
# Модуль "Cloud Database Professional"
# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# Содержание

# 1 Введение

Модуль экспорта данных "Cloud Database Professional" для наших программ-логгеров (например, Advanced Serial Data Logger) используется для записи данных в популярные облачные SQL-совместимые базы данных: Microsoft Azure, MySQL, PostgreSQL, MariaDB, MongoDB, Amazon Aurora, Amazon Redshift.

Модуль обращается к базам данных напрямую, без использования дополнительных драйверов. Это позволяет оптимизировать трафик между компьютером и базой данных, повысить скорость передачи данных, а также позволяет использовать особенности каждой базы данных (например, хранимые процедуры в Microsoft SQLServer или MySQL).

Модуль отправляет данные в базу данных в реальном времени по защищенному соединению и может создавать локальную копию данных, пока удаленная база данных не доступна.

Модуль позволяет выполнять один или несколько SQL запросов в очереди.

# 2 Системные требования

Для установки Cloud Database Professional должны быть выполнены следующие требования:

Операционная система: Windows 2000 SP4 и выше, включая версии для 32-х и 64-х битных систем, а также серверные версии операционных систем.
Наличие установленного последнего сервис-пака желательно.

Свободное дисковое пространство: Рекомендуется не менее 5 MB свободного дискового пространства.

Специальные требования для доступа: Вам потребуются права Администратора для установки модуля. Для некоторых модулей могут потребоваться права администратора для выполнения заданных функций.

Необходимо наличие главного приложения (ядра), например Advanced Serial Data Logger.

# 3 Установка Cloud Database Professional

1. Закройте главное приложение (например, Advanced Serial Data Logger), если оно запущено;
2. Скопируйте программу на ваш жесткий диск;
3. Запустите программу установки модуля, кликнув два раза на имени файла в проводнике Windows;
4. Следуйте инструкциям программы установки. Обычно достаточно нажать несколько раз кнопку "Дальше";
5. Запустите главное приложение. В случае успешной установки название модуля появится в окне настройки, на закладке "Модули".

Если модуль совместим с программой, то его название и номер версии будет отображаться в списке модулей. Примеры установленных модулей можно посмотреть на рис. 1-2. Некоторые типы модулей требуют дополнительной настройки. Для этого достаточно выбрать модуль из списка и нажать кнопку "Настроить" рядом со списком. Процедура настройки модуля описана в следующих главах.

Некоторые типы модулей видны на закладке "Лог-файл". Для их настройки необходимо выбрать модуль из списка "Тип файла" и нажать кнопку "Дополнительно".

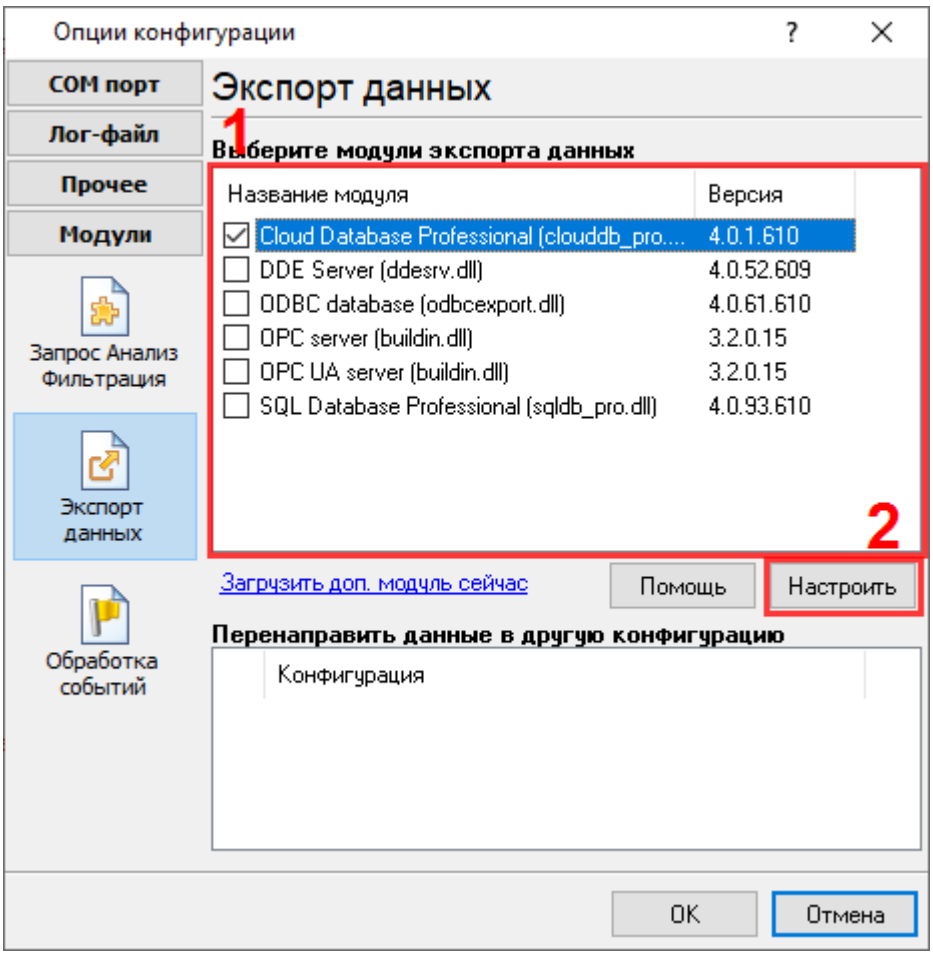


**Рис. 1. Пример установленного модуля**

# 4 Глоссарий

**Plug-in** - модуль

**Главная программа** - программная оболочка, которая использует данный модуль. Например: Advanced Serial Data Logger

**Основная программа** - см. "Главная программа".

**Парсер** - модуль, который обрабатывает поток данных, выделяя из него пакеты данных и переменные из пакетов данных. Затем эти переменные используются в модулях экспорта данных.

**Ядро** - см. "Главная программа".

# 5 Настройка соединения

Настройка модуля осуществляется с помощью дополнительного диалогового окна. Чтобы перейти к окну настройки модуля необходимо:

1. Запустить, если не была запущена ранее, программу-логгер (например, Advanced Serial Data Logger).
2. Выбрать в главном Опции - Управление конфигурациями - Настройка… или нажать кнопку  на панели инструментов.
3. Открыть в опциях программы закладку "Модули - Экспорт данных".
4. На этой закладке в списке модулей публикации выбрать модуль с именем "Cloud Database Professional". Если модуль публикации с таким именем отсутствует, то обратитесь к разделу "Установка" и убедитесь, что вы правильно выполнили все шаги по установке модуля.
5. Нажмите кнопку "Настроить".

# 6 Свойства соединения

Модуль предлагает вам гибко настроить свойства соединения. Модуль может как поддерживать постоянное соединение с базой данных, так и устанавливать соединение с базой данных по мере необходимости. Настройка параметров данной группы осуществляется на закладке "**Режим соединения**" (рис. 2)

Для того, чтобы включить модуль необходимо чтобы не было "галочки" в поле "**Временно отключено**". Вы можете установить "галочку" в этом поле, чтобы временно приостановить операции с базой данных. Это бывает полезно в период изменения конфигурации модуля или на время операций по администрированию базы данных.

