

Модульная интегрированная

SCADA КРУГ-2000<sup>™</sup>

**ODBC-ДРАЙВЕР**

Версия 4.4

Руководство Пользователя

© 1992-2023. ООО НПФ «КРУГ». Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

---

---

## **НПФ «КРУГ»**

440028, г. Пенза, ул. Титова, 1

Телефоны: (841-2) 49-97-75; 49-94-14; 48-34-80

E-mail: [support@krug2000.ru](mailto:support@krug2000.ru)

http:// [www.krug2000.ru](http://www.krug2000.ru)

**СОДЕРЖАНИЕ**

	Стр.
<b><u>ВВЕДЕНИЕ</u></b>	<b><u>1</u></b>
<b><u>1 ODBC-ДРАЙВЕР SCADA КРУГ-2000</u></b>	<b><u>1-1</u></b>
1.1 Назначение	1-1
1.2 Установка ODBC-драйвера	1-1
1.3 Настройка источников данных	1-1
1.4 Интерфейс ODBC-драйвера	1-5
1.5 Ограничения	1-23
1.6 Пример импорта данных в MS Access	1-23
1.7 Пример импорта данных в MS Excel	1-27
<b><u>2 SQL ГРАММАТИКА СИСТЕМЫ КРУГ-2000</u></b>	<b><u>2-1</u></b>
<b><u>3 СПИСОК АТТРИБУТОВ БД</u></b>	<b><u>3-1</u></b>



## **ВВЕДЕНИЕ**

В руководстве дается описание интерфейса драйвера ODBC SCADA КРУГ-2000. Руководство содержит рекомендуемые последовательности действий пользователя и практические советы по установке, настройке и использованию ODBC-драйвера SCADA КРУГ-2000.

ODBC (Open Database Connectivity) – это открытый интерфейс доступа к базам данных. Открытый интерфейс доступа к базам данных представляет собой библиотеку функций, которая позволяет прикладной программе обращаться к различным СУБД, используя структурированный язык запросов SQL.

Архитектура ODBC приведена на рисунке В.1.



Рисунок В.1

**ODBC - драйвер** – программный модуль, который применяет необходимые функции ODBC, позволяющие прикладной программе взаимодействовать с системой управления базами данных (СУБД). Драйверы являются компонентами, которые преобразуют запросы приложения в форму, понятную источникам данных, обрабатывают запросы ODBC и возвращают данные в приложение.

**Источник данных** - данные, к которым Пользователь хочет иметь доступ, и СУБД, которая управляет доступом к данным.

Используйте ODBC Administrator для того, чтобы добавить, сконфигурировать или удалить источник данных.

На рисунке В.1 показано как приложение взаимодействует с ODBC-драйвером:

- Приложение вызывает функции ODBC, чтобы передать SQL-операторы и извлечь результаты
- Менеджер драйверов загружает ODBC-драйвера от имени приложения
- Если необходимо, ODBC-драйвер модифицирует запрос приложения таким образом, чтобы он соответствовал синтаксису соответствующей СУБД, и обеспечивает обмен данными.

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Обычно, приложение, ODBC-драйвер и менеджер драйверов располагаются на одном компьютере, называемом клиентом. База данных и программное обеспечение для управления доступом к ней располагаются на другом компьютере, называемом сервером.

Пример доступа к БД SCADA «КРУГ-2000» через ODBC-драйвер (**KrugODBC.DLL**) при использовании технологии клиент/сервер приведен на рисунке В.2.

