

# All of Oracle development, in one window. Instant, even on the largest schemas.

SQLlightspeed brings the whole Oracle development lifecycle into one application: writing code, compiling, debugging, performance analysis, regression testing, object comparison and database administration. It is a native Windows and macOS application that does not slow down on very large enterprise application databases.

## One application

Editor · debugger · profiler · tests · DBA

## Very large schemas

First index ready in under a second

## Windows and macOS

A native application on both

The screenshot shows the SQLlightspeed application interface. At the top, it says 'SQLlightspeed' and 'ERPPROD (demo) · worksheet and result grid'. On the right, it shows 'objects indexed: 200,000', 'sync 0.1 s', and 'session 1 of 3'. The main area is divided into two panes. The left pane, titled 'OBJECTS · CONNECTIONS', shows a list of database objects with their sizes: ERPAPP (142,880), ERPINFO (18,204), MEGA\_APP (9,061), FIN\_APP (8,340), ARONE\_APP (6,905), TRESOL\_APP (4,812), ACC\_RUL (3,690), and INV\_LIB (3,004). The right pane, titled 'Worksheet 1', shows SQL code for a query on the 'CUSTOMER\_ORDER\_API' table. The code is: 

```
1 -- own Oracle session per worksheet
2 select c.order_no, c.customer_no, c.state, c.wanted_delivery_date
3 from erpapp.customer_order_tab c
4 where c.state in (Released, Reserved)
5 and c.wanted_delivery_date < sysdate + 7
```

 Below the code, it says 'RESULT GRID · 1,204 ROWS FETCHED' and shows a table with columns: ORDER NO, CUSTOMER, STATE, WANTED DELIVERY, and LINES. The table has three rows: CO-104871 (NORTHLINE, Released, 2026-09-04, 14), CO-104872 (HARBOUR-LG, Released, 2026-09-04, 6), and CO-104880 (AEGEAN-MT, Reserved, 2026-09-05, 31). At the bottom, it shows a log of connections and sync status.

## Most of your business logic lives inside the database. Your development environment still stands outside.

A package change is written in one tool, compiled in another, debugged in a third and its plan read in a fourth. What the change broke in Test or Production is usually **discovered on deployment night.**

SQLlightspeed puts that whole loop in one **window** — write, compile, see the error on its line, step through it, measure its profile, run the regression set and see the Dev-Test-Prod difference before the release, not during it.

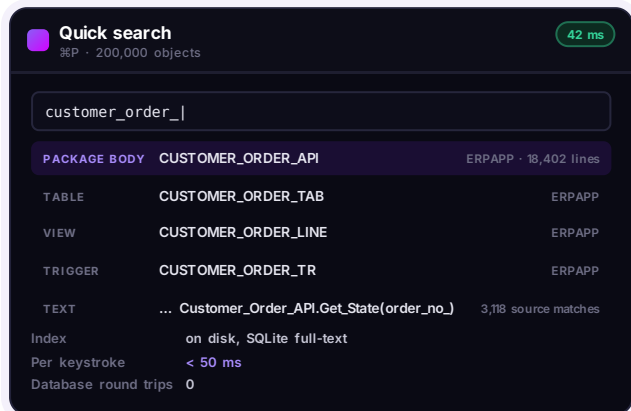
Nothing is installed on the server side; no Instant Client, no Java.

On the Mac, no virtual machine and no remote desktop.

Not a single object in your schema changes.

# What changes on day one.

Not a SQL editor, and not a web console. The whole Oracle working day happens in one application: from the object tree to the DBA panels, from the compile loop to the explain plan. **Every worksheet runs in its own Oracle session** — a long-running query never blocks another tab.

