

Databases: PostgreSQL, MySQL, SQL Server, Oracle Database and MongoDB

From the SourceLace manual: How to connect your own databases for read-only questions. Each person signs in with their own database login, so the database's own grants decide what they see.

SourceLace version 0.1.0 · Made October 4, 2026

The latest version is at <https://sourcelace.onrender.com/docs>

Databases: PostgreSQL, MySQL, SQL Server, Oracle Database and MongoDB

How to connect your own databases for read-only questions. Each person signs in with their own database login, so the database's own grants decide what they see.

Status: in testing. Built against each database's documented behaviour, and tested against a real PostgreSQL and simulated MySQL, SQL Server, Oracle Database and MongoDB servers; not yet used with a customer's live database.

Source	Kind	Covers
SQL database	database	PostgreSQL (including Amazon RDS, Aurora, Azure and Cloud SQL), MySQL and MariaDB, SQL Server and Azure SQL, and Oracle Database 12.1 or later (including Oracle Autonomous Database and Amazon RDS for Oracle). The admin picks the engine.
MongoDB	mongodb	MongoDB, including MongoDB Atlas.

Everything here is **read-only**: `propose_change` refuses changes to database sources.

How sign-in works

Databases have no "Sign in with..." page, so **each person types their own database username and password** into a sign-in form served by SourceLace. There is no shared SourceLace login to your database.

- The sign-in link works once, for 10 minutes, and stops after three wrong passwords.
- SourceLace keeps the password encrypted with your organization's own key, uses it only for that person's requests, and never logs it, shows it, or puts it in the audit trail.
- If the password changes later, SourceLace asks the person to connect again.

1. For your database administrator

Give each person their own read-only login

Create one login per person, with read access only. A group role makes this easy to manage. Narrow it as far as you like (some schemas, tables or collections): SourceLace only ever sees what the person's login can see.

PostgreSQL

```
-- Once: a group that may read the tables people should see.
CREATE ROLE sourcelace_readers NOLOGIN;
GRANT CONNECT ON DATABASE sales TO sourcelace_readers;
GRANT USAGE ON SCHEMA public TO sourcelace_readers;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO sourcelace_readers;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO sourcelace_readers;

-- For each person:
CREATE ROLE ann_lee LOGIN PASSWORD 'a long password' IN ROLE sourcelace_readers;
ALTER ROLE ann_lee SET default_transaction_read_only = on;
```

Use hostssl (not host) lines in pg_hba.conf for these logins, so they can only connect over TLS.

MySQL and MariaDB

```
-- Once:
CREATE ROLE 'sourcelace_reader';
GRANT SELECT, SHOW VIEW ON sales.* TO 'sourcelace_reader';

-- For each person:
CREATE USER 'ann_lee'@'%' IDENTIFIED BY 'a long password' REQUIRE SSL;
GRANT 'sourcelace_reader' TO 'ann_lee'@'%';
SET DEFAULT ROLE 'sourcelace_reader' TO 'ann_lee'@'%';
```

SQL Server

```
CREATE LOGIN ann_lee WITH PASSWORD = 'a long password';
USE Sales;
CREATE USER ann_lee FOR LOGIN ann_lee;
ALTER ROLE db_datareader ADD MEMBER ann_lee;
```

On Azure SQL you can create the user in the database directly (CREATE USER ann_lee WITH PASSWORD = '...') and add it to db_datareader. SourceLace signs in with SQL logins (username and password); Microsoft Entra sign-in to the database is not supported yet.

Oracle Database

Oracle signs people in as database users, and a user's name is also its schema. Create one user per person and give them SELECT through a role:

```
-- Once: a role that may read the tables people should see.
CREATE ROLE sourcelace_reader;
GRANT SELECT ON sales.orders TO sourcelace_reader;
GRANT SELECT ON sales.customers TO sourcelace_reader;
-- (or, on Oracle 23ai, every table of a schema at once: GRANT SELECT ANY TABLE ON SCHEMA sales
  TO sourcelace_reader;)

-- For each person:
CREATE USER ann_lee IDENTIFIED BY "a long password";
GRANT CREATE SESSION TO ann_lee;
GRANT sourcelace_reader TO ann_lee;
```

Existing named users work as they are: SourceLace reads with whatever they can already see. Grant SELECT (or READ), not EXECUTE: a function that changes data inside its own transaction (PRAGMA AUTONOMOUS_TRANSACTION) could get round Oracle's read-only transaction, so only the database's grants stop that.

