ASTRAVERGE / PGDEVELOPER

Руководство пользователя

AstraVerge PGDeveloper 0.7.0

Практическое руководство · 11 ноября 2026

Рабочая среда для PostgreSQL: от структуры базы до кода приложения.

Команды и имена файлов сохраняют техническое имя av-pgdeveloper. Параметры и границы применимости относятся к версии 0.7.0.

Содержание

Работа с продуктом 4

1. Откройте рабочее пространство . . . 4
2. Изучите структуру . . . 4
3. Напишите и проверьте SQL . . . 5
4. Подтвердите изменения осознанно . . . 7
5. Подготовьте код приложения . . . 7
6. Передайте результат . . . 8
7. Подключите дополнительные возможности . . . 10
8. Обратитесь за поддержкой . . . 10

Установка и первый запуск 10

Подготовка . . . 10
Порядок установки . . . 10
Проверка поставки разработчиком . . . 10

Настройка рабочего места 10

Соединения . . . 10
Работа с данными . . . 11
Интерфейс . . . 11
Дополнительные контуры . . . 11
Подтверждение настроек и изменений . . . 11

Эксплуатация и сопровождение 12

Ежедневная работа . . . 12
Изменения схемы . . . 12

Резервное копирование 12
Обновление 12
Приёмка 12
Безопасность 12
Секреты и транспорт 12
SQL и выгрузки 13
Поддержка 13
Интерфейсы интеграции и форматы 13
Диагностика проблем 14
Обзор продукта 14
Что получает команда 15
Для кого 15
Состав и границы 15
Отладчик PL/pgSQL 16
Назначение и запуск 16
Пошаговая работа 16
Серверные требования и границы 17
Интерфейс 17

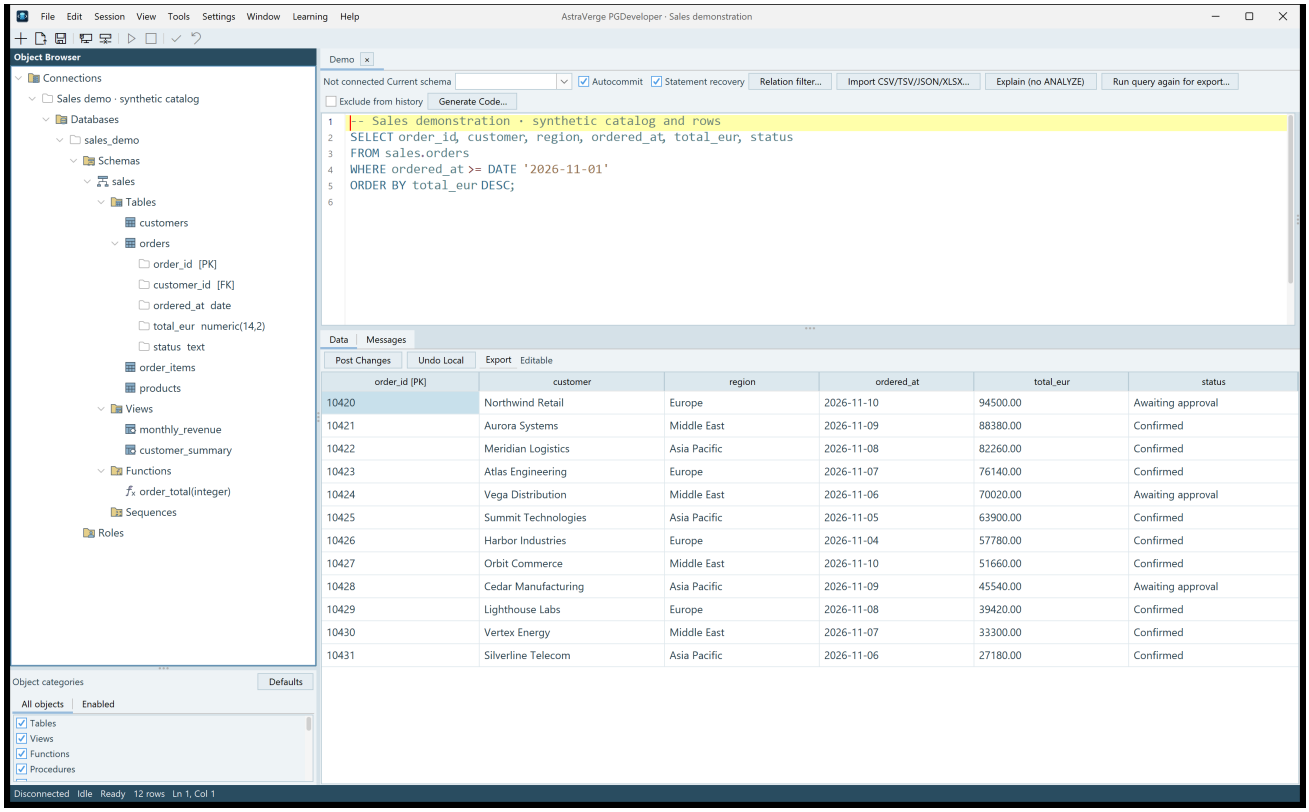
AstraVerge PGDeveloper 0.7.0. Единое руководство.

Работа с продуктом

AstraVerge PGDeveloper 0.7.0 - настольная среда для PostgreSQL. Руководство описывает путь от знакомства со структурой до проверяемого результата. Сначала выполните установку и настройку.

1. Откройте рабочее пространство

Запустите `AV-PGDeveloper.exe` из полного каталога поставки. Для знакомства выберите Demo - локальный учебный режим без подключения к БД. Для реальной работы создайте соединение с согласованной базой PostgreSQL 14-18 и требуемым SSL-режимом. Уточните права у администратора. Не используйте производственные данные для учебных изменений.