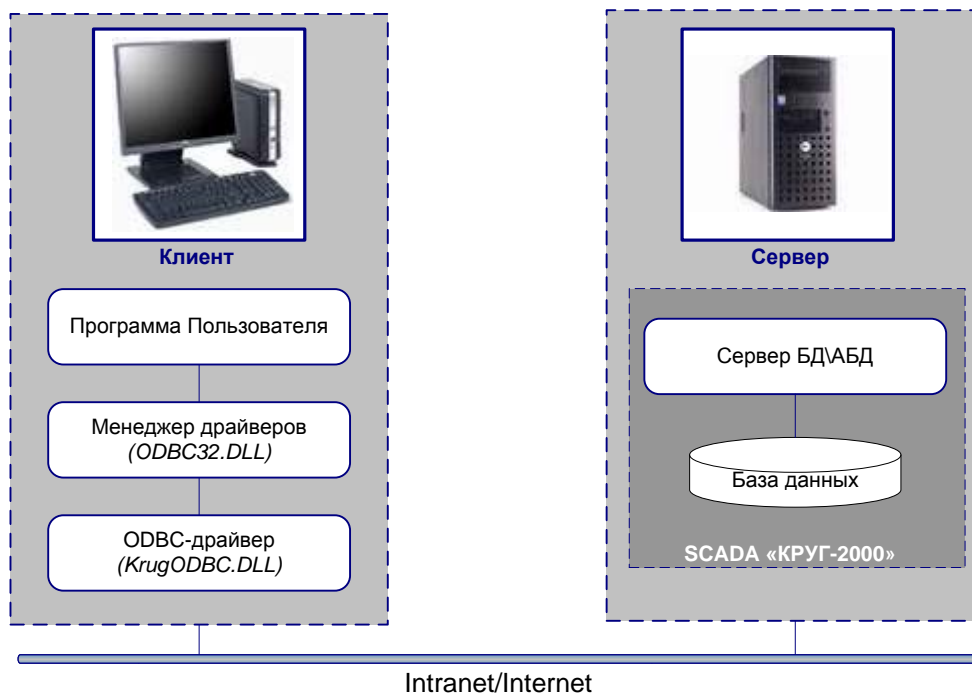


Рисунок В.2 - Доступ к БД через ODBC-драйвер при использовании технологии клиент/сервер

## 1 ODBC-ДРАЙВЕР SCADA КРУГ-2000

### 1.1 Назначение

Открытый интерфейс доступа к базам данных (ODBC) представляет собой библиотеку функций, которая позволяет прикладной программе обращаться к БД реального времени SCADA КРУГ-2000, используя структурированный язык запросов SQL.

### 1.2 Инсталляция ODBC-драйвера



#### **ВНИМАНИЕ!!!**

Для Windows 7 (Business, Enterprise или Ultimate) x86/x64 и Windows Server 2008 R2 запуск инсталлятора ODBC-драйвера (файла *Setup.exe*) следует производить от имени Администратора.

Для установки ODBC-драйвера запустите **<CD\_drive>:\ODBC-драйвер\setup.exe**.

Установите ODBC-драйвер, следуя инструкциям инсталлятора.

### 1.3 Настройка источников данных

При успешной инсталляции драйвера ODBC желательно сразу запустить модуль установки параметров серверов БД для ODBC – **ПУСК\Программы\Система КРУГ-2000\Настройка серверов для ODBC-драйвера**. Пользователь драйвера должен иметь возможность на одном клиентском компьютере соединяться и получать данные с нескольких серверов БД. Модуль установки параметров регистрирует в системном реестре параметры серверов, необходимые для соединения с ними ODBC-драйвера.



#### **Внимание!!!**

Если ODBC-драйвер инсталлируется на компьютере, где уже установлена SCADA КРУГ-2000, то при запуске модуля установки параметров в списке серверов уже могут присутствовать сервера нужные для работы SCADA КРУГ-2000 (OBD, NetMan, IBD, KBD). Убедительная просьба не удалять эти сервера и не изменять их параметры, это может привести к нестабильной работе SCADA КРУГ-2000.

Если ODBC-драйвер инсталлируется на компьютере, где не установлена SCADA КРУГ-2000, то этих серверов в списке не будет.

Интерфейс модуля установки параметров приведен на рисунке 1.3.1.

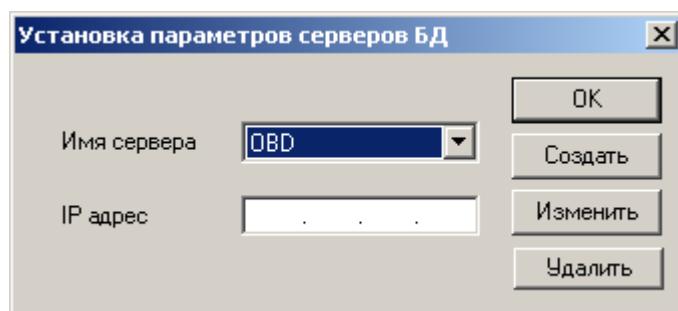



Рисунок 1.3.1

В выпадающем списке **Имя сервера** доступны серверы, уже зарегистрированные в реестре.

Для изменения параметров нажмите кнопку . Окно изменения параметров приведено на рисунке 1.3.2.

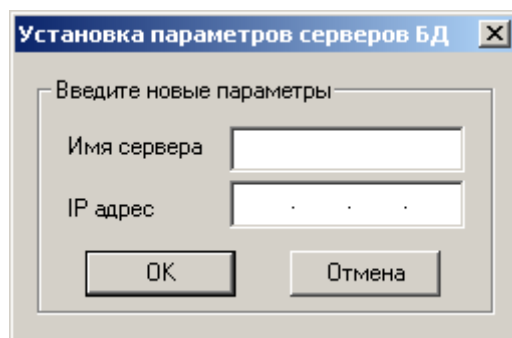
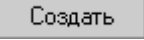


Рисунок 1.3.2

Для регистрации нового сервера нажмите кнопку . Окно регистрации сервера приведено на рисунке 1.3.3.

