a descriptor set from `protoc --descriptor_set_out` (a `.proto` with the data file's name next to it is found on its own). The file holds size-prefixed messages or one message (**Protobuf framing**); **Protobuf message type** picks the message. Fields become columns; nested messages, lists and maps become JSON.
Delta Lake, Apache Iceberg, Apache Hudi	A table folder on this computer becomes one table of its current snapshot (no time travel). Hudi merge-on-read tables show their last compacted state.

Format	Notes
Microsoft Access (`.mdb`)	Table data only. Saved queries, forms, macros and VBA are never read.
LevelDB	A database folder becomes one `key, value` table in key order. See Browse a LevelDB database.
RocksDB, LMDB, bbolt	A database folder (RocksDB, LMDB) or file (bbolt) becomes one table with a `column_family`, db or `bucket` column, then `key, value`. Strata reads the files itself, never writes them and takes no lock.

6.2 Connect to a file or folder

1. Choose **New connection…** and pick the format's card (for example **CSV**).
2. In **File or folder**, give the absolute path to one file or to a folder.
3. For a folder, tick **Include subfolders** if you want them. **Max files** (1000 by default) stops a huge folder from being read by accident; more files than that is an error.
4. For CSV, **CSV delimiter**, **CSV quote** and **CSV header row** start at `auto`; **Encoding** starts at `utf-8`. Change them only if the columns come out wrong.
5. Choose **Test connection**, then **Save**, then double-click the connection.

You should see one table per file in the navigator, its columns labelled with the type Strata inferred, such as `order_id (INTEGER)`. In a folder, tables are named after the file's path, with anything other than letters, digits and _ replaced.

Try it Join two files:

```
SELECT o.country, count(*) AS orders
FROM orders o JOIN events e ON e.order_id = o.order_id
GROUP BY o.country ORDER BY orders DESC;
```

Done when a SELECT on one of the file tables returns its rows.

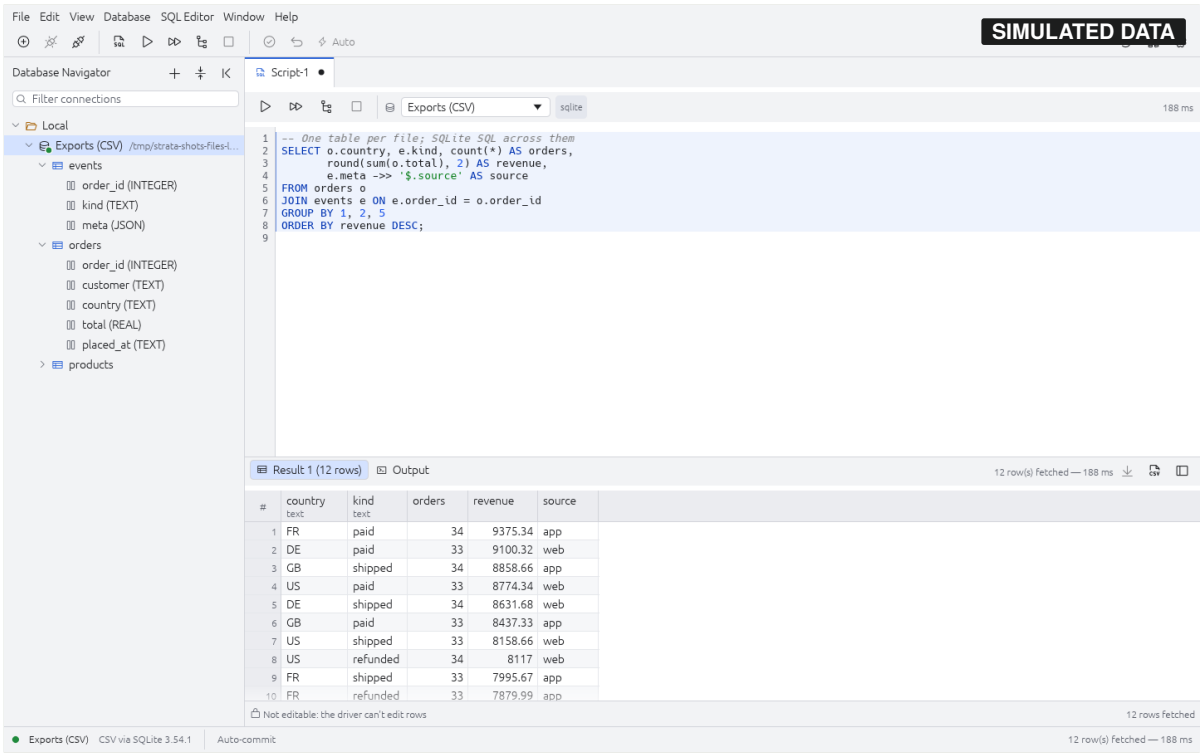


Figure 6.1. A folder of CSV and JSON files as a connection: one table per file, joined with SQLite SQL. *Simulated data.*

6.3 Browse a LevelDB database

LevelDB is the key-value store inside many desktop applications and browsers. A LevelDB database is a folder holding a file called `CURRENT` and a set of numbered files.

1. Choose **New connection…** and pick the **LevelDB** card.
2. In **File or folder**, give the absolute path to the database folder (or to its `CURRENT` file). A folder you connect to for other files can also hold LevelDB folders: each becomes a table of its own.
3. Choose **Test connection**, then **Save**, then double-click the connection.

👁 **You should see** one table named after the folder, with the columns `key (BLOB)` and `value (BLOB)`. A key or value that is valid UTF-8 text is shown as text; anything else is shown as hexadecimal bytes.

Rows come in LevelDB's own key order (byte by byte) and are read as you page, so a large database opens quickly. Conditions on `key` jump straight to the right place instead of reading everything:

```
-- one key
SELECT value FROM settings WHERE key = 'theme';
-- every key starting with user:
SELECT * FROM settings WHERE key GLOB 'user:*';
```

```
-- a range, and keys that are not text
SELECT * FROM settings WHERE key >= 'a' AND key < 'b';
SELECT * FROM settings WHERE key = x'00ff';
```

These comparisons are byte by byte, as LevelDB orders its keys. Other conditions, such as `LIKE`, read the whole database.

Why this matters Opening a LevelDB database with its own application normally changes it: a lock file is taken, recent changes are replayed and new bookkeeping files are written. Strata opens it without changing the folder at all, not even its lock file, so it also works on a read-only folder.

Careful If the application that owns the database is running and writing to it, a query may fail or miss its newest changes, because Strata does not lock the folder. Close the application first, or connect to a copy of the folder. Each query sees the folder as it is on disk at that moment.

6.4 How types are worked out

For CSV and JSON, Strata looks at the first 1000 records. A CSV column is `INTEGER` only if every value reads back exactly as an integer (so `007` keeps the column as `TEXT`), `REAL` for decimals, and `TEXT` otherwise. A value that does not fit its column's type is returned as text, never dropped. XLSX, Parquet, Arrow, Avro and Access columns take their types from the file.

6.5 Safety and limits

Why this matters Files are opened for reading only and never written: no side files, no changes to size or modification time. In a folder, a link that points outside the folder is skipped. The SQL you can run is limited to reading: INSERT, UPDATE, CREATE, ATTACH and functions that read or write other files are refused, whatever the spelling.

Careful There are no indexes. Every query reads each file from the start, and a file larger than 8 MiB is read again for every scan, so a join against a large file can be slow. Filter early, and copy the data into a real database if you query it often.

Honesty Not supported yet: XLSB, ODS and BadgerDB. Table folders are read only on this computer, at their current snapshot: Delta Lake deletion vectors and column mapping, Iceberg delete files and Hudi log files are refused or not merged. RocksDB merge operands and BlobDB values are refused. A program writing a RocksDB, LMDB or bbolt

database while you read it is not locked out, so a query may fail or miss its newest changes. LevelDB databases made with a custom key order are read in byte order or refused. A damaged LevelDB table is an error; a damaged entry in its recent-changes log is skipped, as LevelDB itself does. Password-protected Access files are refused. Access `.accdb` files have a reader but have not been tested (status *Available*).

→ **Next** the licence (Chapter 7).

CHAPTER 7

Activate and manage your licence

TL;DR

Until this computer is activated or on a trial, the **Strata licence** window blocks the app. Tick **I agree to the licence terms**, then paste your key and choose **Activate**, or choose **Start free trial**, and approve the request in your browser. After that, Strata checks the licence offline at every start and online about once a day. Seven days without a successful online check, and it asks you to connect.

Time About 5 minutes, most of it in the browser.