С помощью опций "**Оставаться на связи**", "**Отсоединяться после каждой операции**" и "**Отсоединяться когда нет активности**" вы можете указать тип соединения. Первая опция указывает модулю, что соединение с базой данных необходимо устанавливать при первой необходимости и поддерживать его до момента закрытия программы. Вторая опция "**Отсоединяться после каждой операции**" указывает на то, что модуль будет устанавливать соединение с базой данных при каждой операции с базой данных и затем разрывать соединение (т.е. после того, как будут опубликованы все переданные для публикации данные). Третья опция "**Отсоединяться когда нет активности**" указывает на необходимость устанавливать соединение при первой необходимости и разрывать соединение в случае, если операций по публикации данных не было в течении количества секунд, заданного в поле "**Отсоединяться через**". Данную опцию рекомендуется использовать в случае, если данные

поступают нерегулярно и через большие интервалы времени. Это позволяет снизить загрузку сети за счет сокращения количества пустых обращений.

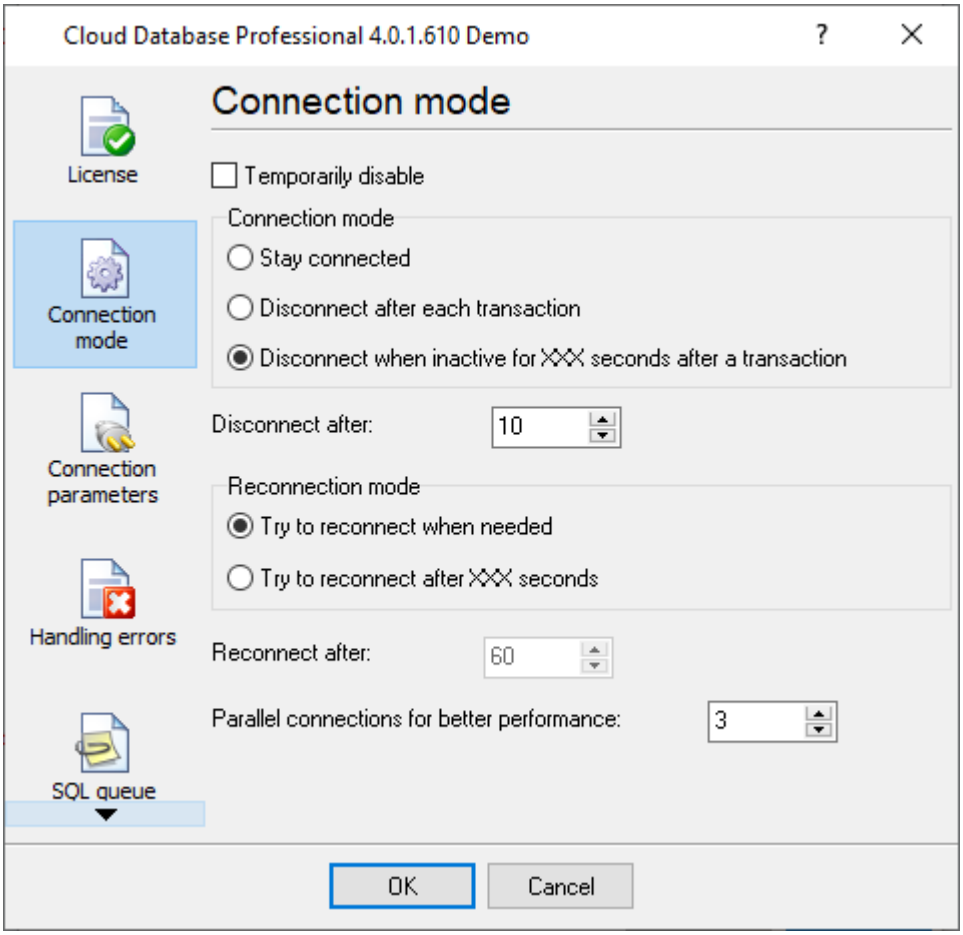


**Рис. 2. Свойства соединения**

За время длительного соединения, связь с базой может быть утеряна по причине обрыва линии связи или "падения" базы данных. Опции в группе "Режим переподсоединения" позволяют установить режим, в котором программа будет пытаться повторно устанавливать связь с базой данных.

- **Пробовать переподсоединиться при необходимости** – попытка установить связь с базой данных будет осуществляться при очередном обращении к модулю;
- **Пробовать переподсоединиться через XXX секунд** - попытка установить связь с базой данных будет осуществляться при очередном обращении к модулю, но не ранее чем через количество секунд, указанное в поле "**Соединяться через**".

Параметр "**Параллельных соединений**" задает максимальное количество одновременных подключений к базе. Это может потребовать при большом потоке данных или при восстановления данных из резервной копии после разрыва соединения с базой данных. Это число должно соответствовать максимальному числу подключений, установленному в настройках базы данных.

# 7 Параметры соединения

Опции, описанные в предыдущем разделе, устанавливают свойства физического взаимодействия, тогда как параметры соединения описывают программный уровень взаимодействия. Настройка данных параметров осуществляется на закладке "Параметры соединения" (рис. 3)