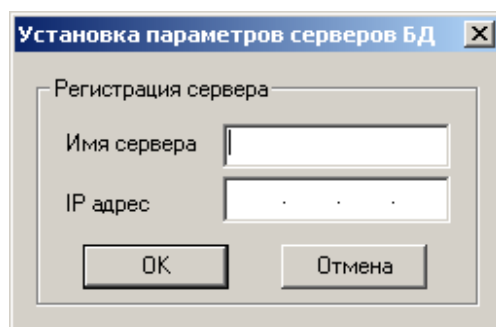
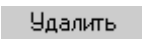


Рисунок 1.3.3

При регистрации нового сервера в поля ввода **Имя сервера** и **IP-адрес** вводятся любое пользовательское имя удаленного сервера и его IP адрес соответственно.

Для удаления сервера нажмите кнопку . Окно подтверждения удаления – рисунок 1.3.4.



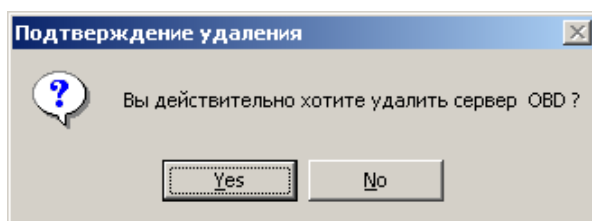
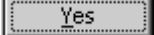


Рисунок 1.3.4

При нажатии на кнопку  (Да) сервер будет удален из реестра.



**ВНИМАНИЕ!!!**

Для ОС Windows 7 x64 и Server 2008 R2 x64 для запуска «Администратора источников данных ODBC» необходимо запускать файл «%WINDIR%\syswow64\odbcad32.exe», где «%WINDIR%» – системная директория операционной системы.

После создания списка серверов необходимо запустить администратор источников данных (рисунок 1.3.5): **ПУСК\Панель управления\Источники данных ODBC**

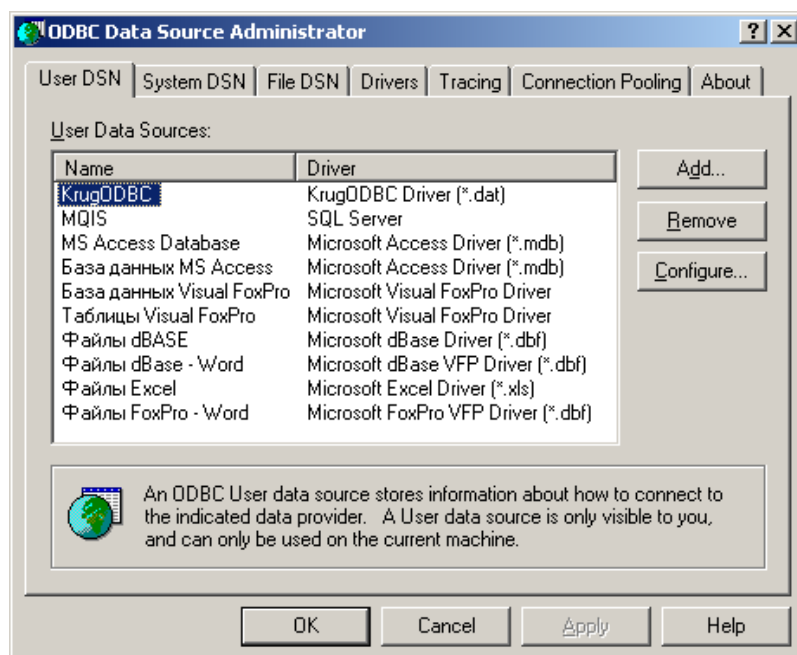



Рисунок 1.3.5

После инсталляции ODBC-драйвера в пользовательских источниках данных появляется источник данных **KrugODBC** (имя источника устанавливается по умолчанию).

Параметры источника данных можно изменить или создать новый источник данных.

Для добавления нового источника данных нажмите кнопку  (Добавить). На экране появится следующее окно (рисунок 1.3.6):