Before you begin a network connection, a browser, and your licence key if you have one.

7.1 Activate this computer

1. Start Strata. The **Strata licence** window appears. (Once licensed, open it again with **Help › Licence…**.)
2. Choose **Read terms** to read the licence agreement, then tick **I agree to the licence terms**. The buttons stay disabled until you do.
3. Either type or paste your key into **Licence key (optional)** and choose **Activate**, or choose **Start free trial**. To paste, press `Ctrl` + `V` (`Cmd` + `V` on a Mac), choose **Paste**, or right-click the field. Spaces and line breaks copied from an email or PDF are removed for you.
4. Your browser opens on Uncomplex's approval page, with this computer's code already in the link. Strata shows the same code, with **Open browser** and **Copy code** in case the browser did not open, and **Waiting for approval…**.
5. Approve the request in the browser.

You should see **Thank you, this computer is now activated.**, and the state at the top of the window: **Licensed.**, **Licensed, 12 day(s) remaining.** or **Trial, 14 day(s) remaining.**, with your numbers. **Close** then returns you to Strata.

> **Tip** The key field is optional. With it empty, **Activate** still opens the browser, and the licence is chosen on Uncomplex's approval page instead.

Done when the window shows **Licensed.** (or a trial with days remaining) and closes.

7.2 The licence window’s buttons

Button	What it does
Activate / **Change licence**	Starts activation with the key in the field. Shown as **Change licence** once licensed.
Start free trial	Asks for a trial. Not offered once a trial has ended.
Check licence	Verifies the licence with the server now.
Deactivate this computer	Frees this computer’s place on the licence, so you can use it elsewhere.
Cancel	Abandons an activation that is waiting for approval.
Quit	Shown while unlicensed: closes Strata (it still asks about unsaved work).

7.3 What the states mean

The window says	What to do
This computer is not activated.	Activate it, or start a trial.
Waiting for activation to be approved.	Approve it in the browser, or **Cancel**.
Your trial has ended. Enter a licence key to keep using Strata.	Activate with a key.
Your licence has expired.	Renew the licence, then **Check licence**.
Your licence starts in 3 day(s).	Wait, or use another licence.
The licence has not been checked for seven days. Connect to the internet and choose Check licence.	Exactly that.
The licence must be checked online. Choose Check licence.	Exactly that.
The system clock moved backwards. Correct the clock, then choose Check licence.	Fix the date and time, then check.
The licensing server cannot be reached. Try again later.	Check your network or proxy; try later.
This computer was removed from the licence. Activate it again to continue.	Activate again.
Your licence is in use on the maximum number of computers. Remove one in the customer portal, then activate again.	Free a computer in the customer portal, then activate.

The window says	What to do
This licence has been suspended or revoked. Please contact support.	Contact Uncomplex.

⚠ **Careful** Strata works offline between checks, but not indefinitely. The signed lease it keeps lasts at most seven days. If Strata cannot renew it in that time (a laptop kept offline, a blocked network), the licence window comes back until a check succeeds. Plan a connection before a long trip.

7.4 What is sent, and when

Licensing is the only network service Strata itself contacts; it never sends your databases, queries or results. The server is `https://licensing.uncomplex.co.uk`.

When	Trigger	What is sent
Activation or trial request	You choose **Activate** or **Start free trial**	Strata's product id; the licence key, if you typed one; a public key created for this installation; this computer's name; and hardware identifiers: on Linux the system UUID and board serial (when readable), maker, product and board names; on macOS the platform UUID; on Windows the system UUID.
Renewal	At start-up, then about every 24 hours while Strata runs	This installation's activation id and a signature made with its private key. No hardware details.
Check licence	You choose it	The same as renewal.
Deactivation	You choose **Deactivate this computer**	The activation id and a signature.

The installation's private key and the signed lease are stored in `<data dir>/licensing/`. The lease is verified offline, against a key that ships with Strata, at every start.

For how long the licensing service keeps activation records and hardware identifiers, and who processes them, see `https://uncomplex.co.uk` or ask `hello@uncomplex.co.uk`.

⚖ **Honesty** Strata cannot run without a licence or an active trial. The trial's length and what each licence includes are set by the licensing service, not by the app, so this guide does not quote them: the window shows your days remaining. The browser approval page and customer portal are part of that service and are not described here.

CHAPTER 8

Extensions

TL;DR

Extensions are optional, signed add-ons. *Driver extensions* let Strata connect to more kinds of database. *Feature extensions* add tools such as Schema Explorer, Migration Studio and Data Comparison; find them in the **Tools** menu and on right-click menus. *Data Laboratory* finds anomalies and statistics in a query result. The Complete edition installs the driver and feature extensions for you. Nothing an extension does is written to a database or sent off your computer until you confirm it in Strata's own dialog.

Time 5 minutes to install an extension; 10 minutes for a first feature or analysis.

Before you begin a saved connection. For Data Laboratory, a query whose result has at least one numeric column.

Skip this if you only use the databases of the Core edition and need none of the tools below.

8.1 What an extension is

An extension is an optional add-on that Strata loads only when you use its feature: a driver extension when you connect to one of its databases, a feature extension when you choose one of its actions, Data Laboratory when you analyse a result. Until you install an extension, you will not see anything of it, apart from the products it would serve, which are greyed in the product list.

Extension	What it adds
Driver extensions	One group of databases each, such as document databases or files. Section 2.1 lists them.
Feature extensions	One tool each: diagrams, schema design and migration, comparison and synchronisation, profiling, test data, anonymising, search, documentation, workspaces, charts, plan advice and safety checks. Section 8.3 lists them.
Data Laboratory	Analysis of query results: anomaly detectors, expected-value bands, severity scores, explanations, detector suggestions, a risk classifier and descriptive statistics.

8.2 Driver extensions

The Core edition's databases (SQLite, the PostgreSQL and MySQL families, SQL Server and Redis-compatible servers) are part of Strata. Every other group of databases comes as a driver extension, named `cx.dixon.` followed by the group, such as `cx.dixon.documents`.

Your edition	What to do
Complete	Nothing, unless you left groups out when you installed. The installer puts the driver extensions beside Strata, and they show as **bundled with Strata**.
Core	Install the driver extension for the group you need, as in Install an extension.

When a product's driver extension is missing, disabled or cannot load, the product is greyed in **Connect to a database** with the badge **needs extension**. A saved connection that uses it is greyed in the **Database Navigator** and marked **unavailable**. Hover over either for the reason, for example:

Can't connect: DuckDB: its driver extension cx.dixon.duckdb isn't installed. Install it, then see File > Extensions…

Other reasons say that the extension **is installed but disabled**, or give its status, such as **untrusted**, with **See File > Extensions…** at the end. Fix it there (If an extension is unavailable); the connection itself is kept unchanged and works as soon as the extension can load.

Done when the product's card in **Connect to a database** is no longer greyed.

8.3 Feature extensions

Feature extensions are tools that work through Strata. They never receive your passwords, host names or tunnels: they name a connection, and Strata runs the work on its own session, with values bound as parameters. The Complete edition installs all 21 and enables them; in the Core edition, install the ones you want as in Install an extension.

Where to find them

Place	What is there
Tools menu	The actions of every enabled feature extension, grouped by extension, each with its shortcut if it has one. **Tools › Extensions…** opens the extension manager. With none enabled, the menu says **No feature extensions are enabled.**
Navigator, right-click	Below Strata's own items: actions for a connection (for example **Schema explorer**) or a table or view (for example **Profile data**). Hover for what the action does and which extension it comes from.
Results grid, right-click	Actions for the result, such as **Chart this result** and **Profile this result**.
SQL editor	SQL Intelligence's completion, **Format** and **Check** bar, folding and **Replace with**, and Operation Preflight's review when you run a statement.

An action opens in a tab of its own, marked with a puzzle-piece icon, with the extension's name under its title. While it works, **Cancel** (**Stop what the view is doing**) is in the tab's header.