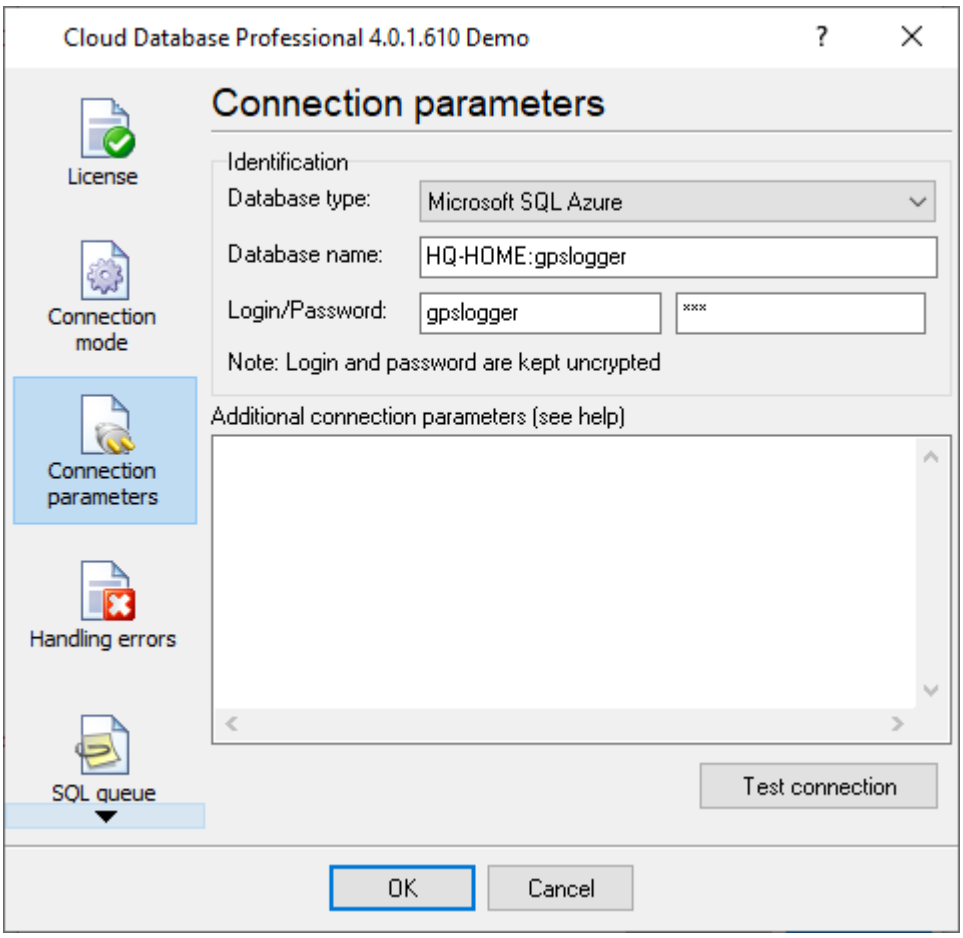


**Рис. 3. Параметры соединения**

В группе **"Авторизация"** описывается тип базы данных (выбирается из списка "**Тип базы**").

В зависимости от выбранного типа базы данных в поле **"Имя базы данных"** указывается:

адрес_сервера:имя_базы_данных

Примеры:

Aurora:
database-1.cluster-copwtiaj8an.us-east-1.rds.amazonaws.com:mydb

Azure:
aggsoft-test.database.windows.net:test

MongoDB:
cluster0-shard-00-02.j4arl.mongodb.net:test

Redshift конечная точка подключения:
redshift-cluster-1.cm01xwy5h8ea.us-east-1.redshift.amazonaws.com:dev

Далее задается имя пользователя и пароль для доступа к базе данных в полях "**Логин**" и "**Пароль**" соответственно.

В поле ввода "**Дополнительные параметры соединения**" указываются атрибуты подключения, специфичные для каждого из серверов.

| Значение | Описание | По умолчанию | Примечание | Пример |
|---|---|---|---|---|
| SERVER PORT | Если база данных использует нестандартный порт для подключения | Стандартный порт для базы данных | Все базы данных | SERVER PORT=8897 |
| SSL KEY | Путь к файлу клиентского ключа SSL (*.pem) для вашего сервера | Не используется | MySQL, MariaDB, PostgreSQL | SSL KEY=c:\MySQL8\data\client-key.pem |
| SSL CERT | Путь к файлу клиентского сертификата SSL (*.pem) | Не используется | MySQL, MariaDB, PostgreSQL | SSL CERT=c:\MySQL8\data\client-cert.pem |
| SSL CA | Путь к файлу сертификату удостоверяющего центра (*.pem) | Не используется | MySQL, MariaDB, PostgreSQL | SSL CA=c:\MySQL8\data\ca.pem |
| SSL CIPHER | (Опционально) Тип желаемого метода шифрования | Не используется | MySQL, MariaDB, PostgreSQL | SSL CIPHER=TLS_AES_128_CCM_SHA256 |
| COMPRESSED PROTOCOL | Включить сжатие данных | Не используется | MySQL, MariaDB | COMPRESSED PROTOCOL=TRUE |
| LOGIN TIMEOUT | Установить пользовательское значение времени ожидания подключения (в секундах) | 120 | MySQL, MariaDB, PostgreSQL | LOGIN TIMEOUT=10 |
| LOCAL CHARSET | Установить желаемый набор символов для данного подключения | Набор символов базы данных | MySQL, MariaDB | LOCAL CHARSET=utf-8 |
| Connection Options | Установить параметры защищенного соединения, если требуется | Не используется | MongoDB | ConnectionOptions=ssl=true |

После того, как вы настроили подключение к базе данных, вы можете немедленно это проверить, нажав на кнопку "Проверить связь". При этом программа попытается соединиться с базой данных. Это может занять довольно длительное время (до трех минут) в зависимости от типа базы данных. Результат проверки будет выведен в виде сообщения. В случае ошибки сообщение будет содержать ответ сервера, который поможет выявить причины ошибки.

# 8 Amazon Aurora и Amazon Redshift

Вы должны создать правила, которые позволяют удаленные подключения к вашему серверу в вашем приватном облаке.

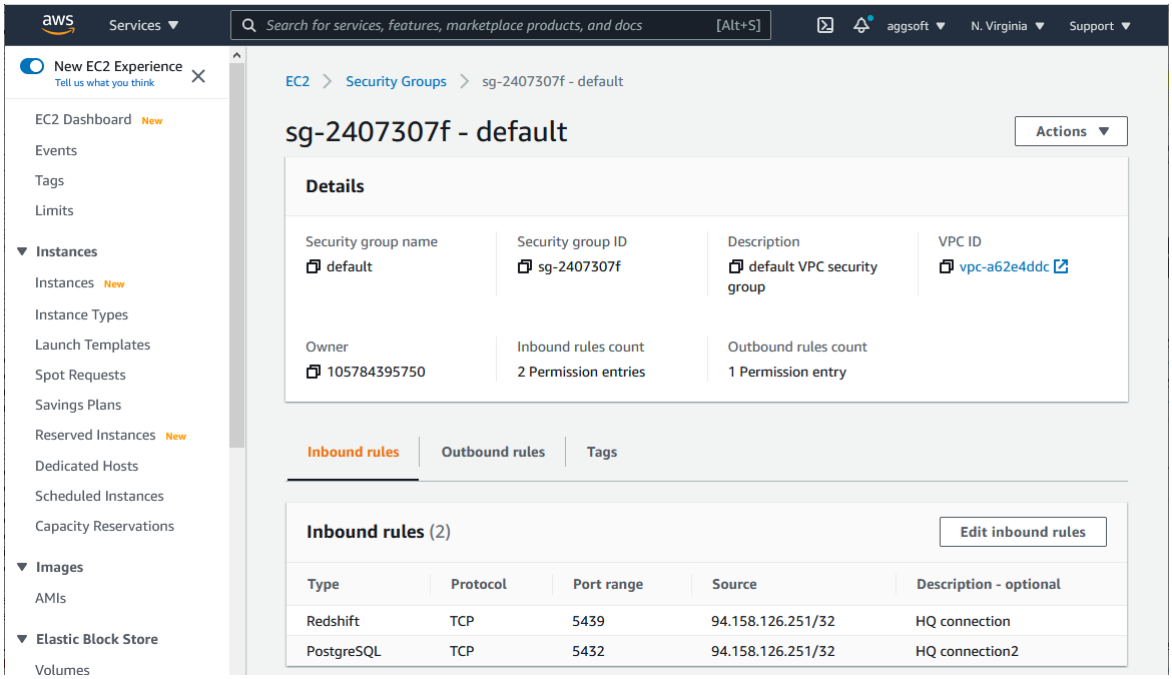


**Рис. 4. Группа параметров безопасности Amazon**

## Amazon Redshift

1. Перейдите к странице Redshift console --> Clusters в панели управления и кликните на закладке "Properties".