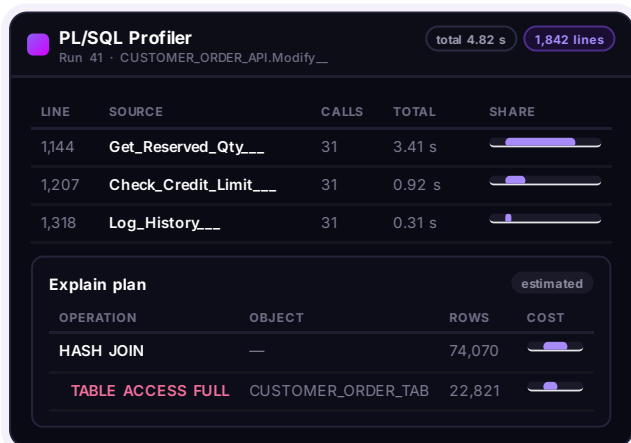
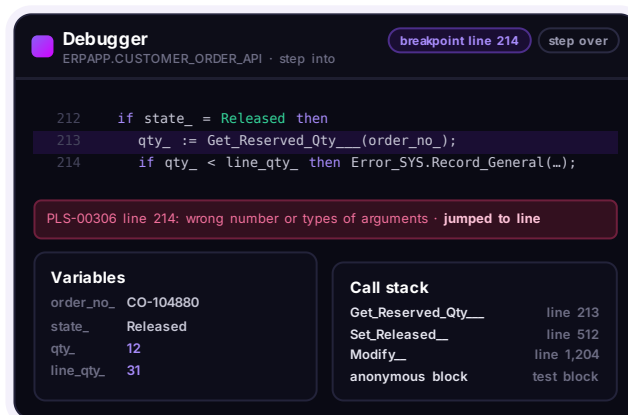


## Writing and finding

Code Assistant suggests columns, parameters and package elements as you type; code folding, hyperlink navigation and the PL/SQL Beautifier keep large packages readable. ⌘P search across hundreds of thousands of objects answers in **under 50 milliseconds** per keystroke — the metadata sits in a local index on disk, not in a memory dump.

## Compile, stop, fix

You compile from the editor and the error lands on its exact line. The integrated debugger steps through procedures, functions, packages and triggers; variable values and the call stack are on the same screen. A test block with prefilled parameters is generated for any procedure.



## Do not optimise by guessing

The PL/SQL Profiler shows how long each line took; SQL statistics and explain-plan cost bars mark the risky step. The reason slow code is slow comes from a **measurement**, not a hunch.

### MEASURED IN THE REFERENCE INSTALLATION

**200,000**

Objects; first index ready in 0.1 seconds

**< 50ms**

Per keystroke in ⌘P quick search

**25MB**

On-disk size of the object index

**0**

Install components: Instant Client, Java, server objects

# Mistakes surface before production does.

Cost does not come from large projects. It comes from small, quiet changes: one extra parameter on a package, an index that exists in Test but not in Prod, a trigger whose firing order moved. **SQLlightspeed stops at exactly those three points.**

**Object compare**  
ERPDEV → ERPTST · 1,204 objects scanned

7 differences DDL ready

ERPDEV source → ERPTST target · ERPPROD next

| TYPE    | OBJECT              | DIFFERENCE        | ACTION  |
|---------|---------------------|-------------------|---------|
| PACKAGE | CUSTOMER_ORDER_API  | source differs ×3 | deploy  |
| INDEX   | CUST_ORDER_STATE_IX | missing in target | missing |
| TRIGGER | CUSTOMER_ORDER_TR   | fire order moved  | deploy  |
| GRANT   | ORDER_MANAGER       | not granted       | missing |

2 differences would fail on deployment — found before the release

## The Dev – Test – Prod difference

Object compare checks packages, views, triggers, indexes and grants across two environments; the differences appear in one list before the release, together with the DDL needed to align them. The surprise arrives in the morning, not at midnight.

## Regression testing

Test Manager re-runs saved test sets after a change; every scenario that deviates from its expected output is listed by name and opens straight in the debugger.

“What does this change break?” becomes a report instead of a guess.