What each one does

Extension	What it does, and where to start it	Writes
Schema		
Schema Explorer	A diagram of a connection's tables, views and relationships, declared and inferred. **Tools › Schema explorer** (`Ctrl` + `Alt` + `S`). Exports SVG or PNG.	No
Schema Designer	Drafts new or changed tables and shows the change script, coloured by risk. **Tools › Schema designer**, or **Design this table**.	Yes
Schema Diff	Compares two schemas, live or from snapshot files, and writes the script that makes the target match. **Tools › Compare schemas**.	No
Migration Studio	Keeps versioned migration scripts, checks them, rehearses and applies the pending ones. **Tools › Migration studio**.	Yes
Schema Time Machine	Schema snapshots when you ask, and what changed between two of them. **Tools › Schema time machine**.	No
Documentation Studio	A data dictionary with your notes, as Markdown or HTML. **Tools › Documentation studio**.	No
Queries and the editor		
SQL Intelligence	Context-aware completion, snippets, checks with quick fixes, formatting with a preview, replace, bracket matching and folding, in the SQL editor.	No
Visual Query Builder	Builds one SELECT on a diagram of tables and joins. **Tools › Query builder** (`Ctrl` + `Alt` + `Q`).	No
Query Plan Doctor	Explains a statement as a plan tree, with findings marked as measured, estimate or hypothesis. **Tools › Plan doctor**.	No

Extension	What it does, and where to start it	Writes
SQL Workspaces	Named workspaces of scripts, notes, favourites and saved filters, and an opt-in query history. **Tools › Workspaces**.	No
Global Search	Searches object names, comments and definitions across connections, and, if you opt in, the data. **Tools › Search objects and data**.	No
Data		
Advanced Data Browser	A table with a visual filter builder, server-side paging and viewers for JSON, binary, images and long text. **Browse with filters** on a table.	No
Record Journey	Follows one row to its parent and child rows through foreign keys. **Follow this record** on a result row.	No
Data Comparison	Compares two tables, or a table and a query, by key, across connections. **Tools › Compare data**.	No
Data Synchronisation	Makes a target table match a source, after a preview. **Tools › Synchronise data**.	Yes
Data Profiler	Missing values, uniqueness, top values, distributions and candidate keys. **Profile data** on a table, **Profile this result** on a result.	No
Test Data Generator	Repeatable invented rows for related tables, exported or inserted. **Generate test data** on a table.	Yes
Data Anonymiser	Masks, redacts or pseudonymises columns into a copy, or in place if you choose. **Anonymise data** on a table.	Yes
Data Visualisations	Bar, line, scatter, histogram or pie charts of a result; exports CSV, SVG or PNG. **Chart this result**.	No
Safety and advice		
Operation Preflight	Reviews risky statements from the editor before they run, after Strata's own production guard, never instead of it. Works on its own once enabled.	No
Database Detective	Investigations of repeated rows, plans, improvements, likely relationships and data quality, with optional advice from a language model. **Tools › Database detective**.	No

Writes means the extension can change a database, always through the confirmation below. Every extension can save files you ask it to export, at a path you choose. Each one's full help, limits included, is in `docs/user/extensions/` of the Strata release.

What Strata asks before an extension acts

⃠ **Your data** An extension that changes a database shows its statements first, then Strata asks again in its own dialog: **Run these statements?**, or **Change a PRODUCTION database?** on a Production connection, with the target and every statement. Only **Run** runs them; `Enter` does not. Where the product supports it they run in one transaction, so if one fails none is kept. MySQL and MariaDB commit each DDL statement on its own, so a failure part-way leaves the earlier ones applied. A read-only connection refuses every extension write.

Strata asks	When
Run these statements?	An extension wants to change a database.
Rehearse (rolled back)	Migration Studio's **Rehearse…**: the statements run in one transaction that is always rolled back. PostgreSQL, SQL Server and SQLite only.
Run EXPLAIN ANALYSE?	An extension wants to measure a statement by running it. Strata rolls it back where the product allows, which is not everywhere.
Send this to *host*?	An extension wants to send something over the network. Only Database Detective does, and only for model advice you ask for.
A file dialog	An extension wants to read or save a file. Only the file you choose is used.

⚠ **Careful** A rehearsal takes locks as a real run would, and sequence or identity values it uses are not given back. Rehearse on a test copy when you can.

🚫 **Security** Before **Send**, read the request in the dialog: it shows everything that will be sent, with secret headers hidden. A request to a remote service discloses schema names, SQL and any values in it to that provider, under its privacy terms. Rows and passwords are never sent. Database Detective's default is **None (deterministic only)**, which sends nothing.

What they keep

Feature extensions keep their settings and saved items, such as drafts, snippets, saved queries, workspaces and schema snapshots, in `<data dir>/extension-data/`, one folder per extension. Schema snapshots are not encrypted. To remove what one extension keeps, choose **Delete stored data** on its card in **File › Extensions…**; the extension stays installed. Removing an extension does not delete its stored data.

⚖ **Honesty** The feature extensions were tested on Linux, and on Windows under Wine. They have not yet been checked on macOS or on a real Windows desktop. Findings from Data Profiler, Query Plan Doctor and Database Detective are evidence to read, not conclusions, and Data Anonymiser does not guarantee that its output cannot be traced back to people.

✅ **Done when** **Tools** lists the extension's actions and one of them opens in its own tab.

8.4 Get Data Laboratory

How you get it depends on which edition of Strata you have.

Your edition	What to do
Complete + Data Laboratory	Nothing. It is already in the `extensions` folder beside the program and shows as **bundled with Strata**.
Core or Complete	Install the separate Data Laboratory package, a folder called `cx.dixon.data-laboratory`, as in the next section.

On a Mac, a `Strata.app` that comes with Data Laboratory carries it inside the app, so there is nothing to install. Otherwise install the package in your own extensions folder, as below.

8.5 Install an extension

On Windows and Linux you can also add a driver extension with the installer or package for your platform, as you installed Strata. To install an extension folder by hand:

> 🚫 **Security** Extensions are programs. An extension runs inside Strata with your permissions and is **not sandboxed**. For that reason Strata loads only extensions signed by the Strata project, checks each extension's files against its signed manifest before running it, and never downloads anything. There is no marketplace, account or telemetry. Install extensions only from a Strata release.

1. Open **File › Extensions…**.
2. Choose **Show folder**. Your file manager opens the extensions folder (Strata creates it if it is missing).
3. Copy the extension's whole folder into it, exactly as the release provides it. Inside are three things: the extension's program file, its `manifest.json` and the signature, `manifest.json.sig`.
4. Back in Strata, choose **Re-scan**. You do not need to restart.

Platform	Extensions folder
Linux	`~/.local/share/strata/extensions/`
macOS	`~/Library/Application Support/Strata/extensions/`
Windows	`%APPDATA%\Strata\extensions\`

👁 **You should see** a card for the extension with its name, version, publisher, capabilities and location, and the status **Enabled**: **Verified; it loads when a feature first needs it.**

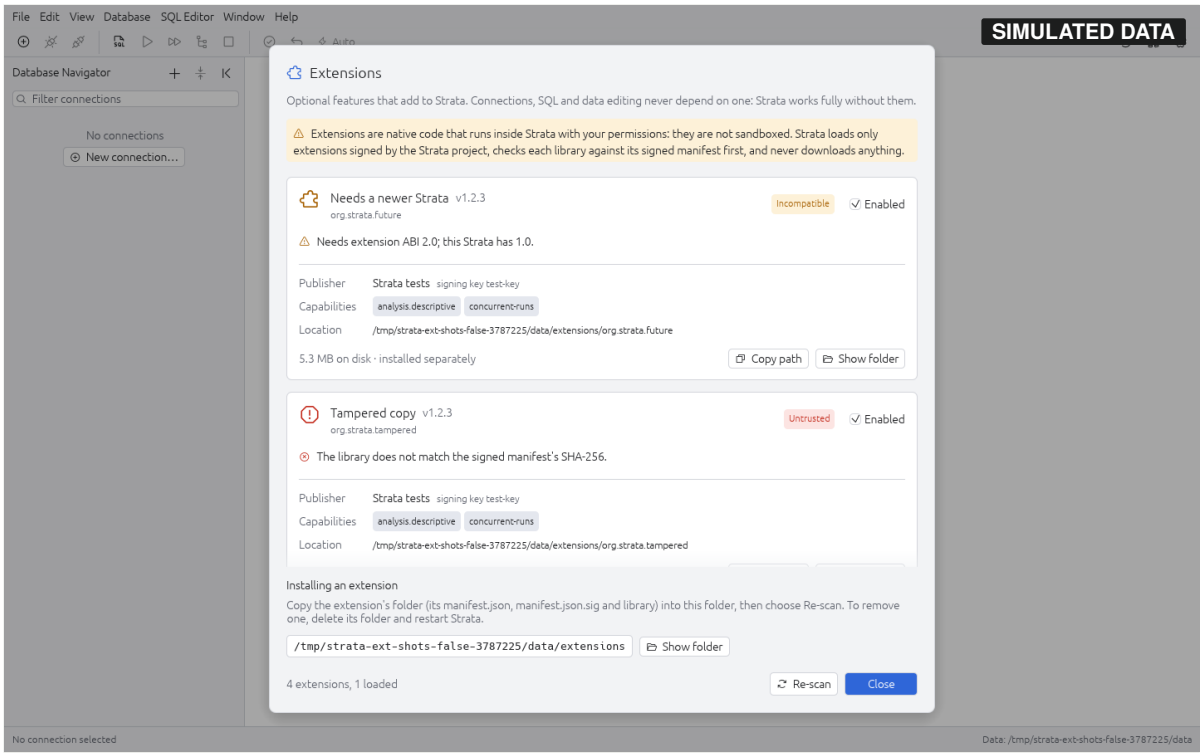


Figure 8.1. The extension manager: one extension needs a newer Strata, one has been changed since it was signed. *Simulated data.*

To check that it works straight away, choose **Load now** on its card.