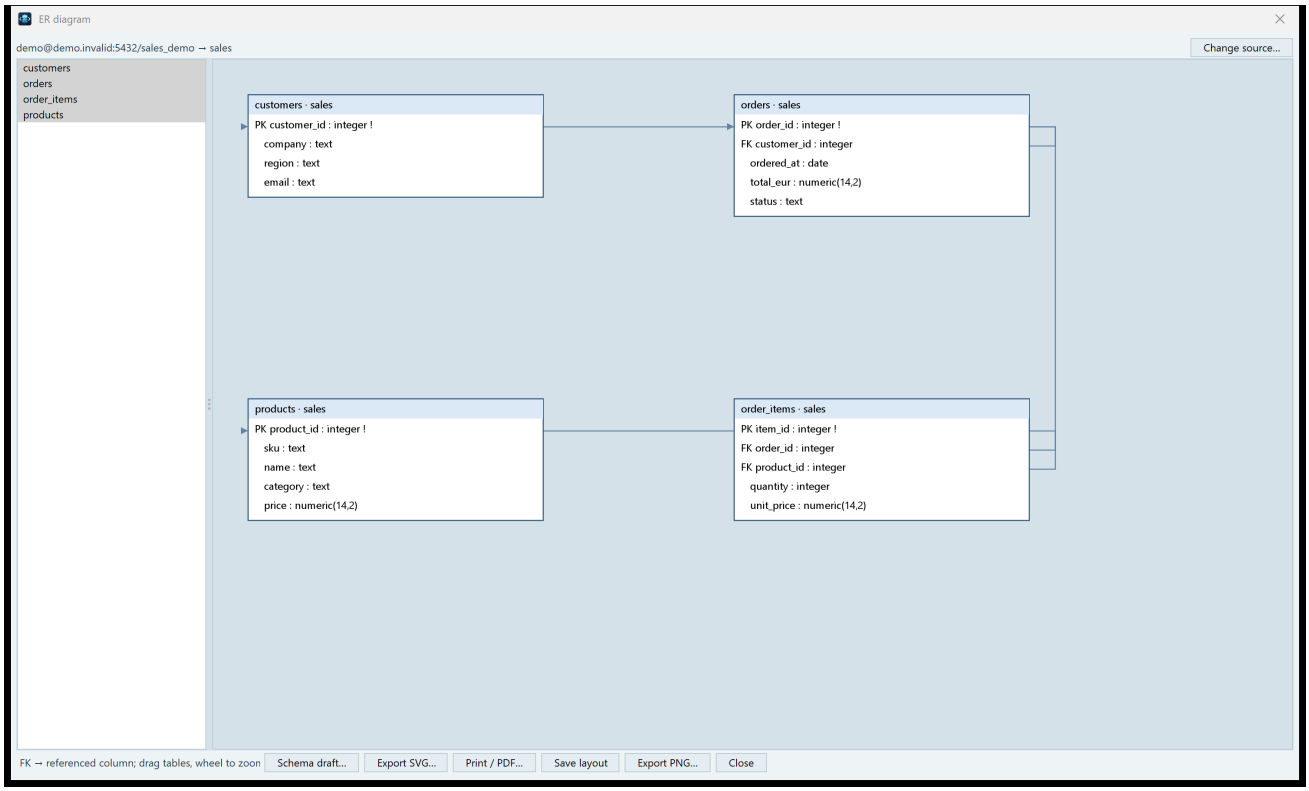


Рабочее пространство

Снимки сделаны с настоящих компонентов на синтетических данных без подключения к серверу при съёмке.

2. Изучите структуру

Выберите базу и схему в Object Browser. Просматривайте объекты, определения DDL и связи таблиц. Используйте ER-диаграмму, чтобы обсудить модель с коллегами. Диаграмма помогает понимать структуру; ограничения и права проверяет PostgreSQL.