2. Прокрутите страницу вниз до группы параметров "Network and Security". Убедитесь, что ваш кластер доступен извне. Параметр "Publicly Accessible" должен иметь значение "Yes". Затем кликните на "VPC Security Group", для того чтобы проверить правила.

3. В окне "Security Group" выберите закладку "Inbound".

4. Там должны быть правила, которые разрешают подключения с ваших компьютеров. Если нужно что-то добавить или изменить здесь, то нажмите "Edit".

5. Отредактируйте существующее правило или добавьте новое, используя кнопку "Add Rule". Для каждого правила выберите тип базы данных и введите номер порта, используемый Redshift. Затем нажмите "Save".

6. Для настройки защищенного соединения (SSL) перейдите к первой группе параметров и установите для параметра "require_ssl" значение "true". Затем, перейдите к Config --> Workload management.

# 9 MongoDB

Этот пример сделан на основе кластера в MongoDB Atlas.

1. Вы должны добавить правила, разрешающие входящие соединения для вашего виртуального приватного облака.

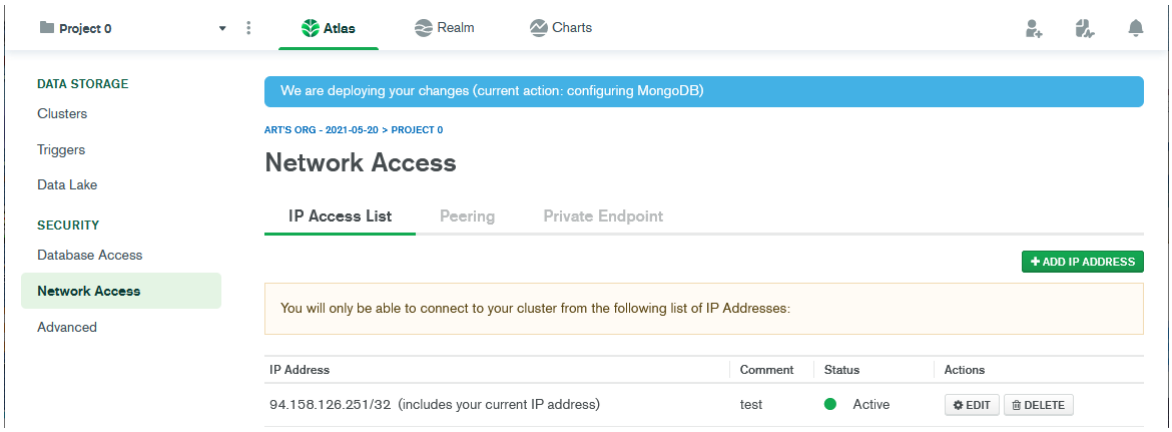


**Рис. 5. Список IP адресов**

2. Вы должны использовать адрес главной (primary) конечной точки при подключении к базе данных.

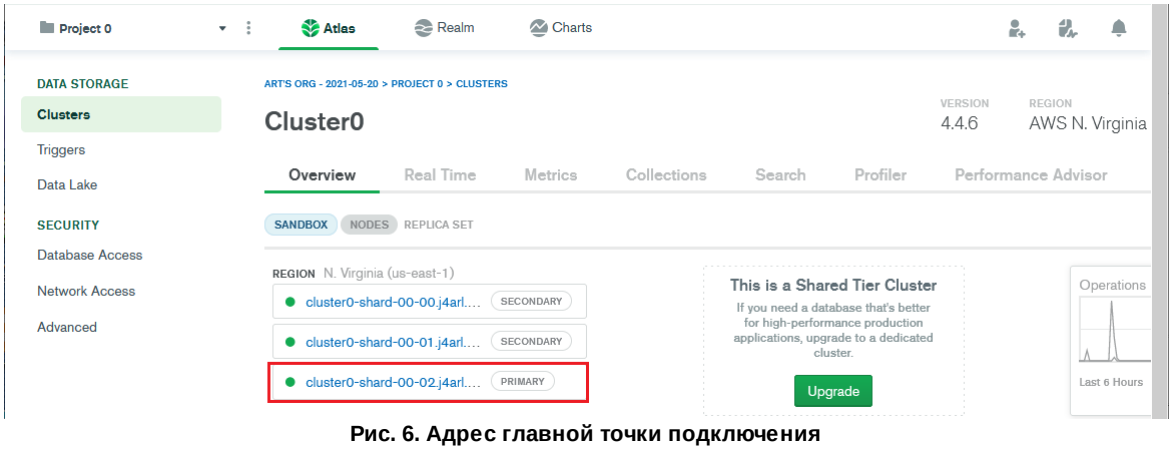


**Рис. 6. Адрес главной точки подключения**

3. Создайте новую базу данных и коллекцию данных в ней.