↺ **If not** the card shows another status and a reason. See If an extension is unavailable below.

✅ **Done when** the card shows **Enabled** or **Loaded**.

8.6 Your first analysis

This walk-through looks for unusual CPU load in a table of metrics with a timestamp `ts`, a host name `host` and a cpu reading. Use any result shaped like it.

1. In an SQL editor, run the query, for example:

```
select ts, host, cpu, errors, failed from metrics order by ts, host
```

2. When it has finished, choose **Analyse** on the results bar, to the right of the row count.

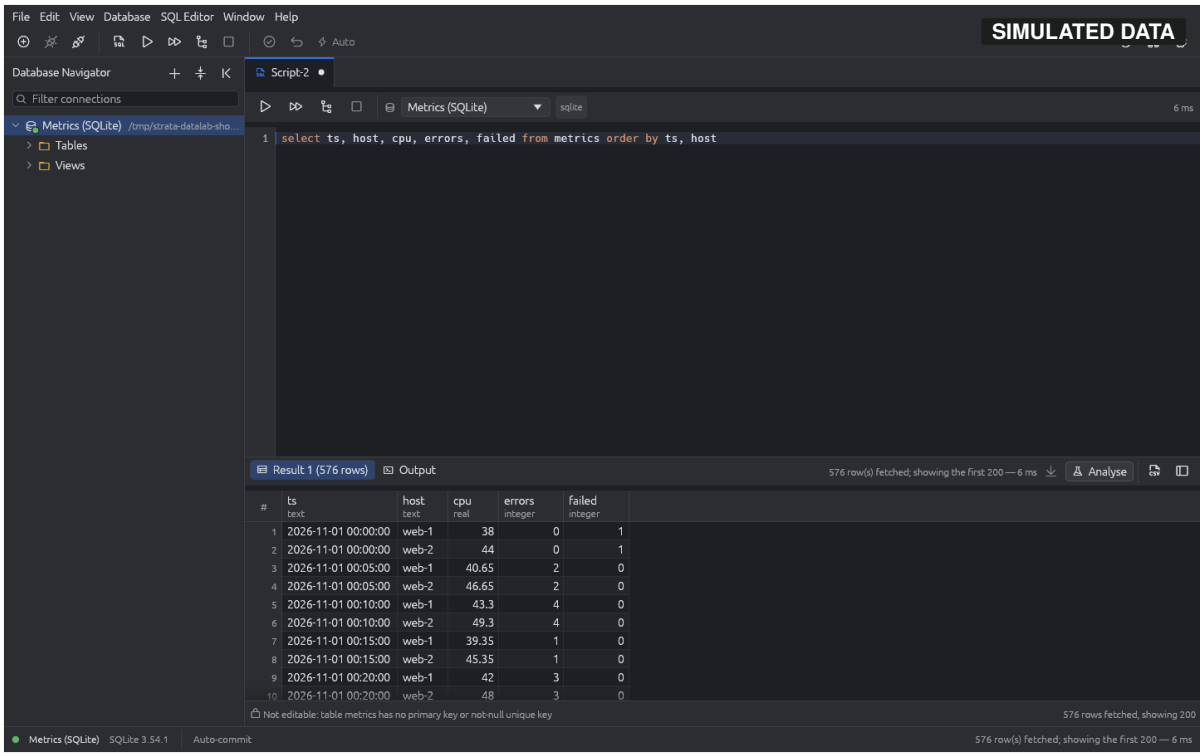


Figure 8.2. The results bar, with **Analyse** to the right of the row count. *Simulated data.*

👁 **You should see** a new tab, **Lab · *your editor's name***. The first time, it shows **Starting Data Laboratory…** for a moment while the extension loads.

> ℹ **Why this matters** The tab works on a copy of the result. Your editor, its grid and any unsaved edits stay exactly as they were, even if you cancel. Nothing is written back to the database.

3. Check the **Columns** panel on the left. Strata guesses: **Time axis** is `ts`, **Values** has `cpu` ticked, and **Group by** has `host` ticked, so each host gets its own series. Change anything that is wrong; **Auto-detect** guesses again.
4. Under **Analysis**, leave **Analyse** selected, choose a **Detector** (here **Rolling baseline**) and a **Preset** (**Balanced** is the default).
5. Choose **Run the detector**.

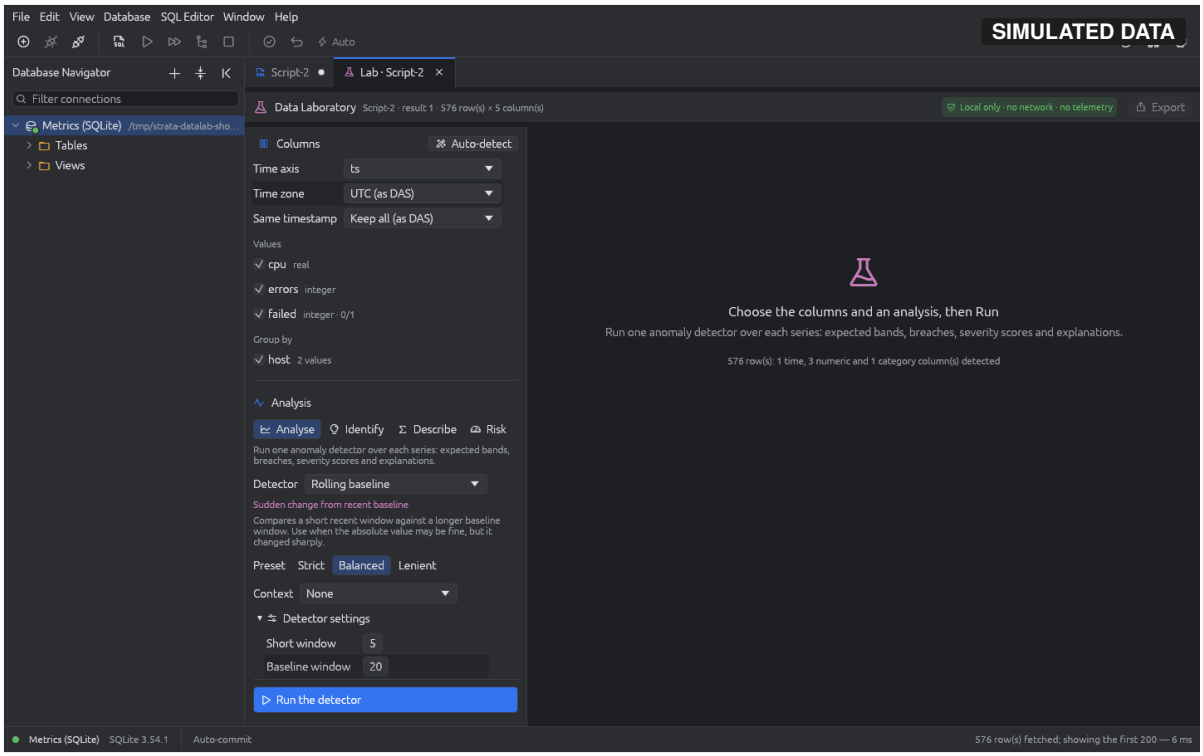


Figure 8.3. The set-up panel: columns on the left, the analysis and its settings below, and **Run the detector** at the bottom. *Simulated data.*

👁 **You should see** a progress bar and **Cancel** in place of the button, then the results on the right.

6. Read the results, from the top:
   - **Summary tiles: Series**, **Findings**, **Highest severity** and **Affected columns**.
   - **Chart:** the series with its **Expected band** shaded, the **Expected** value dashed, each breach as a coloured span and its **Worst point** marked. Scroll to zoom, drag to pan and double-click to reset. With several series, click one in the **Series** list to chart it.
   - **Findings:** most severe first, each with its score, time range, worst value and a plain-English explanation.
7. Click a finding. It is highlighted on the chart, and a table shows how its score is made up (**Component**, **Weight**, **Points**, **Why**).