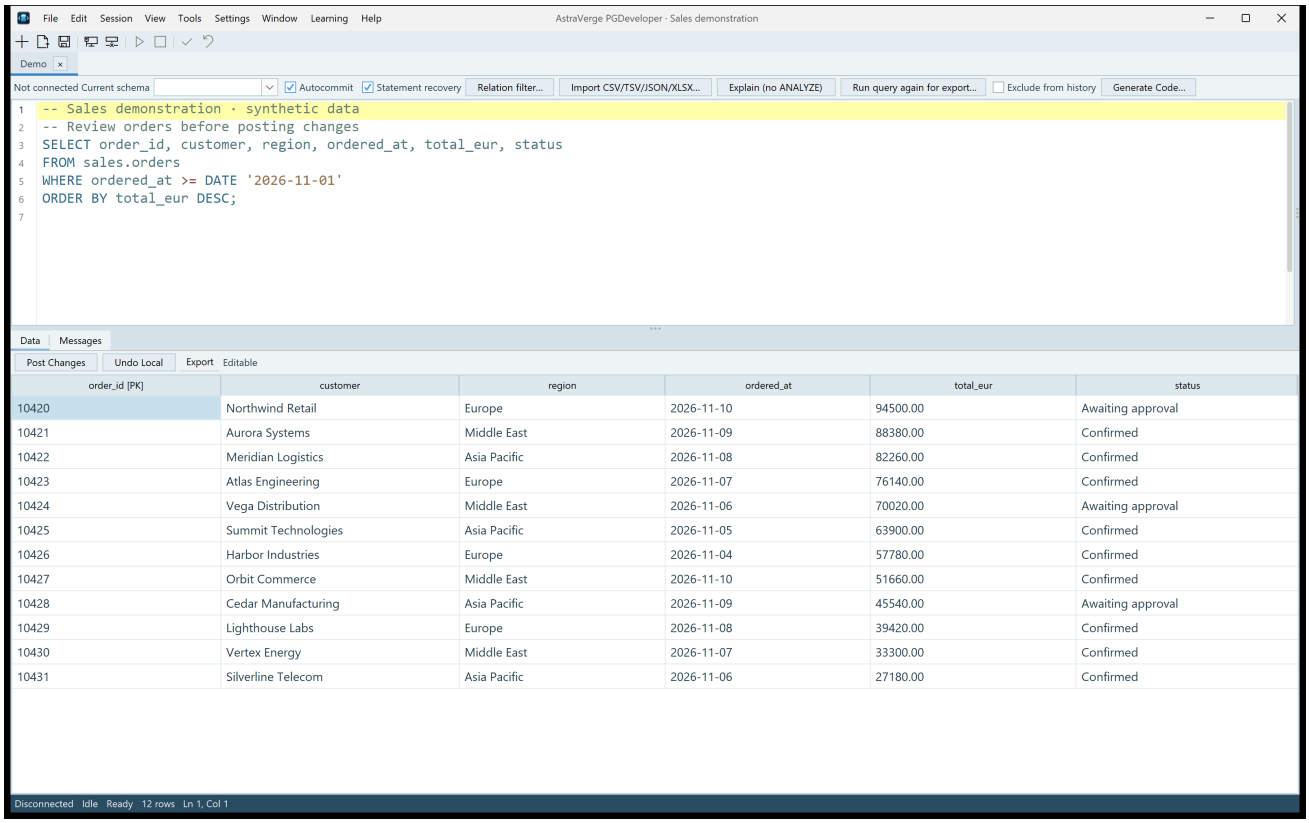
ER-диаграмма

3. Напишите и проверьте SQL

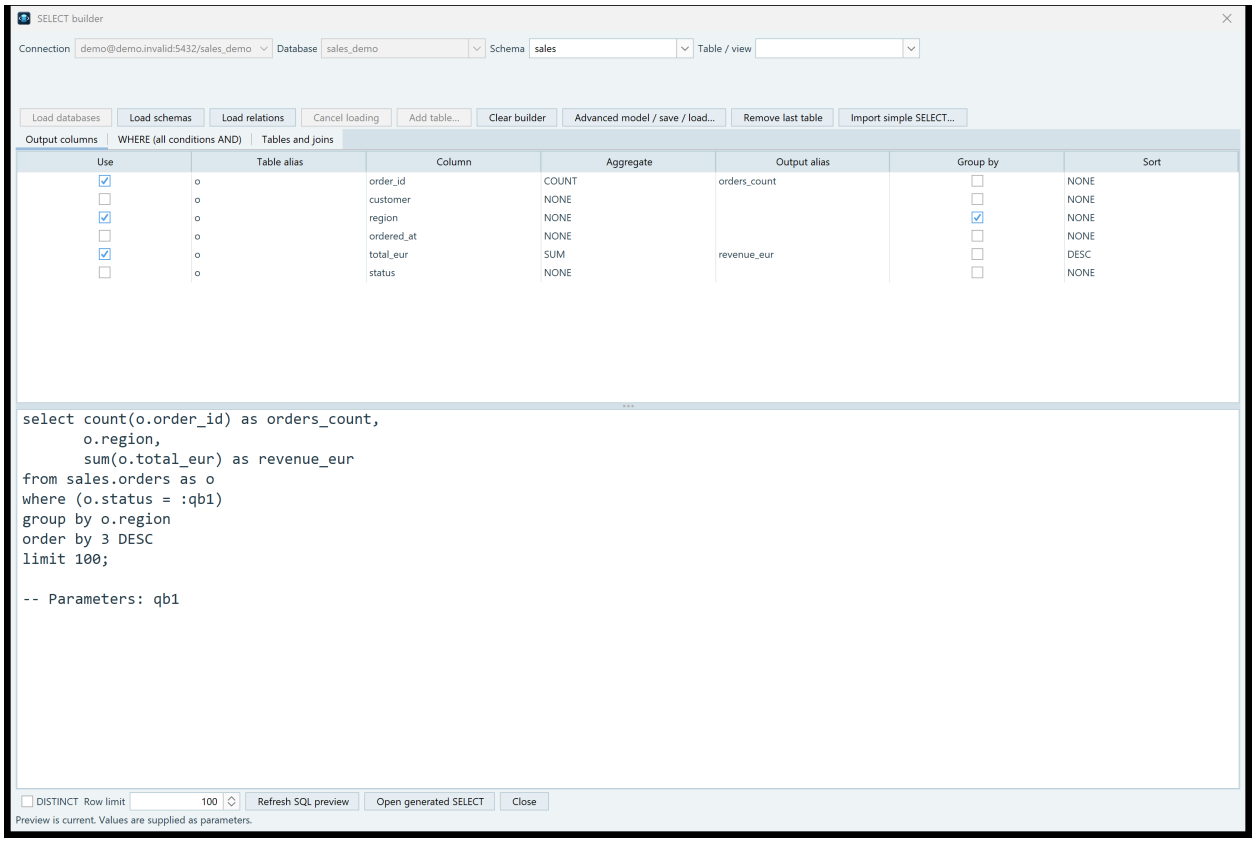
Откройте редактор, используйте дополнение, параметры и таблицу результатов. Для первого безопасного запроса на согласованном сервере выполните:

```
SELECT current_database(), current_user;
```

Убедитесь, что выбраны ожидаемая база и пользователь. Для построения выборки используйте визуальный SELECT Builder; проверьте созданный SQL до выполнения. Сопоставляйте результат с условиями запроса, а не только с отсутствием ошибки.



SQL и результат

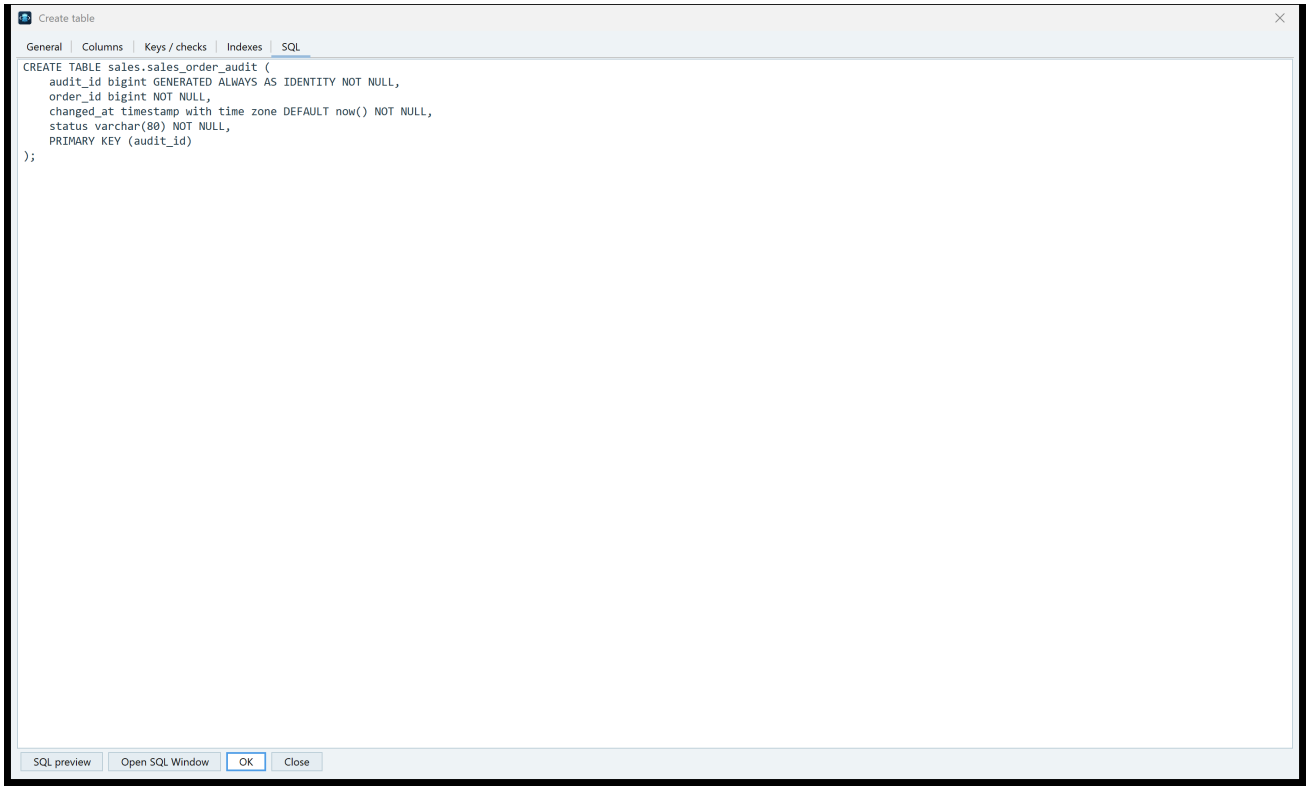


Конструктор SELECT

4. Подтвердите изменения осознанно

Локальное редактирование строк, Post и Commit - разные этапы. Проверьте изменённые значения, отправку и окончательное подтверждение транзакции. Для отказа от изменений используйте предусмотренный Rollback с учётом текущего состояния. При ошибке или разрыве соединения не считайте начатую операцию завершённой и не повторяйте запись без выяснения результата.

В диалогах создания и изменения объектов и таблиц основная кнопка - OK. Она завершает операцию только после подтверждённого успеха. Ошибка оставляет окно открытым для исправления и повторной попытки. Подготовленный SQL можно предварительно открыть и проверить отдельно.