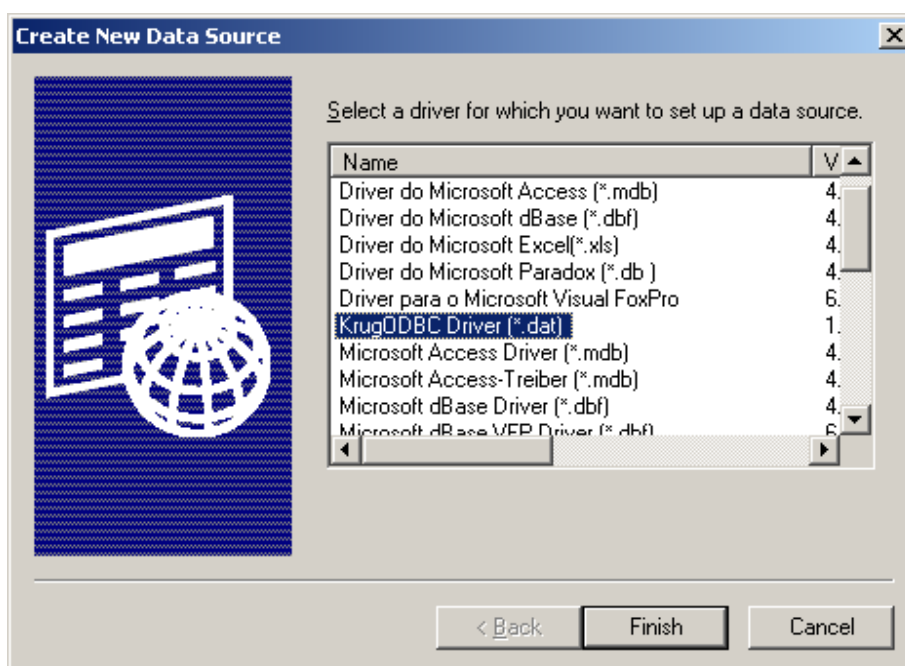


Рисунок 1.3.6

Выберите **KrugODBC Driver** и нажмите кнопку **Finish**. На экране появится окно настроек источника данных (рисунок 1.3.7):

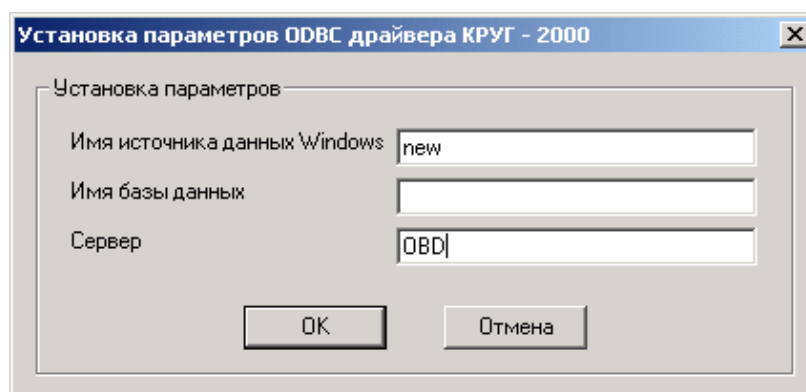


Рисунок 1.3.7

На рисунке 1.3.7:

- **Имя источника данных** – пользовательское имя, которое в дальнейшем будет использоваться в клиентских приложениях для доступа к источнику данных
- **Имя базы данных** – необязательный параметр
- **Сервер** – имя ранее созданного сервера.

Если ODBC-драйвер настраивается на компьютере, где уже установлена SCADA КРУГ-2000, то для доступа к локальному серверу используйте имя сервера **OBD**.

## 1.4 Интерфейс ODBC-драйвера

**SQLAllocEnv** - выделяет область хранения в памяти для идентификатора окружения

Синтаксис: **SQLRETURN SQLAllocEnv(HENV \*FAR phenv)**

Параметры:

Тип	Аргумент	Использование	Описание
HENV *FAR	phenv	Выход	Указатель на идентификатор окружения прикладной программы

**SQLAllocConnect** - выделяет область хранения в памяти для идентификатора соединения в рамках окружения

Синтаксис: **SQLRETURN SQLAllocConnect(HENV henv, HDBC \*FAR phdbc)**

Параметры:

Тип	Аргумент	Использование	Описание
HENV	Henv	Вход	Указатель на идентификатор окружения прикладной программы
HDBC *FAR	Phdbc	Выход	Указатель области хранения в памяти для идентификатора соединения

**SQLAllocStmt** - выделяет область хранения в памяти для идентификатора оператора и связывает этот идентификатор с соединением, которое определяется идентификатором соединения

Синтаксис: **SQLRETURN SQLAllocStmt(HDBC hdbc, HSTMT \*FAR phstmt)**

Параметры:

Тип	Аргумент	Использование	Описание
HDBC	Hdbc	Вход	Идентификатор соединения
HSTMT *FAR	Phstmt	Выход	Указатель области хранения в памяти для идентификатора оператора

**SQLFreeEnv** - освобождает ресурсы связанные с идентификатором окружения

Синтаксис: **SQLRETURN SQLFreeEnv(HENV henv)**

Параметры:

Тип	Аргумент	Использование	Описание
HENV	henv	Вход	Указатель на идентификатор окружения прикладной программы

**SQLFreeConnect** - освобождает ресурсы связанные с идентификатором соединения

Синтаксис: **SQLRETURN SQLFreeConnect(HDBC hdbc)**

Параметры:

Тип	Аргумент	Использование	Описание
HDBC	hdbc	Вход	Указатель на идентификатор соединения прикладной программы

**QLFreeStmt** -

а) останавливает любые SQL-операторы, которые в данный момент обрабатываются и связаны с заданным идентификатором оператора