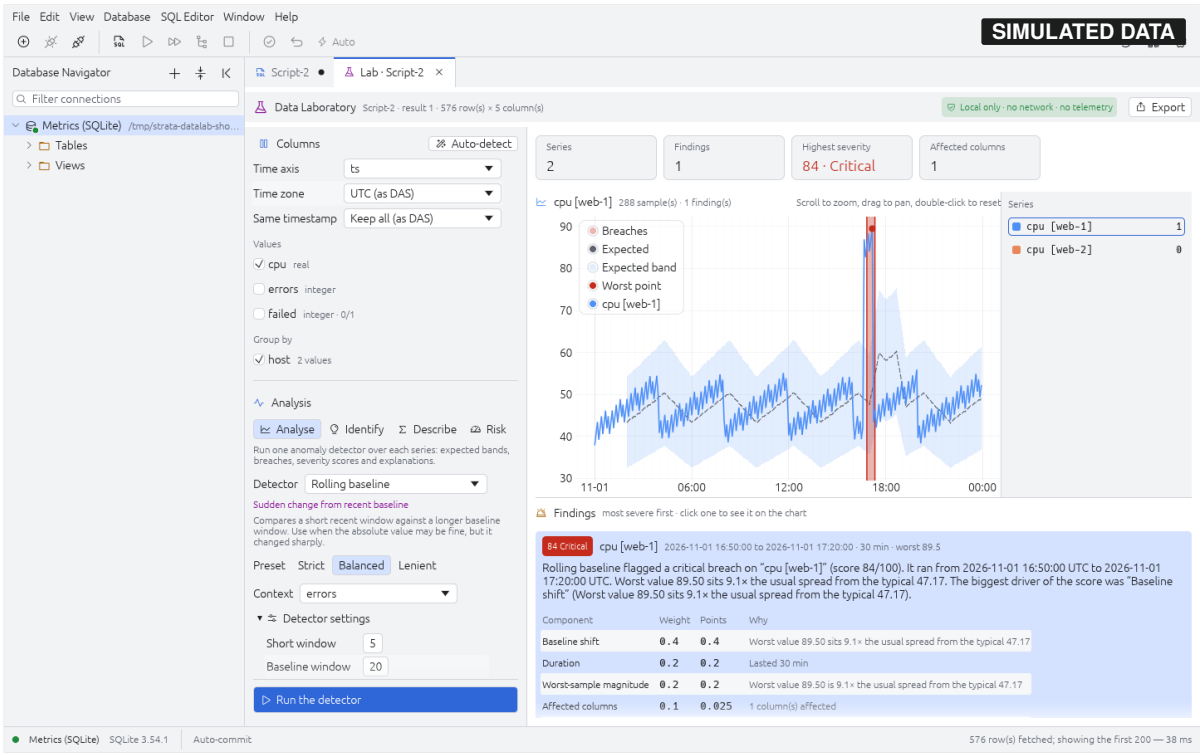


Figure 8.4. Results: summary tiles, the chart with its expected band and one critical breach, and the finding with its explanation. *Simulated data.*

Done when the **Findings** list shows at least one finding, or the message **No breaches: every series stayed within what this detector expects.**

Next try another analysis or export the findings, below.

Severity

Each finding has a score from 0 to 100 and a band.

Band	Score	Read it as
Critical	80–100	Look at this first
High	60–79	Clearly unusual
Moderate	35–59	Worth a look
Low	1–34	Slightly outside the expected range
Informational	0	Noted, not scored

Honesty Findings are statistical estimates, not proof that something is wrong. Treat a finding as a reason to look, not a conclusion. Data Laboratory's results are checked against a reference analysis engine. The risk classifier's quality figures are measured on the same data it was trained on, so they flatter it.

8.7 Choose an analysis

The **Analysis** section offers four choices. Each changes the run button's label.

Choice	Button	What it does
Analyse	**Run the detector**	Runs one detector over each series: expected bands, breaches, severity scores and explanations.
Identify	**Suggest a detector**	Studies the data and suggests a configured detector, with a confidence, the signals behind it and alternatives. **Use and run** applies the suggestion; **Use** applies an alternative.
Describe	**Describe the columns**	Count, missing values, mean, standard deviation, quantiles and a histogram per column. Suits any result, time series or not.
Risk	**Train and score**	Trains a risk classifier on a column holding only 0 and 1 (**Label (0 / 1)**) and scores every row. The other ticked value columns are its features.

Tip Not sure which detector to use? Run **Identify** first, then **Use and run**. If a result has no time column, Strata starts on **Describe**.

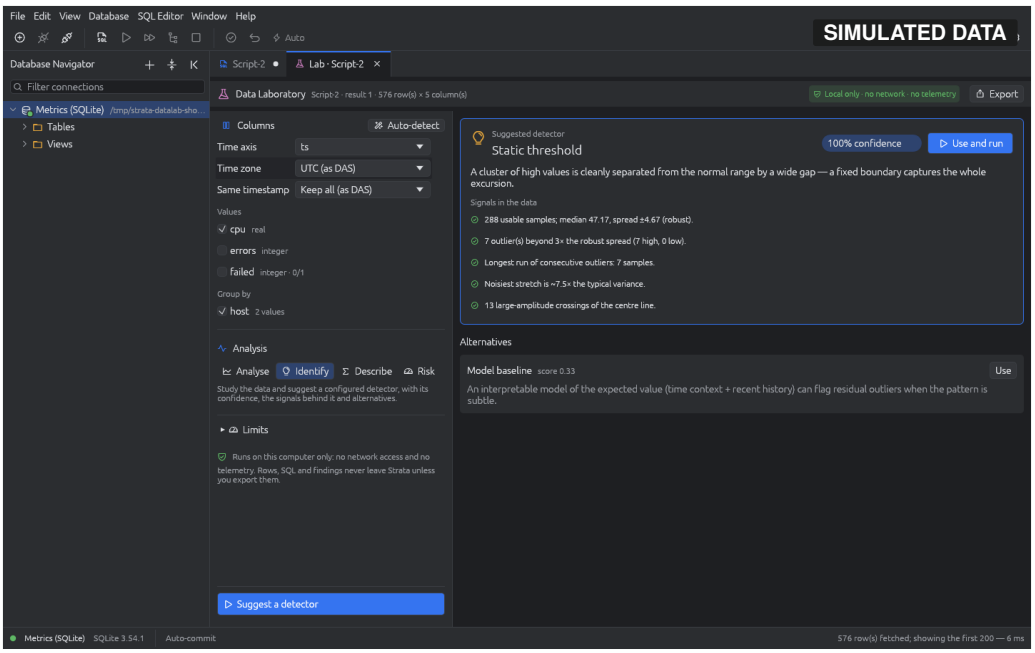


Figure 8.5. Identify: the suggested detector, its confidence, the signals in the data and the alternatives. *Simulated data.*

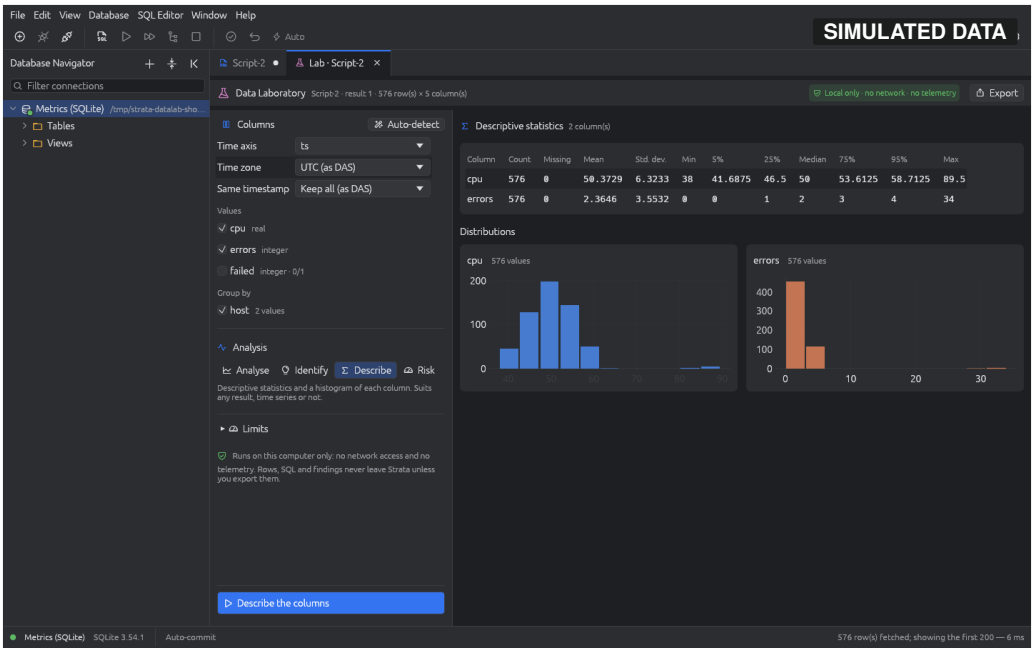


Figure 8.6. Describe: statistics and a histogram for each value column. *Simulated data.*

Settings you may change

Detector One of 13, from **Static threshold** to model baselines. A description of each appears under the list. Seasonal comparison and missing data need a time column; the list says **Needs a time column**.