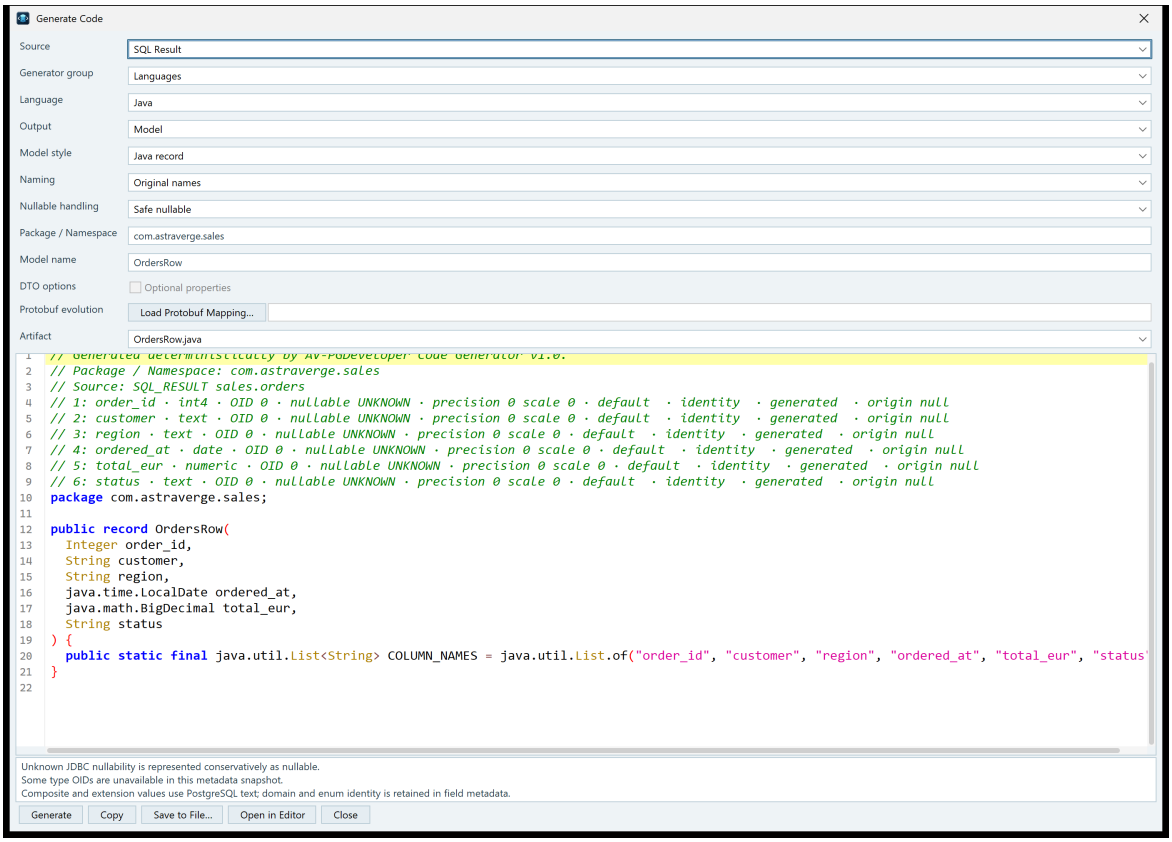


Конструктор таблицы и OK

Настоящий Table Designer на синтетическом определении; OK при съёмке не нажималась, SQL не исполнялся.

5. Подготовьте код приложения

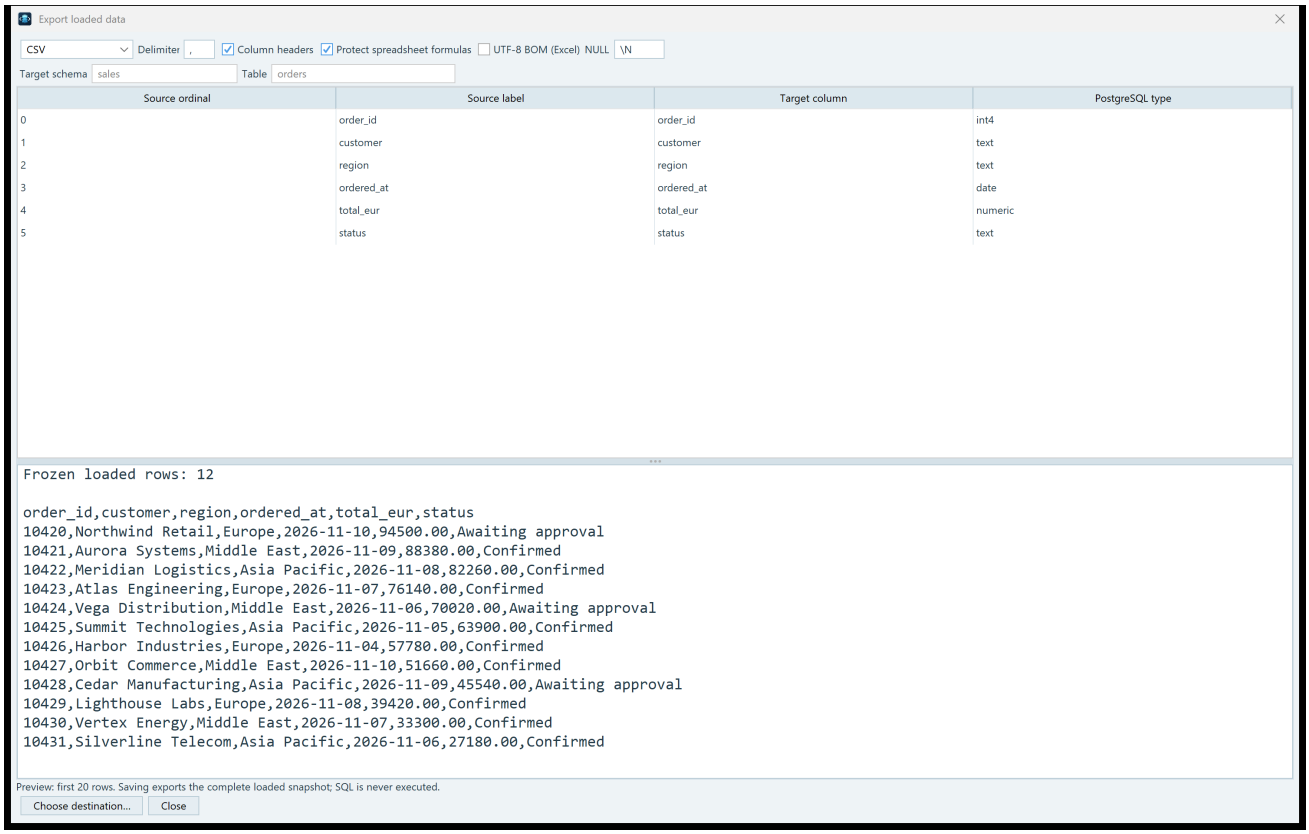
Откройте Generate Code для выбранных метаданных. Выберите поддерживаемый язык, модель, отображение строк или контракт. Проверьте предпросмотр, имена, типы, nullable-поля и сохранённый исходный текст перед включением в проект. Генерация не заменяет ревью и испытания приложения. Подробности - в руководстве генерации.