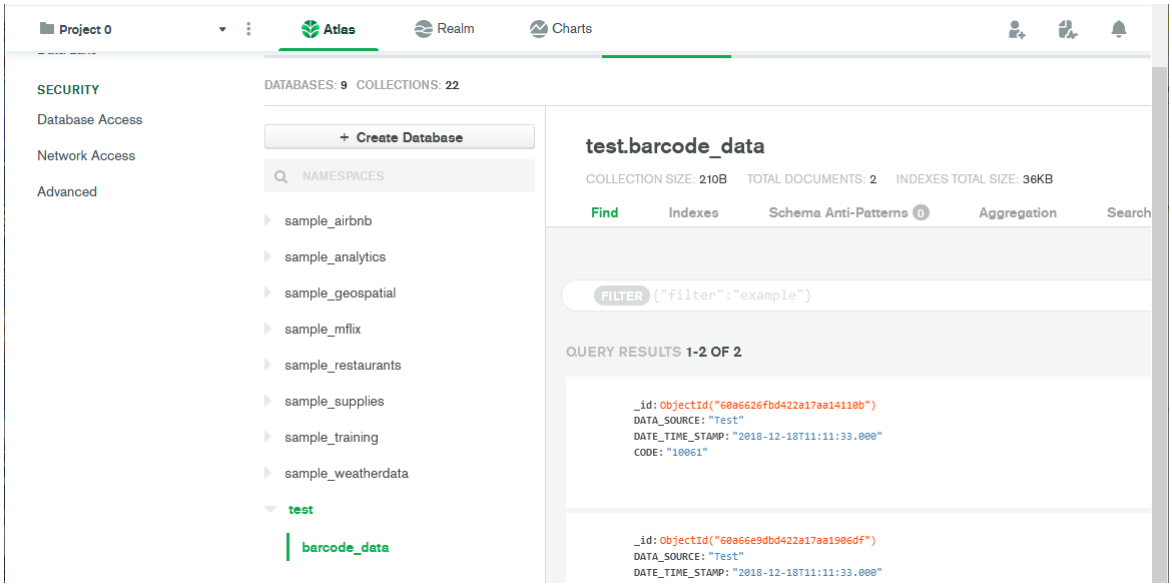


**Рис. 7. Базы данных и коллекции**

4. Параметры защищенного соединения

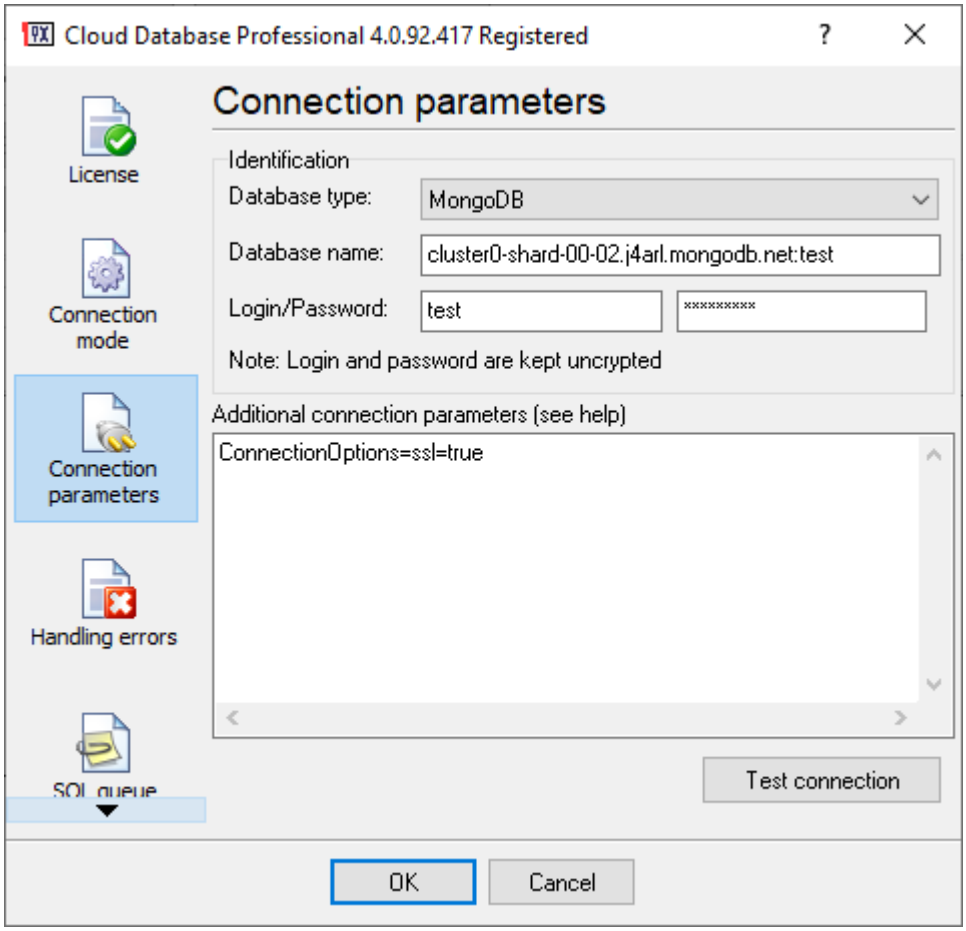


**Рис. 8. Параметры соединения**

5. Запрос типа "INSERT" на вставку данных в MongoDB.

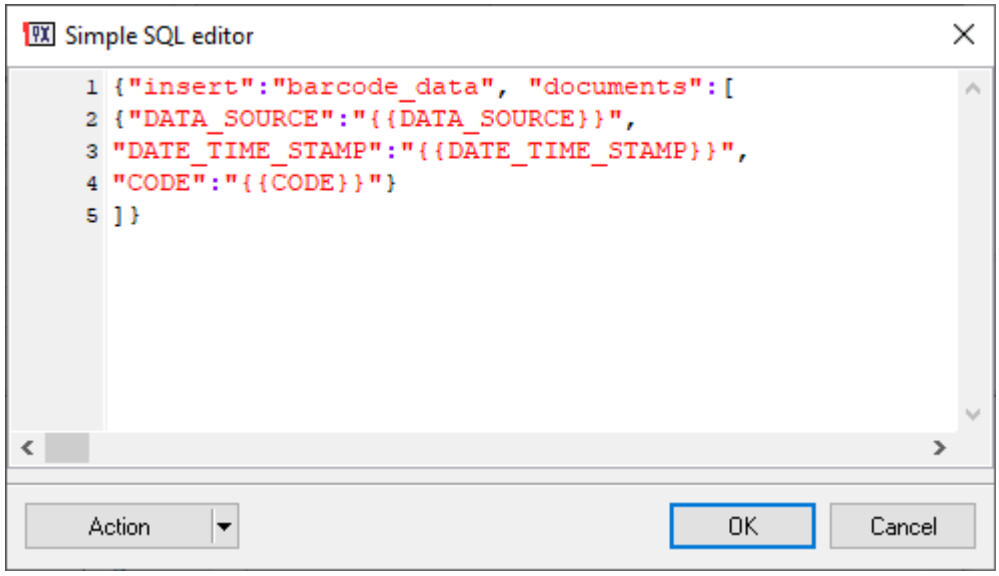


**Рис. 9. Запрос на вставку данных**

# 10 Обработка ошибок

Во время выполнения модулем своих задач могут возникать ошибки при взаимодействии с базой данных. Такие, например, как: нарушение уникальности (PRIMARY KEY) при записи данных в таблицу, нарушение ограничения целостности (FOREIGN KEY) данных, потеря соединения с базой данных и т.п. Вы можете настроить реакцию модуля на ошибки данного рода. Данные установки никак не влияют на обработку внутренних ошибок модуля, обработка которых жестко запрограммирована и нацелена на обеспечение максимальной "живучести" модуля. Настройка параметров данной группы осуществляется на закладке "Обработка ошибок" (рис. 10)