**Test Manager**  
Set: CUSTOMER\_ORDER · after change 4188

24 passed 1 failed

|   |                                                               |        |
|---|---------------------------------------------------------------|--------|
| ✓ | Release order with full reservation · expected state Released | 0.21 s |
| ✓ | Release order over credit limit · expected Error_SYS raised   | 0.34 s |
| ✗ | Partial reservation, 31 lines · expected 31, returned 12      | 0.28 s |
| ✓ | Cancel reserved order · expected state Cancelled              | 0.19 s |

1 of 25 scenarios broke with this change — Get\_Reserved\_Qty\_\_ line 1,144

Re-run failed Open in debugger Export report

**DBA panels**  
Sessions · wait events · blocking tree

no auto-refresh PROD

Sessions Locks Tablespaces Jobs Invalid objects AQ

| SID   | USER    | STATUS  | WAIT EVENT          | SEC | BLOCKER |
|-------|---------|---------|---------------------|-----|---------|
| 7,045 | APPSYS  | active  | db file seq read    | 3   | —       |
| 4,201 | ERPAPP  | blocked | enq: TX row lock    | 128 | 7,731   |
| 7,731 | RPT_USR | active  | SQL*Net from client | 131 | —       |

## Seeing the live system

The DBA panels gather sessions, the blocking-lock tree, tablespace growth, scheduler jobs, invalid objects, AQ queues, parameters and the recycle bin in one tab. Invalid objects are recompiled in bulk in a single action.

### WHERE THE SPEED COMES FROM — THREE ARCHITECTURAL DECISIONS

**Not a feature list but design choices.** Speed was not added later; it was put in the foundation.

#### LOCAL INDEX

##### Metadata on disk

SQLite with full-text search. Hundreds of thousands of objects ≈ 25 MB; queries are measured in milliseconds.

#### RUST CORE

##### A native shell

Tauri 2 over the operating system's own web view. No bundled browser, a small binary.

#### THIN DRIVER

##### Direct connection

Oracle Instant Client and Java are not required. The app carries its own runtime.

There is one standard: **if a feature stutters, that feature is not done.** Performance claims are verified with a stopwatch and an automated suite, and the interface never blocks — not even while indexing runs.

# For every team that works *inside Oracle.*

## Value for every team

For teams whose day is spent *inside* Oracle.

- **Oracle and PL/SQL developers**  
The compile–error–fix loop, go-to-definition, generated test blocks and explain plans in one window.
- **Enterprise application and ERP teams**  
Instant full-text search inside package sources across schemas of hundreds of thousands of objects.
- **Database administrators**  
Sessions, locks, tablespaces and jobs at a glance; bulk recompilation of invalid objects.
- **Technical consultants**  
A single file installs into a customer environment; analysis starts without waiting for an Oracle client.
- **Support teams**  
Root-cause analysis of a production problem through the SQL window and explain plan.
- **Data and application teams**  
Editing in the result grid; CSV, TSV, XML, HTML and Excel output; comparison of two result sets.
- **Teams on Windows and Mac**  
One native application on both; on the Mac, no virtual machine and no remote desktop.

## Trust and control

Speed is *no excuse* for carelessness.

- **Production is marked**  
PROD connections carry a red stripe and destructive actions ask for an extra confirmation.
- **Passwords in the OS vault**  
Credentials live in the macOS Keychain or Windows Credential Manager, never in the application's own files.
- **Your schema is untouched**  
No repository, trigger or helper object is installed on the server side.
- **Signed distribution**  
A code-signed package; the macOS build is Apple-notarized.
- **No AI makes the decision**  
The product runs no generative model that decides code or access; the last call stays with the developer.
- **Your data stays put**  
The application runs locally; the index holds metadata only and no data is sent to a cloud service.
- **Flexible connections**  
Service name, SID, tnsnames.ora, Easy Connect Plus and proxy users are all supported.

# Free

TWO-WEEK PoC

We start on your largest schema: we measure how long the index takes to build and how quickly a search returns, then run the compile–debug loop and a Dev–Test comparison on your own packages. **No cost. No commitment.**

**Request a PoC**

or write to us directly  
contact@dbotech.uk