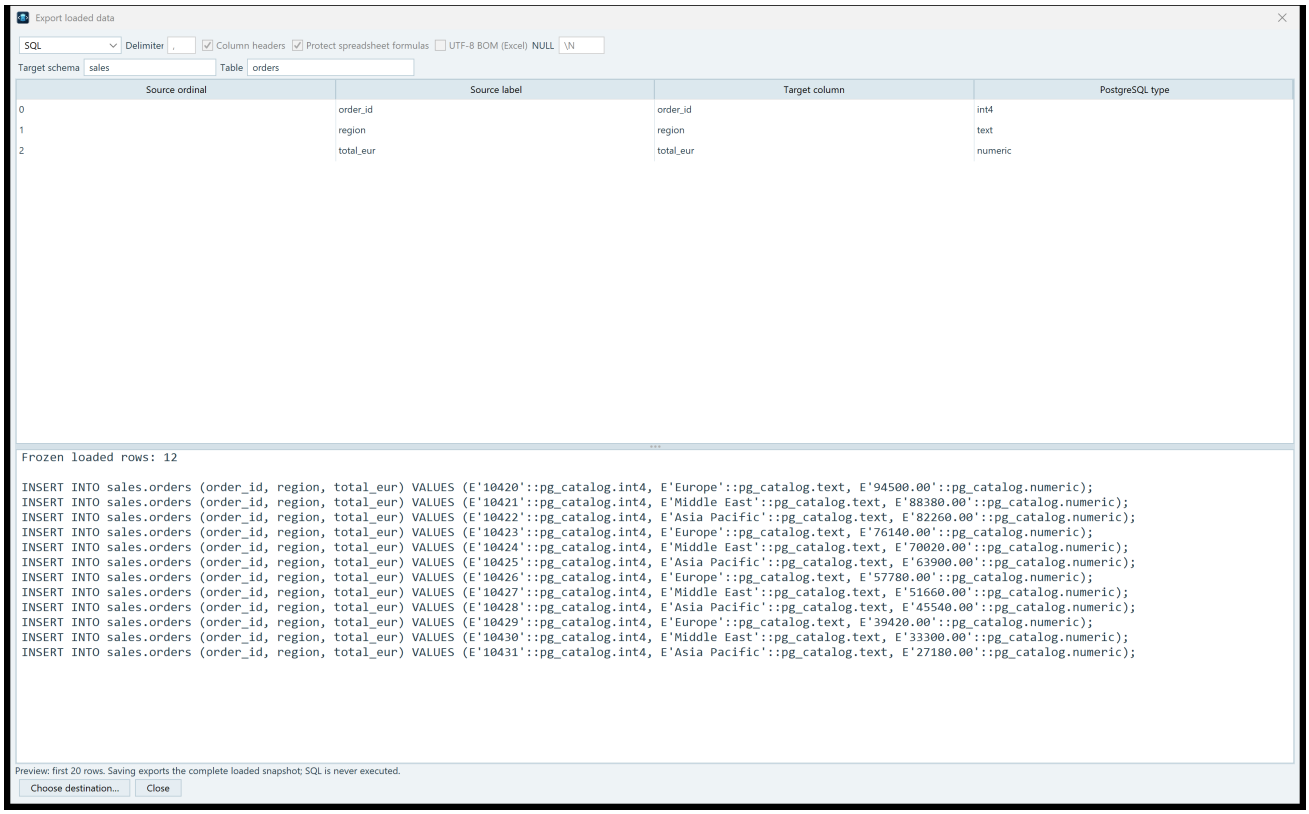
Генерация кода

6. Передайте результат

Настройте формат экспорта, выбранные столбцы и их сопоставление. Проверьте предпросмотр и целевой путь. Дождитесь завершения сохранения; открытие окна предпросмотра не подтверждает запись файла. Для SQL-экспорта проверьте PostgreSQL-типы и сформированные INSERT до исполнения в следующей системе.



Предпросмотр экспорта



SQL-экспорт

7. Подключите дополнительные возможности

Аналитика требует отдельного R runtime и настройки аналитического контура. Backup/restore использует подходящие утилиты PostgreSQL. Перед применением сверяйте зависимости и права с соответствующим инженерным приложением. PostgreSQL 19 имеет предварительный режим, который не входит в стабильную поддержку 14-18.

8. Обратитесь за поддержкой

При ошибке сохраните версию, время, выбранный сценарий, обезличенное сообщение и состояние операции. Не передавайте пароли, строки подключения с секретами или пользовательские данные. Продолжение: диагностика, безопасность, эксплуатация.

Установка и первый запуск

Подготовка

Согласуйте источник дистрибутива, поддерживаемую Windows-среду и доступ к PostgreSQL 14-18. Поставляемый EXE использует встроенный Java runtime: переносите весь каталог приложения, а не один исполняемый файл. Для аналитики R и резервного копирования подготовьте отдельные зависимости соответствующих функций.

Порядок установки

1. Распакуйте согласованный дистрибутив в доступный пользователю каталог. 2. Запустите `AV-PGDeveloper.exe`. Официальное название продукта - AstraVerge PGDeveloper. 3. Для знакомства откройте Demo: это локальный учебный режим без подключения к БД. 4. Создайте соединение, задав сервер, порт, базу, пользователя и требуемый SSL-режим. Пароль передавайте через предусмотренный интерфейс. 5. После успешного подключения выберите базу и схему. Выполните `SELECT current_database(), current_user;` на согласованном сервере. 6. Проверьте просмотр таблицы и экспорт небольшой выборки в отдельный файл.

Не используйте production для первой проверки редактирования. Для начала выдайте отдельную роль с минимальными правами. Ошибка подключения должна исправляться в сохранённой форме: нажатие OK само по себе не подтверждает успех.

Проверка поставки разработчиком

Из корня исходников: `./gradlew.bat windowsExe`. Единственный каталог EXE-сборки - `build/native/AV-PGDeveloper/`. Это команда сборки, не инструкция покупателю скачивать SDK.

Политика runtime и точная версия Java приведены в описании runtime.

Настройка рабочего места

Настройка начинается с профиля соединения. Для каждого окружения задайте понятное имя и проверьте базу, пользователя, режим SSL и таймауты. Пометка production помогает различать окружения, но не заменяет серверные права.

Соединения

Используйте вкладки подключения, SSH tunnel и Client TLS по правилам вашей инфраструктуры. Корневой сертификат проверяет сервер; клиентский сертификат решает другую задачу - аутентификацию клиента. Не отключайте проверку сертификатов как штатный способ устранения ошибки.

Работа с данными

В Settings настройте размер порции Fetch, пороги автоматической загрузки и бюджет памяти результатов. Для больших наборов используйте порционную загрузку или потоковый экспорт. Не увеличивайте лимит памяти без оценки доступной памяти рабочего места.

Интерфейс

Доступны русский и английский языки, темы, независимые настройки шрифтов и цветов. Для цветовых значений применяется HEX-формат. На небольшом экране сохраняйте рабочий размер шрифта и используйте предусмотренную прокрутку, когда все поля физически не помещаются.