Preset **Strict** fires more readily, **Lenient** less.

Detector settings Every setting of the chosen detector, already filled in.

Context Optionally, a second column whose mean during a breach feeds the severity score.

Successes, Totals The two count columns the binomial detector needs.

Time zone How date-times without a UTC offset are read. The default is **UTC (as DAS)**. Values that carry an offset are always exact.

Same timestamp What to do when rows of one series share a timestamp: **Keep all (as DAS)**, **Average**, **Sum**, **First** or **Last**.

8.8 Cancel, limits and messages

Cancel stops a running job. You should see **Analysis cancelled. The query results are unchanged.** The previous results stay on screen.

Every job has three limits, under **Limits** in the set-up panel. A job over a limit stops with a message instead of slowing Strata down.

Limit	Default	When reached
Rows	200,000	The most recent rows are analysed, with a warning
Memory (MB)	1,024	The job does not start
Time (s)	120	The job is stopped

Message	What to do
Wait for the statement to finish, then analyse its result.	Shown on the greyed-out **Analyse** button. Wait, or cancel the statement.
The result has no rows to analyse.	Change the query.
No numeric columns: Data Laboratory analyses numbers.	Select at least one numeric column, or cast one in SQL.
Choose at least one value column.	Tick a column under **Values**.
Risk needs a column holding only 0 and 1 (the label), and no column here does.	Add a 0/1 column to the query, or choose another analysis.
The data has *n* rows; only the most recent *m* were analysed (row limit).	Nothing, or raise **Rows**.
The result stopped at the fetch limit, so only its first *n* rows are analysed.	Raise the fetch limit in Settings, or narrow the query.
***n* value(s) in *column* are not numbers and were treated as missing.**	Check the column; this is a warning only.
***n* value(s) in *column* could not be read as a date or time and were treated as missing.**	Check the column's format, or choose another **Time axis**.
***n* value(s) in *column* have more precision than a 64-bit float holds and were rounded.**	Nothing for most data; a warning only.
The analysis ran past its time limit (120 s) and was stopped.	Raise the limit, choose fewer rows, or set a retraining interval for model baselines.
The analysis would need about *n* MB of memory; the budget is *m* MB.	Select fewer rows or columns, or raise **Memory (MB)**.
The analysis would build *n* series; the limit is *m*.	Select fewer columns or groups.

Message	What to do
No breaches: every series stayed within what this detector expects.	Try the **Strict** preset or another detector.

8.9 Export the results

Export, in the tab's header, saves one file through the usual path prompt:

- **Findings as CSV…** (or **Statistics as CSV…** after Describe): one row per finding, most severe first. The formula guard is on, as for other exports.
- **Full result as JSON…**: everything, including series, bands, findings and warnings.

Before the first run it is greyed out: **Run an analysis first**.

8.10 What stays on your computer

Why this matters Data Laboratory runs only inside Strata. It has no network code, writes no files and sends nothing anywhere: rows, SQL and findings stay on your computer unless you export them. The tab says so in its header: **Local only · no network · no telemetry**. A job runs only when you start it, never in the background or on a schedule.

For extensions in general, Strata itself downloads nothing and reports nothing about them. It stores which extensions you have turned off in `settings.json`; feature extensions keep what they save in `extension-data/`. Of all the extensions, only Database Detective can use the network, and only after you choose **Send**.

8.11 Turn extensions on and off

Each card in **File › Extensions…** has an **Enabled** tick box. A newly installed extension is enabled. An enabled extension is not loaded at start-up: it loads, in the background, when a feature first needs it. A disabled one is never loaded: for Data Laboratory, the **Analyse** button is greyed out with the reason; for a driver extension, its products and connections are greyed; a feature extension's actions leave the menus.

When you need to restart

Strata never unloads an extension while it runs, so only changes to a loaded one need a restart.

You…	Restart?
Install an extension	No: choose **Re-scan**
Enable an extension	No
Disable one that is not loaded	No
Disable one that is loaded	Yes: the card says **Restart required**
Remove or replace an extension's folder	Yes, if it was loaded
See **Failed to load**, **Failed to initialise** or **Failed at run time**	Yes

To remove an extension, delete its folder and restart Strata.

8.12 If an extension is unavailable

Whatever happens to an extension, the rest of Strata keeps working. Only its feature is affected. For a driver extension, its products and connections are greyed, with the reason (Driver extensions). For Data Laboratory, the **Analyse** button is greyed out, and hovering over it gives the reason, for example **Data Laboratory is untrusted: … See File > Extensions… for details.** If the tab is already open, it shows **Data Laboratory is not available** with the same reason.

Status	Meaning	What to do
Enabled	Signature and manifest check out; it loads when first needed	Nothing
Loaded	Running in this session	Nothing
Disabled	You turned it off	Tick **Enabled**
Restart required	Turned off while loaded	Restart Strata
Incompatible	Made for another Strata version, extension interface, platform or processor	Install the version released with your Strata
Untrusted	Not signed by the Strata project, or changed since it was signed	Reinstall it from the Strata release
Invalid	A missing file, a symbolic link or a wrong name in the folder	Reinstall it
Duplicate	Installed twice; neither copy loads	Delete one copy
Missing dependency	Needs another extension that is missing or cannot load	Install or fix that one first

Status	Meaning	What to do
Not installed	Removed after the last scan	Choose **Re-scan**
Failed to load, **Failed to initialise**	The extension would not start	Restart; if it fails again, reinstall
Failed at run time	It hit an internal error and is no longer called	Restart Strata

> ⚠ **Careful** Never edit an extension's `manifest.json` or rename its files. Any change makes it **Untrusted**, and the only fix is to copy the folder again from the release.

Compatibility

Each extension's signed manifest records the Strata version, extension interface, platform and processor it was made for. Strata checks these before loading anything. An extension made for a newer Strata, or for another platform, shows **Incompatible** with the reason, for example **Needs extension ABI 2.0; this Strata has 1.0.** After you update Strata, an extension that still matches keeps working; if one turns **Incompatible**, install the version released with your Strata.

Diagnose a loading failure

1. Open **File** › **Extensions…** and read the reason on the card.
2. If Strata will not show the window, run this in a terminal instead:

   ```
   strata --diagnostics
   ```

3. Look for the `Extensions:` lines at the end. Each lists an extension, its folder and its state, without loading anything:

   ```
   Extensions:
     cx.dixon.data-laboratory (<folder>): valid
   ```

   Anything other than `valid` is the reason, such as `untrusted: …` or `incompatible: …`. Match it to the status table above.

✓ **Done when** the card shows **Enabled** or **Loaded**, or the diagnostics line says `valid`.

→ **Next** if it still fails, send the diagnostics report to support (Collect a diagnostics report).

CHAPTER 9

When something goes wrong

TL;DR

Find the symptom below. Each entry gives the likely cause, the safest first check and what to do. If none fits, collect a diagnostics report (last section) and contact support.

Starting Strata

The licence window will not go away

Likely cause: this computer is not activated, the trial has ended, or the lease has expired offline. **Check:** the sentence at the top of the window; Chapter 7 lists every one and what to do. If it says **Licensing is unavailable:** followed by a reason, the licensing component could not start and the reason says why. Check that `<data dir>/licensing/` exists and is writable, then restart Strata. While the window is open, Strata's keyboard shortcuts and any editor in its own window do nothing; its buttons say **Licensing isn't available in this session. Restart Strata to try again.** when licensing could not start.