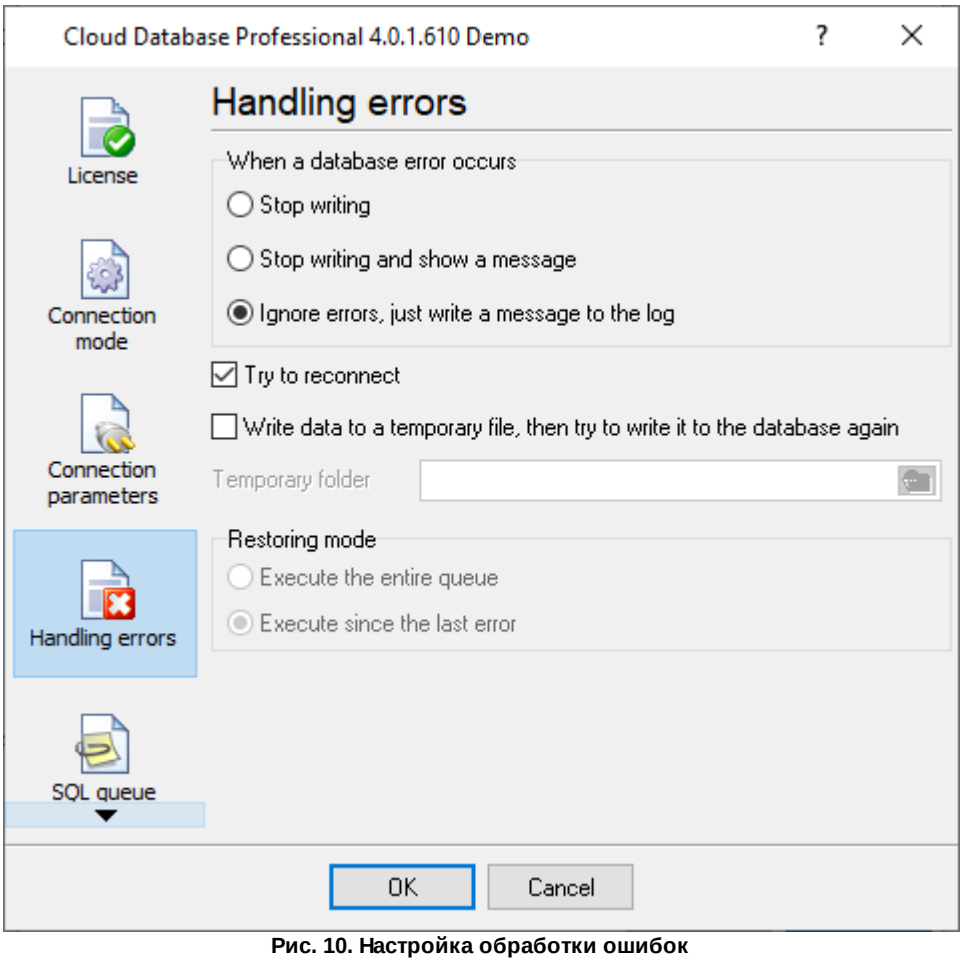


**Рис. 10. Настройка обработки ошибок**

Существует четыре варианта реакции модуля на ошибки:

1. **Остановить запись** – в случае возникновения ошибки программа генерирует сообщение и устанавливает внутренний флаг "**Временно отключено**", т.е. все операции по публикации данных прекращаются до того момента, как будет изменена конфигурация модуля. При этом все данные, которые будут поступать на публикацию после возникновения ошибки НЕ будут записаны во временный файл и базу данных. Данная опция бывает полезна в момент настройки конфигурации модуля.
2. **Остановить запись и показать сообщение** – в случае возникновения ошибки программа генерирует сообщение, устанавливает внутренний флаг "**Временно отключено**" и отображает сообщение на экране в виде диалогового окна, т.е. все операции по публикации данных прекращаются до того момента, как будет закрыто диалоговое окно. При этом все данные, которые поступают на публикацию во время отображения диалогового окна НЕ будут записаны во временный файл и базу данных. В случае утвердительного ответа в отображаемом диалоговом окне модуль возобновит все дальнейшие операции. Данная опция бывает полезна в момент настройки конфигурации модуля.
3. **Игнорировать ошибки и записать в журнал** – в случае возникновения ошибки программа генерирует сообщение и все операции продолжаются, как и заложено в конфигурации модуля.
4. **Пробовать переподсоединиться** – данная опция аналогична предыдущей, за исключением того, то модуль произведет отсоединение от базы данных и попытается его восстановить при следующем обращении к модулю. Данная опция необходимо в случае

частого пропадания связи с базой данных (по вине базы данных или линии связи), т.к. при не всегда по содержанию ответа сервера можно автоматически определить его причину и выполнить необходимее действия.

В двух последних случаях, чтобы избежать потери публикуемых данных существует возможность сохранения их во временных файл, который будет формироваться в папке "**Временная папка**". Данные во временный файл будут помещаться только при включенной опции "**Записывать данные в файл**". В данный файл будут помещаться все данных, при попытке опубликовать которые возникла ошибка. Позднее, при следующем обращении, модуль будет пытаться записать данные из файла в базу данных. Это опция просто необходимо при пропадании связи с базой данных. Однако, если цепочка SQL запросов построена неверно, и при ее выполнении постоянно (или довольно часто) возникает ошибка, это может привести к значительному увеличению затрачиваемых ресурсов сети и сервера за счет многократных попыток записи данных из временного файла. Но при этом можно быть уверенным, что при исправлении возникающей ошибки, все данные из временного файла попадут в базу данных. Стоит заметить, что в поле "**Временная папка**" необходимо указывать путь к папке, к которой имеется доступ на запись и чтение у пользователя, от имени которого запускается программа. В противном случае модуль не сможет создать файл и данные будут утеряны безвозвратно. Если поле "**Временная папка**" оставить пустым, то программа будет создавать файл в папке, куда был установлен модуль.

Поскольку у Вас есть возможность составить очередь SQL запросов, которая будет выполняться последовательно, а ошибка может возникать при выполнении любого SQL запроса в цепочке, то модуль может при попытке записи данных из файла выполнить заново как всю цепочку SQL запросов (опция "**Выполнить всю очередь**"), так и продолжен выполнения с места возникновения ошибки (опция "**Выполнить всю очередь с места последней ошибки**").

Первая опция необходимо, например, когда первыми в цепочке стоят SELECT запросы, которые выбирают данных из связанных таблиц, а ошибка возникает при записи данных по INSERT в другую таблицу.

Вторая опция допустима, когда ваша цепочка состоит из нескольких INSERT, и чтобы избежать создания дублирующих записей, необходимо пропускать успешно выполненные SQL запросы.

### Сообщения модуля

В ходе выполнения своих действий модуль генерирует три сообщений: ошибки, предупреждения, информационные сообщения. Данные сообщения отображаются в главном окне программы. Детализация отображаемых сообщений настраивается с помощью опций ядра программы, например, Advanced Serial Data Logger. За более детальным описанием этой процедуры обратитесь к руководству пользователя по основной программе. Стоит уточнить, что сообщения, генерируемые модулем, относятся к группе "Экспорт данных".

## 11 Очередь SQL запросов

Очень часто данные в базу данных записываются с использование INSERT или хранимой процедуры. При выполнении INSERT или хранимой процедуры требуется получить или записать в базу данных вспомогательные данные. Почти всегда это нельзя сделать одним SQL запросом. Наш модуль позволяет создать очередь SQL запросов (далее по тексту "очередь"),

которые будут выполняться последовательно, начиная с самого верхнего. Причем данные, полученные с использованием SELECT могут быть доступны последующим SQL запросам. Это позволяем расширить сферу применения нашего модуля и отказаться от написания сложных вложенных SQL запросов или специальных хранимых процедур. Настройка очереди осуществляется на закладке "**Очередь SQL**" (рис. 11)