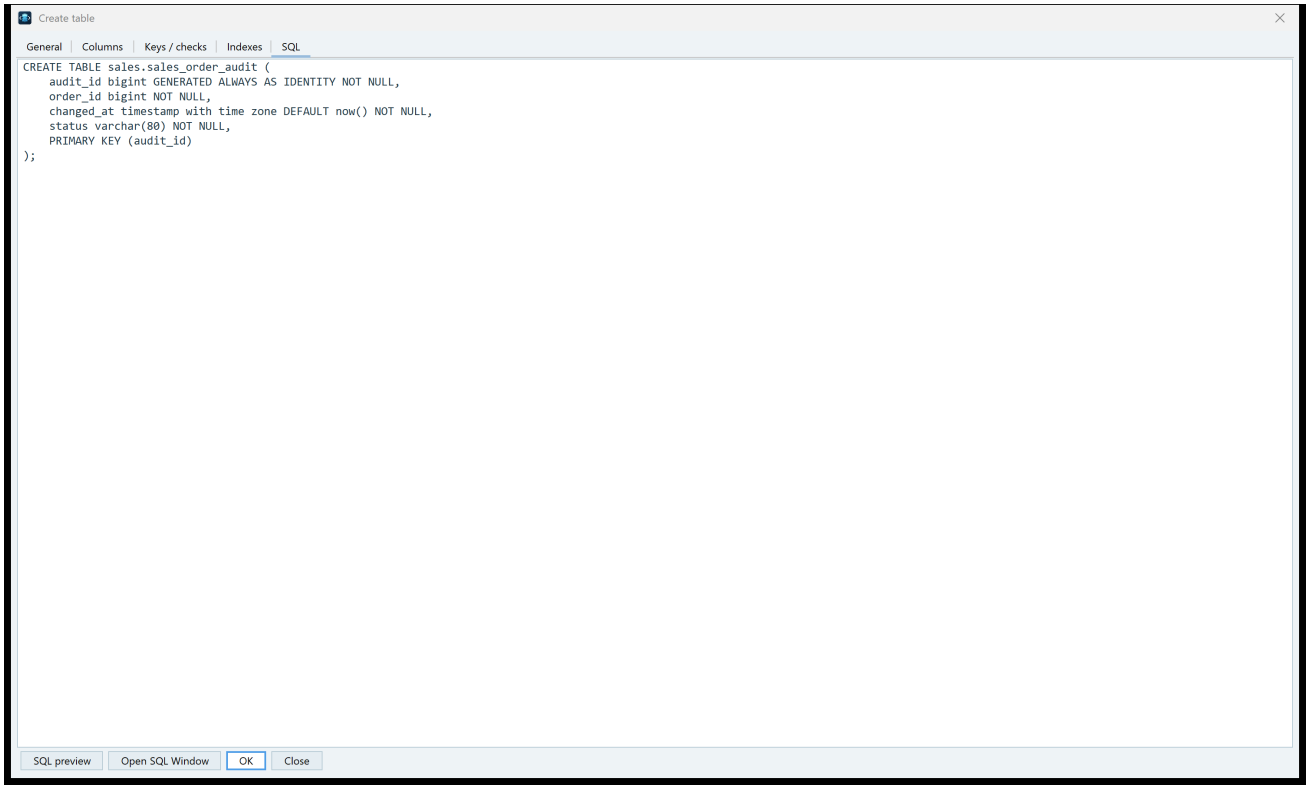
Дополнительные контуры

R runtime устанавливается отдельно. Аналитический доступ и окружение настраиваются согласно руководству аналитики. Kafka, RabbitMQ и Valkey/Redis требуют собственных серверов и учётных данных. Наличие команды в меню не означает, что внешняя служба уже развёрнута.

Подробности: аналитика, messaging, выбор источника.

Подтверждение настроек и изменений

В AI/Ollama кнопка OK проверяет и сохраняет настройки, затем закрывает окно. При ошибке ввод сохраняется для исправления и повторной попытки. В диалогах создания и изменения объектов и таблиц OK завершает операцию только после подтверждённого успеха. В палитре и настройках шрифтов кнопка «Применить» позволяет оценивать несколько вариантов без закрытия окна.



SQL создания таблицы и OK

Реальный Table Designer на синтетическом определении; SQL при съёмке не выполнялся.

Эксплуатация и сопровождение

Ежедневная работа

Перед выполнением SQL проверьте активную вкладку, соединение и текущую схему. При правке данных различайте локальные изменения в гриде, Post и Commit. Rollback относится к транзакции SQL-сессии; он не отменяет уже подтверждённые изменения другой сессии.

Изменения схемы

Перед применением DDL просмотрите определение и зависимости объекта. Для сравнения схем согласуйте источник и назначение; проверьте сформированный план. Применение миграции требует серверных прав и отдельного решения оператора.

Резервное копирование

Подготовьте совместимые pg_dump/pg_restore и доступ к целевому серверу. Проверяйте формат архива, выбранные объекты, владельцев и роли. Для физического backup/PITR действуют отдельные требования. Успешное создание файла ещё не доказывает восстановимость: выполните тестовое восстановление в отдельную БД.

Обновление

Сохраните пользовательские SQL-файлы, согласованный резервный набор локальных настроек и предыдущую поставку. Завершите фоновые работы и транзакции перед заменой файлов приложения. После обновления проверьте подключение, чтение данных, экспорт и критичные для команды функции. Не накладывайте новую поставку на работающий процесс.

Приёмка

Фиксируйте версии клиента и сервера, роль, сценарий, ожидаемый результат и фактический статус. При сбое сохраняйте диагностическое сообщение без секретов. Для воспроизведения используйте обезличенные данные.

Подробные процедуры: backup/restore, PITR, рабочие сценарии.

Безопасность

AstraVerge PGDeveloper выполняет действия с правами подключённой роли PostgreSQL. Клиент не должен использоваться для обхода серверной политики доступа. Разделяйте повседневную роль чтения и административные роли.

Секреты и транспорт

Пароли вводите в штатных формах и храните через предусмотренные механизмы продукта. На Windows используется защита секретов текущего пользователя; на других сценариях применяются предусмотренные хранилища. Не копируйте файлы секретов как универсальный способ переноса подключения между пользователями.

Для удалённого доступа настройте TLS с проверкой сервера либо согласованный SSH-туннель. SSH host keys должны быть заранее доверены в known_hosts. Клиентский TLS требует корректного сертификата, ключа и пароля контейнера, если он используется.

SQL и выгрузки

Параметризуйте значения запросов. Перед исполнением DDL и сгенерированного SQL проверяйте объекты назначения. Предпросмотр не является применением операции. Выгруженные CSV, XLSX, JSON и SQL-файлы могут содержать чувствительные данные независимо от защищённости соединения. Храните и передавайте их по политике организации.

Поддержка

Перед передачей логов или скриншотов удалите пароли, токены, персональные данные и внутренние адреса. Коммерческие снимки этого комплекта используют только синтетические данные. Защита клиента не заменяет серверный аудит, резервное копирование, разграничение доступа и контроль рабочего места.

Смежные материалы: контекст объектов, backup, messaging TLS/SASL.

Интерфейсы интеграции и форматы

Это обзор доступных способов работы с продуктом, а не обещание публичного REST API настольного клиента.