"The licensing server cannot be reached"

Likely cause: no network, a firewall or proxy blocking `https://licensing.uncomplex.co.uk`, or the service is down. **Do:** check that a browser on the same computer can open that address, then choose **Check licence**. Strata keeps working on its offline lease for up to seven days after the last successful check.

A banner says the data directory is read-only

Likely cause: Strata cannot write to its data directory, or you passed a read-only `--data-dir`. Preferences then apply to this session only, and new connections and passwords cannot be saved. **Do:** fix the folder's permissions, or start Strata with a writable `--data-dir`.

Strata asks for a store password at start-up

You protected saved passwords with a password (Section 2.8). Type it and choose **Unlock**. **Continue without** works, but connections then ask for their passwords and nothing new is saved until **File › Unlock saved passwords…**. A forgotten store password cannot be recovered.

Saved passwords cannot be opened after a system change

Likely cause: the OS keyring holding the key is unavailable (locked, reset, or a different user session). Strata deliberately does not fall back to a weaker key. **Do:** unlock or restore the keyring, then restart Strata.

Connecting

The product is greyed out in "Connect to a database"

Hover over the card. A card badged **needs extension** needs a driver extension that is not installed, is disabled or cannot load; the reason ends with what to do, such as **Install it, then see File › Extensions…** or **Enable it in File › Extensions…**. Install or enable it (Section 8.2). A card badged **not implemented** is a *Catalogued* product: no driver exists yet.

A connection shows "unavailable" in the navigator

Its driver extension is missing, disabled or cannot load, for example after importing connections from DBeaver or copying them from an installation with more extensions. Hover for the reason, which starts **Can't connect:**. The connection is kept unchanged; it works again once its driver extension can load (Section 8.2).

"Unknown SSH host key"

Expected the first time. Compare the fingerprint with one from the server's administrator before choosing **Trust and connect** (Section 2.4).

“SSH host key CHANGED: not connecting”

> **Security** Stop. Someone may be intercepting the connection. Ask the server’s administrator whether the key was replaced. Only if it was, remove the line the message names from that `known_hosts` file and connect again.

“TLS certificate verification failed”

Likely cause: the server’s certificate is from a private CA, has expired, or does not match the host name you typed. **Do:** put the CA’s certificate in **CA certificate** on the **SSL / TLS** tab, and connect with the name the certificate was issued for. Do not switch to `required` to make the error go away unless you understand that the server is then not verified.

Password asked for every time

Save password is not ticked on the connection, or saved passwords are locked. Tick it in **Edit connection…**, or unlock the store.

“Certificate pins need TLS mode required, verify-ca or verify-full”

Pins cannot be checked on a connection that may fall back to plaintext. Choose one of those modes, or clear the pins.

Running SQL

Only 200 rows, or a tab title ending in “+ rows”

The fetch limit stopped reading. Use **SQL Editor › Fetch more rows**, raise the limits in **File › Settings…**, or export (Chapter 4).

Explain plan is greyed out

The connection’s database cannot explain, or the editor has no connection. The toolbar tooltip says which.

A statement asks “Change every row on a production connection?”

Working as intended: an UPDATE or DELETE without WHERE on a Production connection. Read the statement again. Choose **Cancel** unless you meant every row.

“uses a query language Strata can’t run yet”

The connection’s product has no editor language in Strata yet. Browse it in the navigator instead.

The script stopped and says the transaction is lost

The connection was reset mid-transaction. Nothing after the reset ran, and the open transaction’s changes were rolled back by the server. Reconnect and run the script again from the start.

Results and data

The grid says “Not editable”

The reason follows: the result comes from more than one table, the table’s key is not in the result, the connection is read-only, or the driver cannot edit rows. Select the key columns too, or edit with SQL.

“Import CSV…” is greyed out

Can’t import: the connection is read-only or its driver can’t import. Clear **Read-only**, or import through a driver that supports it.

A spreadsheet shows a leading apostrophe in exported cells

The formula guard is on (Chapter 5). It is protecting you; turn it off only for files that will not be opened in a spreadsheet.

Extensions

There is no Analyse button, or it is greyed out

Likely cause: no button means Data Laboratory is not installed; a greyed-out one means the statement is still running, or the extension is disabled or cannot load. **Check:** hover over the button for the reason. See If an extension is unavailable.

The Tools menu says No feature extensions are enabled.

Likely cause: you have the Core edition, the feature extensions were left out when Strata was installed, or they are turned off. **Do:** open **Tools › Extensions…**, install or tick **Enabled** on the ones you want (Section 8.3).

A feature's tab says "*name* stopped"

Likely cause: the extension hit an error. Its other tabs, and the rest of Strata, keep working. **Do:** close the tab and choose the action again. If it stops again, restart Strata: a library that failed is reloaded only after a restart.

An extension is "Untrusted", "Incompatible" or "Invalid"

The card gives the exact reason. See the status table in Chapter 8. Never edit an extension's `manifest.json`: that alone makes it untrusted.

9.1 Collect a diagnostics report

```
strata --diagnostics
```

This prints Strata's version, the data directory, your connections (without host names), where the credentials key comes from, recent error kinds and the installed extensions. It writes nothing and loads no extension. Add `--with-hosts` only if support asks for host names.

The log is `<data dir>/logs/strata.log`. By default it holds only Strata's own events; events from the libraries Strata uses are left out, because they can carry host names, addresses and server messages. Set the environment variable `STRATA_LOG` to `strata=debug` for more detail:

```
STRATA_LOG=strata=debug strata
```

Why this matters Strata keeps passwords, SQL text and result data out of the log and the report. Before you send either, still read it through: connection names, folder

> names and file paths can say more about your organisation than you intend. Remove anything you would not put in an email.

Where to send it: Appendix C.

APPENDIX A

Keyboard shortcuts

Checked against Strata at this guide's revision. **Help › Keyboard shortcuts & about** shows the main ones in the app.

On a Mac: where this table says `Ctrl`, press `Cmd`, except for `Ctrl` + `Tab` and `Ctrl` + `Shift` + `Tab`, which use the Control key on every platform. Menus show `Ctrl` on every platform.

Feature extensions can bring their own shortcuts. The **Tools** menu shows each one beside its action; a shortcut that would clash with Strata's own, or with an earlier extension's, is not assigned.

Keys	Action
Anywhere	
`Ctrl` + `Shift` + `N`	New connection
`Ctrl` + `]`	New SQL editor on the selected connection
`Ctrl` + `O`	Open an SQL script
`Ctrl` + `S`	Save the SQL script
`Ctrl` + `W`	Close the tab
`Ctrl` + `Tab` /	Next / previous tab
`Ctrl` + `Shift` + `Tab`	
`Ctrl` + `,`	Settings
`Ctrl` + `Q`	Quit; asks first about unsaved editors and open transactions
`F5`	Refresh the selected navigator node
SQL editor	
`Ctrl` + `Enter`	Execute the statement at the cursor (or the selection)
`Alt` + `X` or	Execute the script; one result tab per statement
`Ctrl` + `Shift` + `Enter`	
`Ctrl` + `Shift` + `E`	Explain the statement's plan
`Esc`	Cancel the running statement
`Ctrl` + `/`	Comment or uncomment lines
`Ctrl` + `Space`	Complete the word at the cursor
`Ctrl` + `F`	Find in the script; `Enter` next match, `Esc` close
`Shift` + `Alt` + `F`	Format the selection or statement (SQL Intelligence)

Keys	Action
`Shift` + `Alt` + `L`	Check the script (SQL Intelligence)
Feature extensions	
`Ctrl` + `Alt` + `S`	Schema explorer
`Ctrl` + `Alt` + `Q`	Query builder (Visual Query Builder)
Results grid	
Arrows, `Page Up`, `Page Down`	Move; with `Shift`, extend the selection
`Home` / `End`	First / last column
`Ctrl` + `Home` / `Ctrl` + `End`	First / last cell
`Ctrl` + `A`	Select all
`Ctrl` + `C` / `Ctrl` + `Shift` + `C`	Copy cells / with column names
`F2` or `Enter`	Edit the cell (editable results)
`Del`	Delete the selected rows (editable results)

⚖ **Honesty** On a Mac, `Cmd` + `Space` is usually taken by Spotlight before Strata sees it, so completion may need Spotlight's shortcut changed.

APPENDIX B

Glossary

Auto-commit Each statement commits on its own. The alternative is manual commit, where statements wait for **Commit** or **Rollback**.

CA (certificate authority) Whoever signed a server's TLS certificate. A private CA's certificate goes in **CA certificate** so Strata can verify servers it signed.