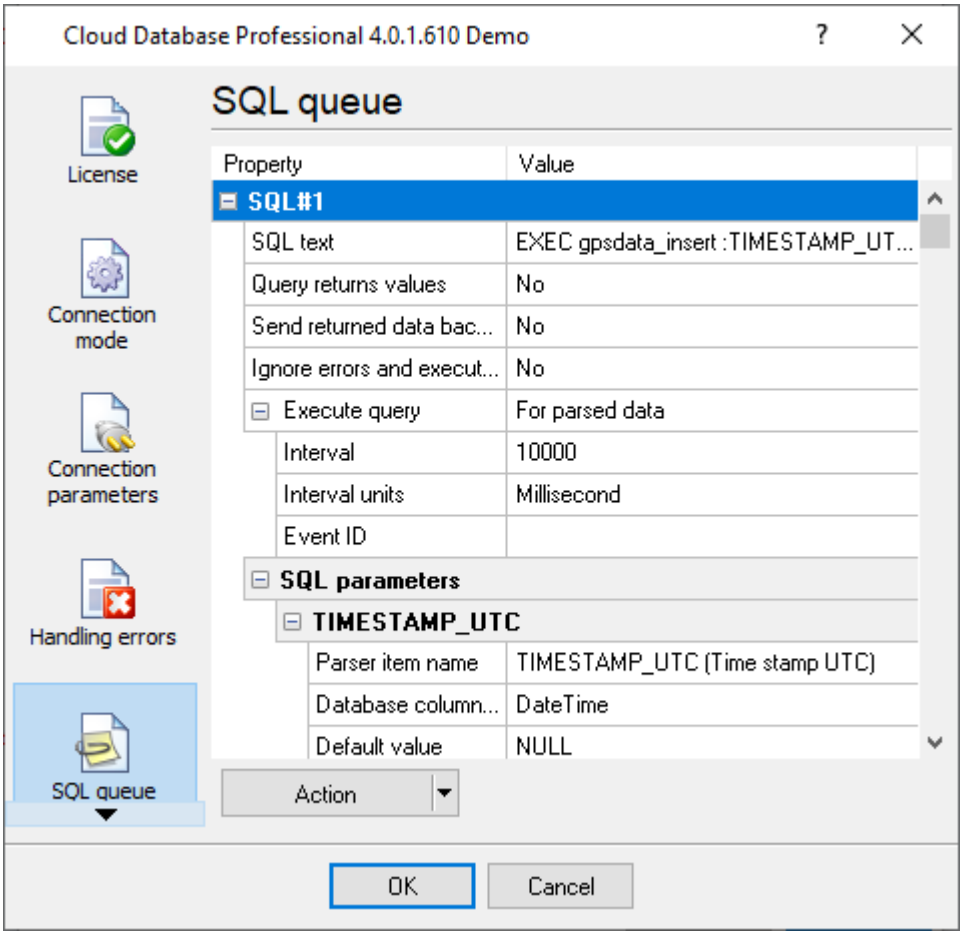


**Рис. 11. Очередь SQL запросов**

Все действия над очередью выполняются с помощью всплывающего меню или при помощи кнопки "Действие" над выбранным SQL запросом. Для того, чтобы выбрать SQL необходимо кликнуть по его заголовку (синяя строчка на рис. 11) или на любом из параметров SQL запроса

**Добавить SQL в очередь** – добавляет новый SQL запрос в конец очереди и перемещает фокус на него

**Удалить SQL из очереди** – удаляет выбранный SQL из очереди.

**Переместить SQL вверх**, **Переместить SQL вниз** – перемещает выбранный SQL в очереди вверх и вниз соответственно.

**Загрузить очередь SQL из файла** – загружает новую очередь из специального файла. Эта опция полезна при переносе конфигурации с одного компьютера на другой.

**Сохранить всю очередь SQL в файл** – сохраняет всю очередь в специальный файл. Эта опция полезна при переносе конфигурации с одного компьютера на другой. Созданный файл можно загрузить с помощью предыдущей функции.

# 12 Создание нового SQL запроса

Для создания нового SQL запроса в очереди выполните действие "**Добавить SQL в очередь**", описанное выше. После этого в очереди появится новый SQL запрос (рис. 12) и его параметры. Запросу автоматически будет присвоено имя, содержащее его порядковый номер.

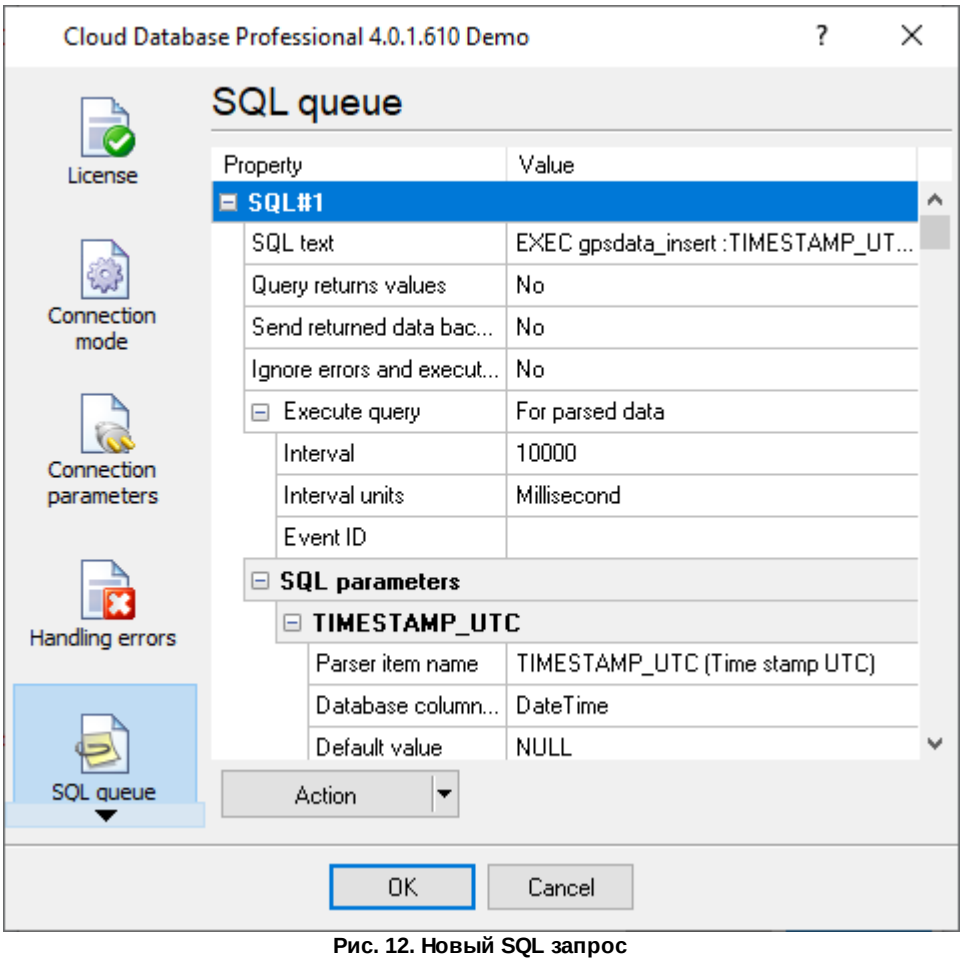


**Рис. 12. Новый SQL запрос**

**Текст SQL** – в данном поле задается текст SQL запроса. Для того, чтобы ввести текст SQL запроса кликните на столбце "Значение" и нажмите на появившейся справа кнопке. Появится окно простого SQL редактора (рис.13). В котором вы должны ввести текст SQL запроса. При составлении SQL запросов вы можете использовать параметры. Запись вида ":P1" означает, что это параметр с именем "P1". В дальнейшем вы сможете сопоставить (привязать) этому параметру переменную, созданную разборщиком (парсером) данных. Вы также можете сохранить или загрузить SQL в/из файла. После окончания редактирования вы можете нажать "OK" для сохранения изменений или "Отмена" для их отмены.