Интерфейс	Назначение	Условия
SQL/JDBC к PostgreSQL	Запросы, метаданные и транзакции	Поддерживаемая версия сервера, сетевой доступ и права
Result Grid и экспорт	Передача загруженных или потоковых данных	Выбор формата, типов и целевых столбцов
Generate Code	Модели, отображения строк, поддерживаемый CRUD	Проверка исходного текста и зависимостей целевого языка
Контракты	JSON Schema, OpenAPI, Protobuf, Avro, GraphQL	Возможности отличаются по генератору
Аналитический контур	R-сценарии, консоль и отчёты	Отдельный runtime и настроенный доступ
Messaging	Работа с поддерживаемыми брокерами	Конфигурация брокера, транспорт, аутентификация

Генераторы охватывают Java, Go, Python, TypeScript, C#, PHP, C++, Rust, R и MATLAB. Не каждый генератор поддерживает каждый вариант Output. Экран показывает доступные варианты для выбранной цели. Generated code остаётся исходным текстом, который команда адаптирует и проверяет в своём проекте.

Для точных numeric, bigint и NULL поведение зависит от языка и метаданных. Не приводите все значения к double ради удобства передачи. Используйте правила соответствующего генератора.

Подробные контракты: Generate Code, messaging, R.

Диагностика проблем

Симптом	Что проверить	Следующее действие
Не удаётся подключиться	Адрес, порт, сеть, база, роль, SSL, сертификаты, SSH	Исправить параметры в открытой форме и повторить
Нет нужного объекта	Текущая база и схема, область объекта, права роли	Выбрать правильный источник и обновить каталог
Грид только для чтения	Режим просмотра, происхождение результата, идентификатор строки, права	Открыть поддерживаемый сценарий Edit Data, не обходить ограничение
Правки не видны другой сессии	Выполнены ли Post и Commit	Проверить статус транзакции; не повторять запись вслепую
Загружена только часть строк	Fetch, автоматические лимиты, бюджет памяти	Дозагрузить порцию либо использовать потоковый экспорт
Ошибка экспорта	Доступность каталога, формат, типы, сопоставление	Сохранить ввод, исправить причину, повторить; SQL-экспорт открывает новый источник
Неверный результат generated code	Исходные метаданные, типы, nullable, выбранный артефакт	Перегенерировать и проверить в проекте; не исполнять код без review
Не работает аналитика	R runtime, аналитический доступ, настройки контура	Выполнить диагностику R по профильному руководству
Не запускается backup	Версия и путь утилит, права и место на диске	Сверить требования pg_dump/pg_restore и повторить в тестовой среде
Поля не помещаются	Размер окна, масштаб, шрифт	Использовать доступную прокрутку; приложить обезличенный снимок к обращению

В обращении укажите версию клиента и PostgreSQL, последовательность действий, ожидаемый результат и полный текст ошибки без секретов. Если исход транзакции неизвестен, сначала проверьте состояние сервера. Не считайте повторное нажатие OK подтверждением предыдущей операции.

История проверки исправлений: code review.

Обзор продукта

AstraVerge PGDeveloper - настольная рабочая среда для разработки и сопровождения PostgreSQL. Она объединяет SQL-редактор, результаты запросов, работу со структурой базы, генерацию прикладного кода и инструменты анализа. Разработчик проходит путь от объекта базы до проверяемого результата в одной среде, сохраняя контроль над SQL и транзакцией.

Что получает команда

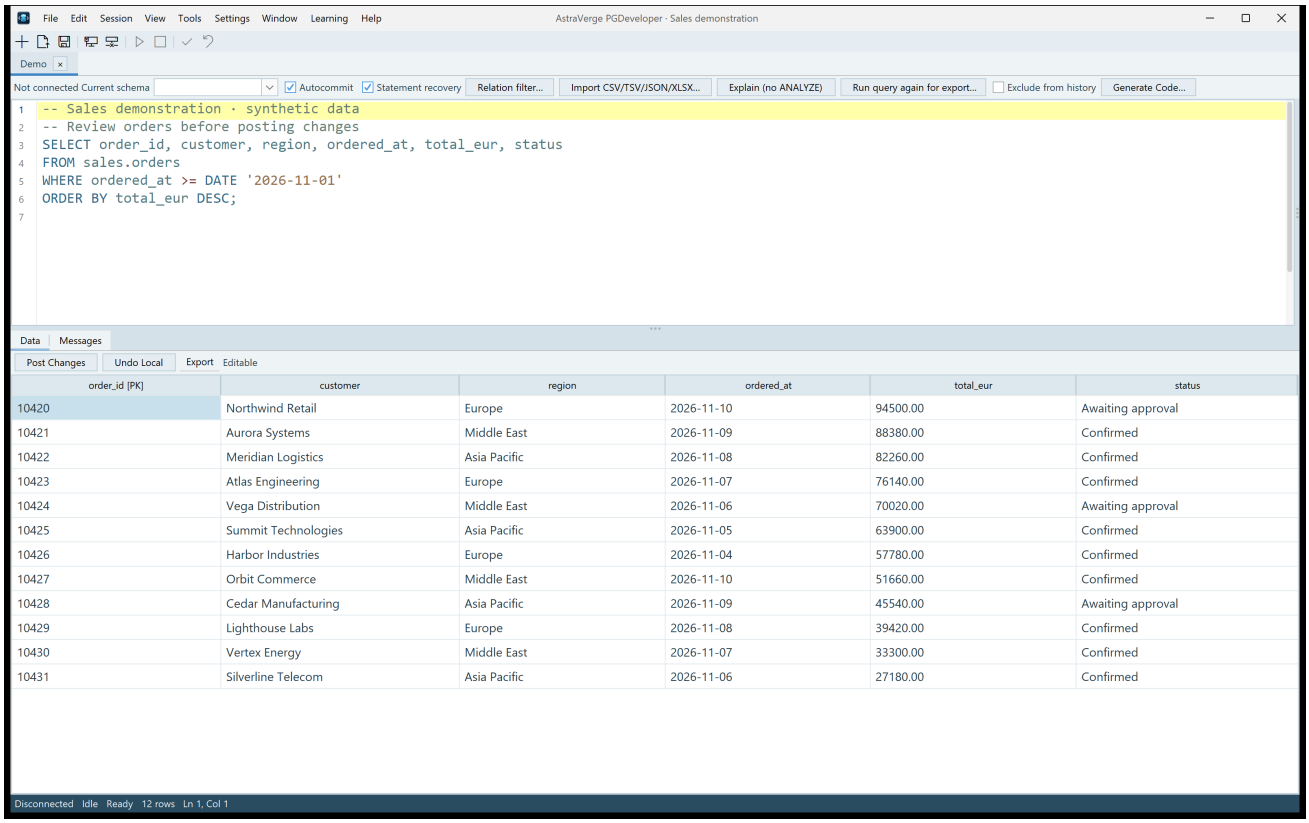
Задача	Возможность	Практическая ценность
Разобраться в базе	Object Browser, DDL и ER-диаграммы	Видеть структуру и связи до изменения
Написать и проверить запрос	Подсветка SQL, дополнение, параметры, планы	Работать с запросом и его результатом рядом
Исправить данные	Result Grid, Post, Commit, Rollback	Отделять локальное редактирование от подтверждения транзакции
Подключить приложение	Generate Code для языков и контрактов	Получать исходный текст для проверки и включения в проект
Передать результат	Предпросмотр, сопоставление столбцов, экспорт	Проверять формат и целевую структуру перед записью
Исследовать данные	R-анализ, консоль, отчёты	Перейти от выборки к статистическому исследованию

Для кого

Разработчику - запросы, DDL и код доступа к данным. DBA - диагностика, привилегии, backup/restore и сравнение структур. Аналитику - выборки, экспорт и R-сценарии. Руководителю команды - единый демонстрируемый рабочий процесс от данных до приложения.