б) закрывает любые открытые курсоры, которые имеют отношение к определенному идентификатору оператора

в) отбрасывает ожидаемые результаты

г) освобождает все ресурсы, связанные с определенным идентификатором оператора

Синтаксис:

**SQLRETURN SQLFreeStmt(SQLHSTMT StatementHandle, SQLUSMALLINT Option)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLUSMALLINT	Option	Вход	Одна из следующих опций: SQL_CLOSE – закрывает курсор, связанный с StatementHandle(если он был определен) и отбрасывает все ожидаемые результаты. SQL_DROP – освобождает StatementHandle, освобождает все ресурсы, связанные с ним, закрывает курсор, если он открыт, и отбрасывает все ожидаемые строки. Эта опция завершает все обращения к StatementHandle. StatementHandle обязательно должен быть переназначен для повторного использования. Опция SQL_DROP освобождает все ресурсы< которые были определены с помощью функции SQLAllocStmt. SQL_UNBIND – освобождает все буферы столбцов, которые повторно используются функцией SQLBindCol для данного идентификатора оператора.

**SQLConnect** - загружает драйвер базы данных и устанавливает соединение с источником данных

Синтаксис:

**SQLRETURN SQLConnect(SQLHDBC ConnectionHandle, SQLCHAR \*ServerName, SQLSMALLINT NameLength1, SQLCHAR \*UserName, SQLSMALLINT NameLength2, SQLCHAR \*Authentication, SQLSMALLINT NameLength3)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLCHAR *	ServerName	Вход	Имя источника данных
SQLSMALLINT	NameLength1	Вход	Длина ServerName. Это значение может быть равно SQL_NTS, если строка имеет нулевое окончание.
SQLCHAR *	UserName	Вход	Идентификатор пользователя
SQLSMALLINT	NameLength2	Вход	Длина UserName. Это значение может быть равно SQL_NTS, если строка имеет нулевое окончание.
SQLCHAR *	Authentication	Вход	Строка достоверности, обычно пароль
SQLSMALLINT	NameLength3	Вход	Длина Authentication. Это значение может быть равно SQL_NTS, если строка имеет нулевое окончание.

**SQLDriverConnect** - обеспечивает альтернативный метод соединения, отличающийся от того который предлагает функция SQLConnect

Синтаксис:

**SQLRETURN SQLDriverConnect(SQLHDBC ConnectionHandle, SQLHWND WindowHandle, SQLCHAR \*InConnectionString, SQLSMALLINT StringLength1, SQLCHAR \*OutConnectionString, SQLSMALLINT BufferLength, SQLSMALLINT \*StringLength2Ptr, SQLUSMALLINT DriverCompletion)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLHWND	WindowHandle	Вход	Идентификатор окна. Прикладная программа может передать идентификатор родительского окна или нулевой указатель в том случае, идентификатор окна не используется или SQLDriverConnect не представляет никаких диалогов.
SQLCHAR *	InConnectionString	Вход	Строка полного соединения, строка частичного соединения, или пустая строка
SQLSMALLINT	StringLength1	Вход	Длина InConnectionString
SQLCHAR *	OutConnectionString	Выход	Указатель области хранения строки законченного соединения. В момент успешного соединения с выбранным источником данных этот буфер содержит строку завершеного соединения. Прикладная программа должна выделить для этого буфера по крайней мере 1024 байта.
SQLSMALLINT	BufferLength	Вход	Максимальная длина буфера OutConnectionString
SQLSMALLINT *	StringLength2Ptr	Выход	Указатель общего числа байт (за исключением байта нулевого окончания), которые могут быть возвращены в OutConnectionString. Если это число больше или равно BufferLength, строка завершеного соединения в OutConnectionString усекается до числа байт, равного BufferLength-1 байт.
SQLUSMALLINT	DriverCompletion	Вход	SQL_DRIVER_PROMPT SQL_DRIVER_COMPLETE SQL_DRIVER_COMPLETE_REQUIRED или SQL_DRIVER_NOPROMPT

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

**SQLDisconnect** - закрывает соединение с источником данных

Синтаксис: **SQLRETURN SQLDisconnect(SQLHDBC ConnectionHandle)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения

**SQLGetInfo** - возвращает общую информацию о драйвере и источнике данных

Синтаксис:

**SQLRETURN SQLGetInfo(SQLHDBC ConnectionHandle, SQLUSMALLINT InfoType, SQLPOINTER InfoValuePtr, SQLSMALLINT BufferLength, SQLSMALLINT \*StringLengthPtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLUSMALLINT	InfoType	Вход	Тип информации. InfoType должен быть значением, представляющим тип информации, которую вы хотите выбрать.
SQLPOINTER	InfoValuePtr	Выход	Указатель области хранения информации. В зависимости от запрашиваемого InfoType, возвращаемая информация будет одной из нижеследующих: строка символов с NULL-окончанием, 16-битное (SQLUSMALLINT) целое значение, 32-битный (SQLINTEGER) флаг или 32-битное двоичное значение. Если аргумент InfoType равен SQL_DRIVER_HDESC или SQL_DRIVER_HSTMT, то InfoValuePtr используется и как входящий и как выходящий параметр.
SQLSMALLINT	BufferLength	Вход	Длина * InfoValuePtr. Если *InfoValuePtr не является строкой символов или InfoValuePtr равно NULL, то BufferLength игнорируется. Драйвер определяет размер *InfoValuePtr (SQLUSMALLINT или SQLINTEGER), на основе InfoType.
SQLSMALLINT *	StringLengthPtr	Выход	Указатель на буфер, в котором возвращается общее число байт (за исключением байта нулевого окончания), которые могут быть возвращены в InfoValuePtr. Для всех символьных данных кол-во возвращаемых байт больше или равно BufferLength, то информация в *InfoValuePtr усекается до числа байт, равного BufferLength-1 байт. При этом *InfoValuePtr имеет байт

Тип	Аргумент	Использование	Описание
			нулевого окончания, который вставляется драйвером. Для всех других типов данных значение BufferLength игнорируется и драйвер принимает размер *InfoValuePtr равным SQLUSMALLINT ИЛИ SQLINTEGER, в зависимости от InfoType.

**SQLGetFunctions** - возвращает информацию о том, поддерживает ли драйвер определенную функцию ODBC

Синтаксис:

**SQLRETURN SQLGetFunctions(SQLHDBC ConnectionHandle, SQLUSMALLINT FunctionId, SQLUSMALLINT \*SupportedPtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLUSMALLINT	FunctionId	Вход	Функция ODBC, информацию о которой необходимо получить.
SQLUSMALLINT *	SupportedPtr	Выход	TRUE, если определенная функция поддерживается драйвером, связанным с определенным идентификатором соединения (ConnectionHandle); в противном случае – FALSE.

**SQLNativeSql** - возвращает SQL-строку, которая была транслирована драйвером для специфического источника данных в синтаксис SQL, соответствующий этому источнику

Синтаксис:

**SQLRETURN SQLNativeSql(SQLHDBC ConnectionHandle, SQLCHAR \*InStatementText, SQLINTEGER TextLength1, SQLCHAR \*OutStatementText, SQLINTEGER BufferLength, SQLINTEGER \*TextLength2Ptr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLCHAR *	InStatementText	Вход	Текстовая SQL-строка для трансляции
SQLINTEGER	TextLength1	Вход	Длина текстовой строки *InStatementText
SQLCHAR *	OutStatementText	Выход	Указатель области хранения для транслированной SQL-строки
SQLINTEGER	BufferLength	Вход	Длина буфера *OutStatementText.
SQLINTEGER *	TextLength2Ptr	Выход	Указатель на буфер, в котором возвращается общее число байт (за исключением байта нулевого окончания), которые могут быть возвращены в *OutStatementText. Если кол-во возвращаемых байт больше или равно BufferLength, то транслированная SQL-строка в *OutStatementText усекается до числа байт, равного



## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Тип	Аргумент	Использование	Описание
			BufferLength-1 байт.

**SQLPrepare** - подготавливает оператор SQL к выполнению

Синтаксис:

**SQLRETURN SQLPrepare(SQLHSTMT StatementHandle, SQLCHAR \*StatementText, SQLINTEGER TextLength)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLCHAR *	StatementText	Вход	SQL-строка, которая должна быть выполнена после того, как она будет подготовлена.
SQLINTEGER	TextLength	Вход	Длина *StatementText.

**SQLExecute** - выполняет подготовленный оператор

Синтаксис: **SQLRETURN SQLExecute (SQLHSTMT StatementHandle)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора.

**SQLExecDirect** - реализует непосредственное выполнение SQL-оператора

Синтаксис:

**SQLRETURN SQLExecDirect(SQLHSTMT StatementHandle, SQLCHAR \*StatementText, SQLINTEGER TextLength)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора.
SQLCHAR *	StatementText	Вход	SQL-оператор для выполнения.
SQLINTEGER	TextLength	Вход	Длина StatementText

**SQLNumResultCols** - возвращает число столбцов, которые были получены с помощью подготовленного или выполненного оператора

Синтаксис:

**SQLRETURN SQLNumResultCols(SQLHSTMT StatementHandle, SQLSMALLINT \*ColumnCountPtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLSMALLINT *	ColumnCountPtr	Выход	Указатель на буфер, в который возвращается число столбцов в результирующем множестве.