Catalogue Strata's list of every product it targets, with test status and limitations (**View › Database catalogue**).

Change set The statements that one feature extension proposes to run, such as Schema Designer's edits or Migration Studio's migration. Nothing in it runs until you confirm.

Connection Saved settings for reaching one database: product, host, account, network route and TLS. Not the same as a session, which is an open connection.

Connection type **Development**, **Test** or **Production**. Production adds a red marker and asks before UPDATE or DELETE without WHERE.

Core edition, Complete edition Strata with its own drivers only, and Strata with a driver extension for every further group of databases.

Data directory The folder where Strata keeps connections, encrypted secrets, settings, extensions and what they save, the licence lease and the log.

Data Laboratory An optional, signed extension for analysing results: anomaly detectors, bands, severity scores and explanations.

Detector In Data Laboratory, one method of deciding what is unusual in a series, such as a rolling baseline or a static threshold.

Display limit The most rows the grid shows at once (200 by default).

Driver The part of Strata that speaks one product's protocol. Drivers come in groups; the Core groups are part of Strata, the others come as driver extensions.

Explain plan The database's description of how it will run a statement. *Explain analyze* runs the statement to measure it.

Exploration tested A product whose connecting, querying and browsing passed live tests against a real server. This guide's meaning of *supported*.

Extension A signed add-on: a group of database drivers (a *driver extension*), a tool such as Schema Explorer (a *feature extension*), or Data Laboratory. Every extension is a program running inside Strata. Not sandboxed.

Feature extension A signed tool that works through Strata rather than on its own, such as Schema Explorer or Data Comparison. Its actions are in the **Tools** menu.

Fetch limit The most rows Strata reads for one result (200 by default). The rest is not read unless you fetch more or export.

Finding In Data Laboratory, one stretch of a series outside its expected range, with a severity score from 0 to 100 and an explanation.

Fingerprint A short digest of an SSH server's key (`SHA256:…`), to compare before trusting it.

Formula guard The export option that prefixes cells starting with =, +, − or @ with ' so spreadsheets treat them as text.

known_hosts A file of trusted SSH server keys. Strata reads yours and writes its own.

Lease The signed record that this computer is licensed, valid for at most seven days and renewed online about daily.

Navigator The **Database Navigator** panel on the left: connections, schemas, tables and columns.

Pin (certificate pin) A SHA-256 digest of a server's public key that the server must match, in addition to normal TLS checks.

Read-only A connection option that asks the database to refuse writes. A guard against mistakes, not a permission.

Rehearse To run a change set in one transaction that is always rolled back, to see each statement's outcome without keeping anything. PostgreSQL, SQL Server and SQLite only.

Simulated data Invented data used in a screenshot. Every screenshot in this guide is simulated.

SSH tunnel A route to a database through an SSH server you can log in to.

TLS Transport Layer Security: encryption for a connection, and, in verifying modes, proof of the server's identity.

APPENDIX C

Version and support notes

C.1 Which version this guide describes

Field	Value
Product	Strata, by Uncomplex
Guide	Strata — User Manual
Guide revision	3
Last verified	22 November 2026
Release	`1.0.0` (**Help › Keyboard shortcuts & about** shows *Strata 1.0.0*)
Status	Release
Applies to	Linux, macOS and Windows; Core and Complete editions
Licence	Proprietary (EULA); includes Apache-2.0 components

When you report a problem, give the version from **Help › Keyboard shortcuts & about** *and* your edition, and how you got Strata.

C.2 How it was checked

Every label, menu path, shortcut and message quoted in this guide was checked against Strata at this revision. Product support comes from Strata's database catalogue, which records the live tests each product passed. Screenshots show the real interface with invented data. Automated tests run on Linux, macOS and Windows; live tests against real database servers ran on Linux only.

C.3 Licence

Strata is proprietary software licensed under Uncomplex's Master Software Licence and Services Agreement, which you accept in the licence window (**Read terms**). It includes open-source components under the Apache License 2.0 and other licences; fonts bundled with the interface are under the SIL Open Font Licence and the Ubuntu Font Licence.

To read the full list of third-party components and their licences, open **Help › Keyboard shortcuts & about** and expand **Third-party licences**. The list covers every edition and extension, so your installation contains a subset of it.

C.4 Getting help

For help with Strata or your licence, including a licence in use on too many computers, email `hello@uncomplex.co.uk` or visit `https://uncomplex.co.uk`. The legal contact, `legal@uncomplex.co.uk`, is not a support channel.

Security problems: report them privately to `security@uncomplex.co.uk`, never in a public forum.

Before you contact anyone: collect `strata --diagnostics` (Section 9.1), read it, and remove anything you would not want to share.

APPENDIX D

Limitations

This appendix collects what Strata does not do, or has not been shown to do, in this revision. Each chapter says the same things where they matter; here they are in one place.

D.1 Honesty

Honesty

- **Testing is narrow.** "Exploration tested" means one version of a product passed Strata's live suite, nearly always against a test server on Linux. Managed cloud services (Amazon RDS and Aurora, Azure, Google Cloud SQL, Neon, Supabase and others) connect with the same drivers but have not been tested. Other versions and operating systems were not tested against real servers.
- **Keep backups.** Back up anything you change through Strata, as with any tool that writes to a database.
- **Feature extensions and the editor's own window** were checked on Linux and on Windows under Wine, not yet on macOS or a real Windows desktop.
- **Screenshots are simulated.** Every image in this guide shows invented data.
- **The PDF is not tagged for assistive technology.** It has bookmarks, real text, linked contents and document language set, but it is not a tagged, accessible PDF. Ask Uncomplex for the content in another format if you need one.

D.2 Connections and security

- **Read-only** is a guard, not a permission boundary; SQL Server connections cannot be read-only at all.
- No jump hosts, SSH through a proxy, keyboard-interactive or two-factor SSH logins, or SSH certificates. The SSH agent works on Unix only.
- No use of the operating system's certificate trust store; private CAs go in **CA certificate**. MySQL cannot check certificate pins.
- With the key-file fallback (no OS keyring), anyone who can read the data directory can read your saved secrets. Protect the store with a password on such machines.
- Passwords held in memory while Strata runs are not wiped.
- The local end of an SSH tunnel can be reached by other processes on the same computer for as long as the connection is open.

D.3 SQL and results

- A result is read up to the fetch limit (200 rows by default); **Fetch more rows** and export run the statement again.
- Statements that commit implicitly on the server (DDL on MySQL) are not detected by the transaction counter.
- Editing needs a single-table result with a primary or unique key.
- Query history lasts for the session only.
- SQLite connections do not run multi-statement batches as one call; run the script with `Alt` + `X`, which runs statements one by one.
- CSV import is tested end to end on SQLite only.
- Folding needs the SQL Intelligence extension.

D.4 Feature extensions

Each extension's help page lists its own limits; the main ones are:

- Extensions run inside Strata and are not sandboxed. Strata's checks of what they ask for guard a cooperating extension; your database permissions are the real boundary.
- Rehearsal works on PostgreSQL, SQL Server and SQLite only. MySQL and MariaDB commit DDL as it runs, so a failed change script can leave part of it applied.
- Schema Time Machine's snapshots are stored unencrypted, and taken only when you ask.
- Migration Studio has no down scripts or baselining.
- Data Synchronisation has no transaction across two servers.
- The Visual Query Builder makes one SELECT of at most 16 tables, without subqueries or UNION. Right and full joins need SQLite 3.39 or later; MySQL has no full join.
- PNG exports draw characters outside Latin, Greek and Cyrillic, such as Chinese, as ?. No extension exports PDF.
- Plan Doctor and Database Detective's editor entries are not shown yet; start them from **Tools**.

D.5 Files

- No indexes: every query scans whole files; large files are re-read per scan.
- Not supported: `.xls`, XLSB, ODS, ORC, Iceberg, Delta Lake, Hudi, Protocol Buffers, password-protected Access files. Access `.accdb` is untested.

D.6 Products not offered

Some products are deliberately not offered yet, for example Apache Arrow Flight SQL, LanceDB, RethinkDB, MarkLogic, Apache RocketMQ, LMDB and RocksDB, SQream and Ocient. Products that can only be reached through a vendor's own client software are not in the catalogue at all.

D.7 Licensing

- Strata does not run without a licence or an active trial.
- It needs a successful online check at least every seven days.
- Activation sends hardware identifiers and the computer's name to Uncomplex's licensing service (Section 7.4).

Connect to something. Run one query.
Read the result before you change it.

uncomplex. Complicated things. Thoughtfully simplified.