Состав и границы

Текущая линия поддерживает PostgreSQL 14-18. Для PostgreSQL 19 предусмотрен отдельный предварительный режим; его не следует выдавать за стабильную поддержку. Поставляемая Windows-сборка содержит Java runtime. R устанавливается отдельно, аналитический контур требует своей настройки и доступа. Backup/restore зависит от подходящих утилит PostgreSQL. Это клиентская среда: серверные права и ресурсы остаются под управлением PostgreSQL.



SQL и результаты

Реальный интерфейс; демонстрационный набор заказов, запрос в редакторе не выполнялся на сервере при съёмке.

Отладчик PL/pgSQL

Назначение и запуск

Отладчик помогает разбирать выполнение функций и процедур PL/pgSQL: видеть текущую строку, просматривать переменные и стек вызовов, находить место расхождения с ожидаемым результатом. В Object Browser выберите функцию или процедуру и команду **Debug routine**, затем задайте параметры вызова и нажмите **Start debug call**.

Пошаговая работа

Step into выполняет шаг с заходом во вложенный вызов, Step over - без захода, Step out возвращает управление из текущего вызова. Continue продолжает выполнение до следующей остановки или завершения. Точки останова устанавливаются и снимаются на строке под курсором. Вкладки Variables, Watches, Call stack и Breakpoints показывают состояние; выбор кадра стека переключает контекст просмотра.

Можно наблюдать выбранную переменную, изменять значение неконстантной переменной серверным выражением и выполнять шаги до точного совпадения её текстового предпросмотра с заданным значением. Такой поиск ограничен 1000 шагами и не является произвольным условием SQL. Предпросмотры значений ограничены 4000 символами; усечённые значения нельзя использовать для проверки совпадения.

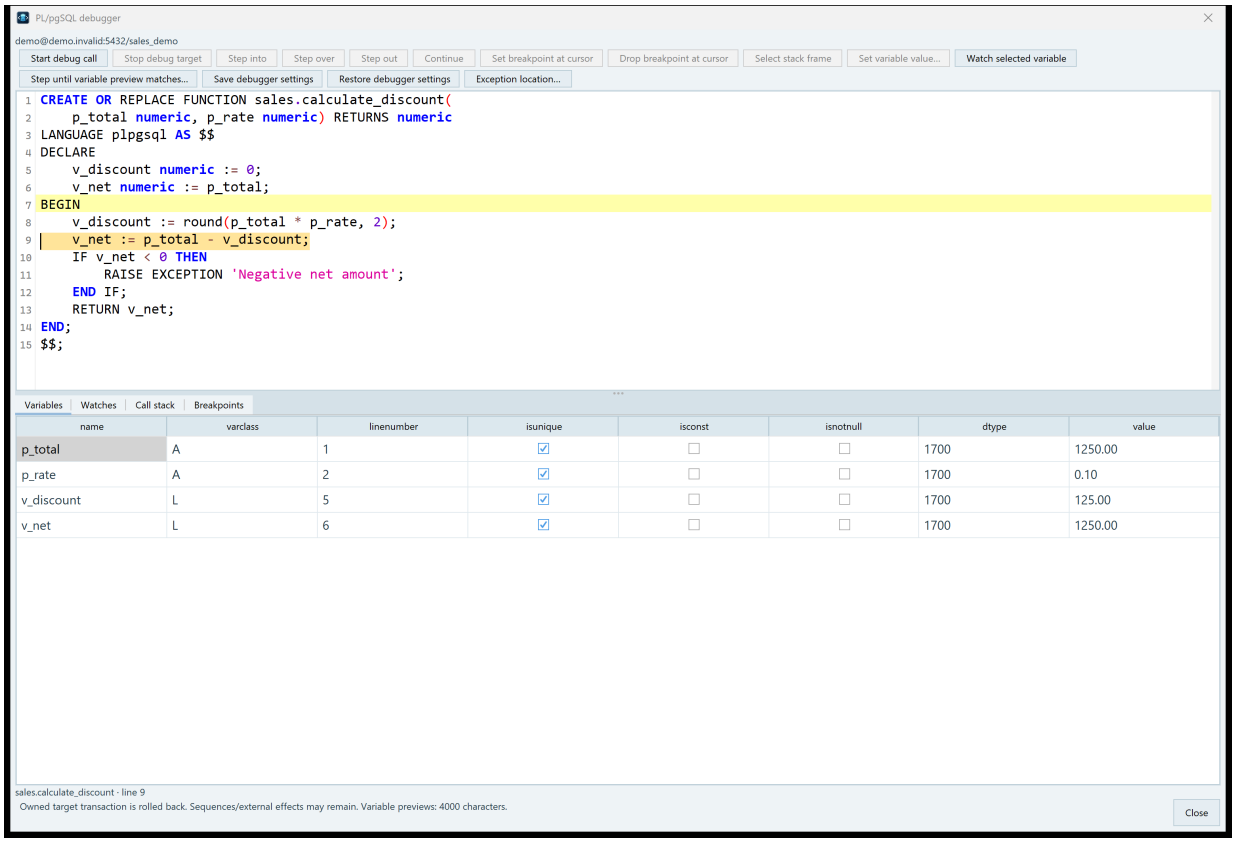
Настройки точек останова и наблюдений сохраняются локально. Восстановление строковых точек проверяет хеш исходного текста: после изменения функции прежние позиции автоматически не применяются. Команда Exception location открывает контекст ошибки завершившегося вызова.

Серверные требования и границы

Нужны расширение pldbgapi и загруженный plugin_debugger на сервере, права владельца процедуры/функции либо superuser и соединение, допускающее запись. Приложение проверяет доступность, но не устанавливает расширения и не меняет конфигурацию PostgreSQL. Прямая отладка триггеров и отладка в чужом контексте не поддерживаются.

Отладчик использует собственные соединения управления и целевого вызова. Транзакция собственного вызова откатывается при завершении или остановке. Последовательности и внешние эффекты серверного кода откатом не отменяются: для знакомства используйте согласованную тестовую среду. Нажатие Stop сообщает о запросе остановки/отката; оно само по себе не доказывает отсутствие всех внешних эффектов.

Интерфейс



Отладчик PL/pgSQL: исходный текст и переменные

Настоящий DebuggerDialog, демонстрационный исходный текст и синтетический снимок переменных. Подключение к PostgreSQL не создавалось, функция не выполнялась. Изображение показывает интерфейс, а не испытание серверного расширения; команды шагов без активной сессии недоступны.