**Запрос возвращает значения** – указывает на то, что данный запрос возвращает данные (например, SELECT). Данный параметр может принимать два значения "Да" и "Нет", которые выбираются из списка, появляющегося при клике на столбец "Значение"

**Отправлять данные запроса в источник данных** – если запрос возвращает строковое значение, то программа интерпретирует его как набор байт и отправляет в источник данных. Это позволяет при вызове хранимой процедуры возвращать ответ к устройству. Данные для устройства должны быть подготовлены на стороне базы данных, поскольку отправляются в источник данных без изменений и перекодировки.

**Игнорировать ошибки и выполнять следующий** – Указывает на то, что если при выполнении данного запроса возникает ошибка, то нужно ее проигнорировать и продолжить выполнение очереди дальше. Данный параметр может принимать два значения "Да" и "Нет", которые выбираются из списка, появляющегося при клике на столбец "Значение". Это необходимо в случае, если действие, которое выполняет данный SQL второстепенно и от него не зависит ход выполнения других SQL запросов в очереди.

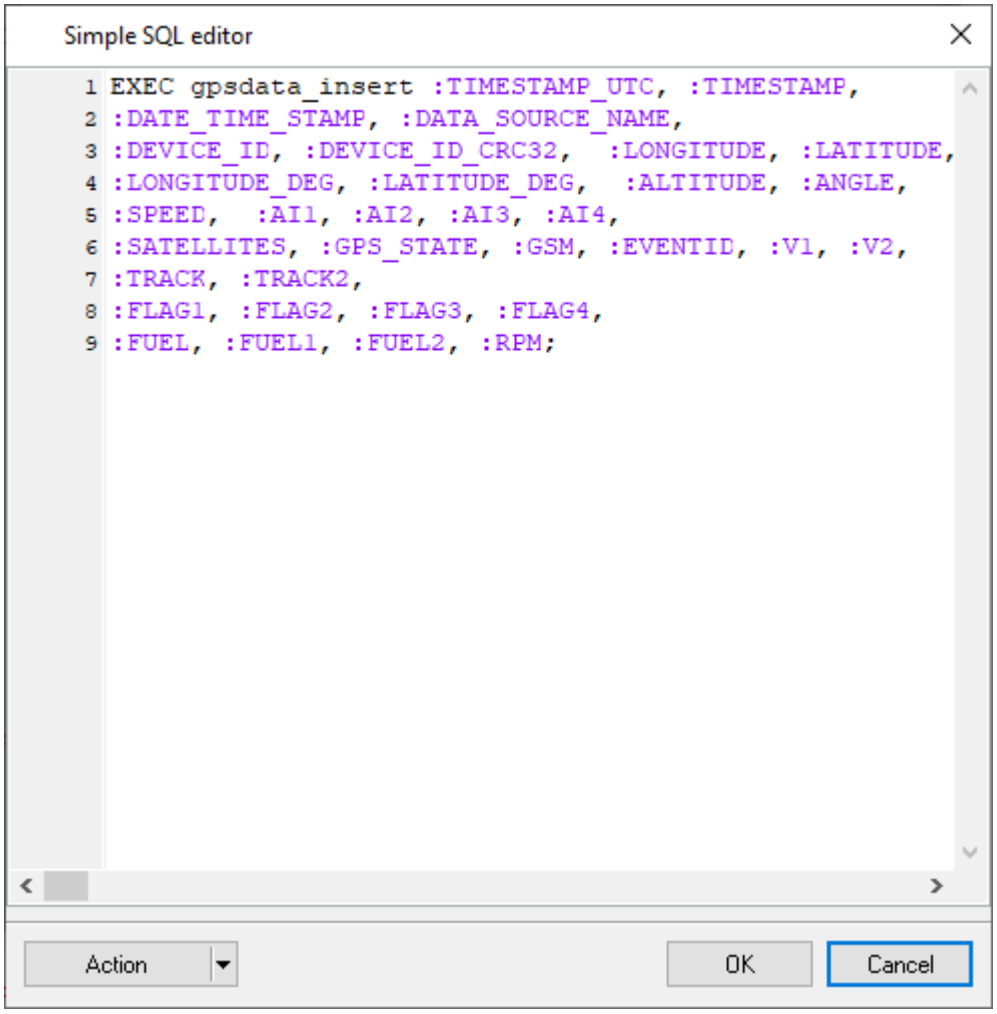


**Рис. 13. SQL редактор**

После сохранения изменений в SQL редакторе окно его закроется, а текст SQL запроса будет помещен в поле "Текст SQL" (рис. 12). Далее текст запроса будет подвергнут анализу и из него будут выделены (если есть) имена параметров. Имена всех параметров будут отображены в группе "**Параметры SQL**".

Каждый из параметров имеет 3 свойства:

**Имя переменной** – имя переменной, которую создает разборщик (парсер) данных (для любой операции публикации необходимо настроить парсер. Как это сделать читайте в руководстве по Advanced Serial Data Logger). В данном поле вы можете как выбрать имя переменной из списка, так ввести его вручную. Помимо переменных, которые создает парсер в списке всегда существуют две предопределенные переменные: NULL и DEFAULT, обозначающие, что значение данного параметра всегда будет NULL или значение по умолчанию, которое задается в поле "**Значение по умолчанию**".

Вручную вводятся имена переменных, которые образуются путем выполнения запросов очереди, возвращающих данные. Это получается следующим образом. Например, первым в очереди выполняется SQL запрос:

```
select (max(id)+1) as max_id from test_datas
```

Результатом этого запроса будет максимальное значение столбца ID, которое будет присвоено имя max_id. Поэтому вручную, в последующих запросах, вводится имя переменной MAX_ID (регистр символов не имеет значения). Причем имя MAX_ID может использоваться в нескольких SQL запросах одновременно. Стоит отметить, что если MAX_ID будет иметь значение null (например, таблица test_datas не содержит записей), то в нашем примере при использовании этой переменной в паре с параметром P1 (см. рис. 13) будет использовано значение по умолчанию 1.

Операцию по привязке переменных необходимо произвести для всех параметров, которые вы указали в SQL запросе.

**Тип данных поля таблицы** – определяет тип параметра, запись в который осуществляется через указанный параметр. В зависимости от указанного типа модуль осуществляется преобразование публикуемых данных к нужному типу.

**Значение по умолчанию** – значение по умолчанию, которое будет использовано, если переменная, с указанным именем не передана для публикации или имеет значение null.