**SQLRowCount** - возвращает число строк, на которые воздействовал SQL-оператор

Синтаксис:

**SQLRETURN SQLRowCount(SQLHSTMT StatementHandle, SQLINTEGER \*RowCountPtr)**

Параметры:

Тип	Аргумент	Использование	Описание
-----	----------	---------------	----------



**SQLDescribeCol** - возвращает дескриптор результата для одного столбца результирующего множества (дескриптор включает: имя столбца, тип, точность, размер и возможность принимать значения NULL)

SQLRETURN	SQLDescribeCol(SQLHSTMT StatementHandle, ColumnNumber, SQLCHAR *ColumnName, SQLSMALLINT BufferLength, SQLSMALLINT *DataLengthPtr, SQLSMALLINT *ColumnSizePtr, SQLSMALLINT *DecimalDigitsPtr, SQLSMALLINT *NullablePtr)	SQLSMALLINT *ColumnIndex, SQLSMALLINT *DataLengthPtr, SQLSMALLINT *DataOffsetPtr,
-----------	--	--

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLUSMALLINT	ColumnNumber	Вход	Номер столбца данных результата. Столбцы упорядочены слева направо, начиная с 1.
SQLCHAR *	ColumnName	Выход	Указатель области хранения имени столбца. Если столбец не имеет имени или имя столбца не может быть определено, то драйвер возвращает пустую строку.
SQLSMALLINT	BufferLength	Вход	Максимальная длина буфера ColumnName.
SQLSMALLINT *	NameLengthPtr	Выход	Общее число байт (исключая байт нулевого окончания), которое может быть возвращено в ColumnName. Если это число больше или равно BufferLength, имя источника данных в ColumnName усекается до BufferLength-1 байт.
SQLSMALLINT *	DataTypePtr	Выход	SQL-тип данных столбца. Может принимать одно из следующих значений: SQL_BIGINT SQL_BINARY SQL_BIT SQL_CHAR SQL_DATE SQL_DECIMAL SQL_DOUBLE SQL_FLOAT SQL_INTEGER SQL_LONGVARBINARY SQL_LONGVARCHAR SQL_NUMERIC SQL_REAL SQL_SMALLINT

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Тип	Аргумент	Использование	Описание
			SQL_TIME SQL_TIMESTAMP SQL_TINYINT SQL_VARBINARY SQL_VARCHAR или специфичный для драйвера SQL-тип данных. Если тип данных не может быть определен, то драйвер возвращает ноль.
UDWORD	DecimalDigitsPtr	Выход	Точность столбца в источнике данных. Если точность не может быть определен, то драйвер возвращает ноль.
SWORD	ColumnSizePtr	Выход	Размер столбца источника данных. Если размер не может быть определен или не используется, драйвер возвращает ноль.
SDWORD *FAR	NullablePtr	Выход	Указывает, допускаются ли в столбце NULL-значения. SQL_NO_NULLS – столбец не допускает NULL-значения. SQL_NULLABLE – столбец допускает NULL-значения. SQL_NULLABLE_UNKNOWN – драйвер не может определить, допускает ли столбец NULL-значения.

**SQLBindCol** - назначает область хранения в памяти и тип данных для столбца результирующего множества

Синтаксис:

**SQLRETURN SQLBindCol(SQLHSTMT StatementHandle, SQLUSMALLINT ColumnNumber, SQLSMALLINT TargetType, SQLPOINTER TargetValuePtr, SQLINTEGER BufferLength, SQLINTEGER \*StrLen\_or\_IndPtr);**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLUSMALLINT	ColumnNumber	Вход	Номер столбца результирующего множества, упорядоченный слева направо, начиная с 1. Столбец с номером 0 используется для выборки идентификатора записи.
SQLSMALLINT	TargetType	Вход	С-тип данных столбца результирующего множества. Может принимать одно из следующих значений: SQL_C_BINARY SQL_C_BOOKMARK SQL_C_CHAR SQL_C_DATE SQL_C_DEFAULT SQL_C_FLOAT SQL_C_SLONG SQL_C_SSHORT SQL_C_STINYINT

Тип	Аргумент	Использование	Описание
			SQL_C_TIME SQL_C_TIMESTAMP SQL_C_ULONG SQL_C_USHORT SQL_C_UTINYINT SQL_C_DEFAULT определяет, что данные должны быть переданы в его C-тип данных по умолчанию
SQLPOINTER	TargetValuePtr	Вход	Указатель области хранения данных. Если TargetValuePtr является нулевым указателем, то драйвер “отвязывает” столбец. Для отвязывания всех столбцов прикладная программа вызывает SQLFreeStmt с опцией SQL_UNBIND.
SQLINTEGER	BufferLength	Вход	Максимальная длина буфера TargetValuePtr. Для символьных данных TargetValuePtr должен также включать в себя место для байта нулевого окончания.
SQLINTEGER *	StrLen_or_IndPtr	Вход	SQL_NULL_DATA или число байт (за исключением байта нулевого окончания для символьных данных), которое может возвращаться в TargetValuePtr при вызове SQLExtendedFetch или SQLFetch, или SQL_NO_TOTAL, если число возможных байт не может быть определено. Если для символьных данных число байт, которое может быть возвращено равно SQL_NO_TOTAL или больше или равно BufferLength, то данные в TargetValuePtr усекаются до числа байт, равного BufferLength-1, и имеют байт нулевого окончания благодаря драйверу. Если для двоичных данных число байт, которое может быть возвращено равно SQL_NO_TOTAL или больше чем BufferLength, то данные в TargetValuePtr усекаются до числа байт, равного BufferLength. Для всех других типов данных значение BufferLength игнорируется и драйвер считает, что размер TargetValuePtr равен размеру C-типа данных, определенного в TargetType.