SourceLace connects with the database's **service name**, such as SALESPDB or sales.corvanta.com (SELECT sys_context('USERENV', 'SERVICE_NAME') FROM dual shows it); the older SID form is not supported. No Oracle Client software is needed, and Oracle Database 12.1 or later works.

MongoDB

```
use sales
db.createUser({ user: "ann_lee", pwd: passwordPrompt(), roles: [{ role: "read", db: "sales" }] })
```

On MongoDB Atlas: **Database Access → Add New Database User**, password authentication, with the built-in role **read** on the database (under *Specific Privileges*).

Let SourceLace in

SourceLace connects from its own servers to your database, so the database must:

1. **Have an address reachable from the internet**, such as `db.corvanta.com` or an RDS endpoint. SourceLace's cloud refuses private addresses (10.x, 172.16-31.x, 192.168.x, localhost), so that no organization can point it at another's internal systems.
2. **Allow SourceLace's outbound IP addresses** in its firewall or security group (AWS security group, Azure SQL firewall rule, Atlas **Network Access → IP Access List**), and nothing else from the internet. Ask support@sourcelace.com for the current list.
3. **Listen on the port** your SourceLace admin enters (usually 5432, 3306, 1433, 1521 or 27017; an Oracle TLS listener is often on 2484, and Oracle Autonomous Database on 1522).

TLS (encryption)

Every connection is encrypted, and SourceLace checks the database's certificate and that it was issued for the host name your admin entered. Make sure:

- TLS is on in the database (`ssl = on` in PostgreSQL, `require_secure_transport = ON` in MySQL, *Force Encryption* in SQL Server, a TCPS listener endpoint in Oracle's `listener.ora`, `net.tls.mode: requireTLS` in MongoDB). Managed services have it on already. For Oracle Autonomous Database, allow TLS connections without a wallet (mutual TLS not required), with SourceLace's IP addresses on its access control list.
- The certificate is valid for the host name people connect to. If you use a different name (a CNAME), add it to the certificate.
- If the certificate was signed by a private certificate authority (Amazon RDS, some Azure servers, or your company's own CA), give your SourceLace admin that CA's certificate (a PEM file starting `-----BEGIN CERTIFICATE-----`). For Amazon RDS this is the regional bundle from AWS's *Using SSL/TLS to encrypt a connection* page.

2. For your SourceLace admin

Add the source on **Manage sources → Add a source**.

SQL database options

Kind: **SQL database** (database).

Option	Type	Default	Example	What it is
engine	One of postgres, mysql, sqlserver, oracle	(required)	postgres	The kind of database. MariaDB uses mysql; Azure SQL uses sqlserver; Oracle Database uses oracle.
host	Host name	(required)	db.corvanta.com	The server's name only: no <code>https://</code> , no port.
port	Number	5432, 3306, 1433 or 1521, by engine	5433	The port. For Oracle, the port of the database's TLS listener (often 2484).

Option	Type	Default	Example	What it is
database	Text	(required)	sales	The database to read. For Oracle, the service name , such as SALESPDB (not the SID).
ca_cert	PEM certificate	(none)	-----BEGIN CERTIFICATE----- ...	The certificate of a private CA (see TLS above). It may be pasted on one line.
tls	verify or off	verify	verify	off is accepted only for a database on the same computer as SourceLace, so in practice always verify.

MongoDB options

Kind: **MongoDB** (mongodb).

Option	Type	Default	Example	What it is
host	Host name	(required)	mongo.corvanta.com or cluster0.abcde.mongodb.net	The server's name, or the Atlas cluster address.
port	Number	27017	27018	The port (not used with srv).
database	Text	(required)	sales	The database to read.
auth_source	Text	the database option	admin	The database where users are defined.
srv	true or false	false	true	true for a mongodb+srv:// address, as MongoDB Atlas uses.
ca_cert, tls				As for SQL databases.

Then each person clicks **Connect** and signs in on SourceLace's form with their own username and password.

3. What people can do, and how it stays read-only

- **SQL databases** take sql: one SELECT (or WITH ... SELECT) in the engine's own dialect, on tables named schema.table as search_schema lists them (SELECT TOP 10 ... on SQL Server, FETCH FIRST 10 ROWS ONLY on Oracle, LIMIT 10 elsewhere). On Oracle, a table name without a schema is looked up in the person's own schema, and names written without quotes are matched in upper case, as Oracle stores them. get_record works on tables with a single-column primary key.
- **MongoDB** takes mongodb: one JSON object, either a find, such as {"collection": "orders", "filter": {"status": "open"}, "sort": {"placed": -1}, "limit": 50}, or an aggregation pipeline, such as {"collection": "orders", "aggregate": [{"match": {"status": "open"}}, {"group": {"_id": "\$region", "orders": {"sum": 1}}}]}. Ids and dates use MongoDB

Extended JSON, such as {"\$oid": "..."} and {"\$date": "2026-10-01T00:00:00Z"}.

Checked before anything is sent:

- SQL: a single statement starting with SELECT or WITH; nothing that writes (INSERT, UPDATE, DELETE, MERGE, CREATE, DROP, ALTER, TRUNCATE, GRANT, SELECT ... INTO, COPY, EXEC, CALL and so on), nothing that locks rows (FOR UPDATE, FOR SHARE), no system procedures (xp_..., sp_...), and no functions that reach outside the query or change state (such as pg_sleep, pg_read_file, dblink, nextval, LOAD_FILE, OPENROWSET, WAITFOR). On Oracle, also no PL/SQL packages (DBMS_..., UTL_... such as UTL_HTTP, OWA_..., HTP), no URI types (such as HTTPURITYPE), nothing in the SYS schema, no database links (orders@other_db) and no functions defined inside the query (WITH FUNCTION).
- MongoDB: only read stages (\$match, \$project, \$group, \$sort, \$limit, \$skip, \$unwind, \$lookup, \$facet and similar). \$out and \$merge (which write) are refused, and so are \$where, \$function and \$accumulator (which run code) anywhere in the query. system. collections are off limits.

Enforced on the database too:

Database	Read-only	Time limit
PostgreSQL	A READ ONLY transaction, always rolled back	statement_timeout
MySQL and MariaDB	START TRANSACTION READ ONLY, rolled back	max_execution_time (MariaDB: max_statement_time)
SQL Server	A transaction that is always rolled back, with ApplicationIntent=ReadOnly. SQL Server has no read-only transaction, so the login's db_datareader role matters most.	Query timeout
Oracle Database	SET TRANSACTION READ ONLY first, then the query, then always a rollback	call_timeout on the connection
MongoDB	MongoDB has no read-only session, so the read role and the checks above do this work. Reads prefer a secondary.	maxTimeMS

Each query may run for 30 seconds and returns at most 2,000 rows to SourceLace, 50 shown at a time. Results stay in memory for 30 minutes and are never written to a database. Each request opens its own connection and closes it straight after. Schema comes from the database's own catalog (for Oracle, ALL_TABLES, ALL_VIEWS, ALL_TAB_COLUMNS and ALL_CONSTRAINTS, leaving out the schemas Oracle itself maintains), which lists only what the person can read; for MongoDB, describe_object samples 20 documents and reports field names and types only.

When something goes wrong

What you see	What to do
"Set the option 'host': the database server's name, such as db.corvanta.com." (or 'database', 'engine')	Fill in that option.
"...' is not a host name. Give only the name, such as db.corvanta.com."	Remove https://, the port, and any path.
"TLS can only be turned off for a database on the same computer as SourceLace (localhost)..."	Remove tls: off.
"The option 'ca_cert' must be a PEM certificate (-----BEGIN CERTIFICATE-----...)."	Paste the whole CA certificate, including the BEGIN and END lines.

What you see	What to do
"... is a private network address, and this SourceLace server only connects to databases on the internet..."	Use the database's public address and allow SourceLace's outbound IP addresses.
"... could not be reached. Check the host and port, and that the database allows connections from SourceLace's IP addresses."	Check host, port and the firewall allowlist. If it ends "(the database does not know that service name)", fix the Oracle database option: it must be the service name, not the SID.
"... could not be reached securely: its TLS certificate could not be checked. An admin can set the source's ca_cert option..."	Set ca_cert, or fix the certificate's host name.
"The database refused that username and password too many times. Ask for a new sign-in link to try again."	Connect again for a new link, and check the username and password with your DBA.
"This sign-in link has expired or was already used. Ask for a new one."	Click Connect again.
"... no longer accepts your database username or password." (or MongoDB)	The password changed or the login was removed. Connect again.
"... did not finish the query in time. Narrow it down and try again."	Add filters or a smaller limit.
"... refused the query: ..."	The database's own reason follows, such as a missing grant or an unknown column.
"There is no table or view ... that you can read. Call search_schema to find it."	Check the name (schema.table); the login may lack a grant.
"... has no single-column primary key. Use query with a WHERE clause instead."	Use query instead of get_record.
"'...' is not a collection you can read. Call search_schema to list them."	Check the collection name and the login's read role.