**SQLFetch** - извлекает строку данных из результирующего множества для SQL-операторов. Драйвер возвращает данные для всех столбцов, которые были связаны для предварительного хранения данных с помощью функции SQLBindCol.

Синтаксис: **SQLRETURN SQLFetch(SQLHSTMT StatementHandle)**

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора.

**SQLGetData** - выполняет выборку данных из столбцов, которые не были связаны с помощью SQLBindCol

Синтаксис:

**SQLRETURN SQLGetData(SQLHSTMT StatementHandle, SQLUSMALLINT ColumnNumber, SQLSMALLINT TargetType, SQLPOINTER TargetValuePtr, SQLINTEGER BufferLength, SQLINTEGER \*StrLen\_or\_IndPtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLUSMALLINT	ColumnNumber	Вход	Номер столбца результирующего множества, упорядоченный слева направо, начиная с 1. Столбец с номером 0 используется для выборки идентификатора записи для строки. Идентификатор записи не поддерживается драйверами ODBC 1.0 и функцией SQLFetch.
SQLSMALLINT	TargetType	Вход	С-тип данных столбца результирующего множества. Может принимать одно из следующих значений: SQL_C_BINARY SQL_C_BIT SQL_C_BOOKMARK SQL_C_CHAR SQL_C_DATE SQL_C_DEFAULT SQL_C_FLOAT SQL_C_SLONG SQL_C_SSHORT SQL_C_STINYINT SQL_C_TIME SQL_C_TIMESTAMP SQL_C_ULONG SQL_C_USHORT SQL_C_UTINYINT SQL_C_DEFAULT определяет, что данные должны быть переданы в его С-тип данных по умолчанию
SQLPOINTER	TargetValuePtr	Выход	Указатель области хранения данных.
SQLINTEGER	BufferLength	Вход	Максимальная длина буфера TargetValuePtr. Для символьных данных TargetValuePtr должен также включать в себя место для байта нулевого окончания. Для символьных и двоичных С-данных, BufferLength

Тип	Аргумент	Использование	Описание
			определяет кол-во данных, которое может быть получено при единичном вызове SQLGetData. Для всех других типов C-данных, BufferLength игнорируется. Драйвер полагает, что размер TargetValuePtr равен размеру типа данных определенного с помощью TargetType и возвращает полное значение данных.
SQLINTEGER *	StrLen_or_IndPtr	Вход	SQL_NULL_DATA или общее число байт (за исключением байта нулевого окончания для символьных данных), которое может возвращаться в TargetValuePtr для текущего вызова SQLGetData или SQL_NO_TOTAL, если это число возможных байт не может быть определено. Если для символьных данных число байт, которое может быть возвращено равно SQL_NO_TOTAL или больше или равно BufferLength, то данные в TargetValuePtr усекаются до числа байт, равного BufferLength-1, и имеют байт нулевого окончания, который вставляется драйвером. Если для двоичных данных *StrLen_or_IndPtr равно SQL_NO_TOTAL или больше чем BufferLength, то данные в TargetValuePtr усекаются до числа байт, равного BufferLength. Для всех других типов данных значение BufferLength игнорируется и драйвер считает, что размер TargetValuePtr равен размеру C-типа данных, определенного в TargetType.

**SQLTables** - возвращает список имен таблиц, которые содержит указанный источник данных

Синтаксис:

**SQLRETURN SQLTables(SQLHSTMT StatementHandle, SQLCHAR \*CatalogName, SQLSMALLINT NameLength1, SQLCHAR \*SchemaName, SQLSMALLINT NameLength2, SQLCHAR \*TableName, SQLSMALLINT NameLength3, SQLCHAR \*TableType, SQLSMALLINT NameLength4)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора для выбираемых результатов
SQLCHAR *	CatalogName	Вход	Имя квалификатора (каталога). Если драйвер поддерживает квалификаторы только для некоторых таблиц, а также при выборке драйвером данных из

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Тип	Аргумент	Использование	Описание
			различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют квалификаторов.
SQLSMALLINT	NameLength1	Вход	Длина *CatalogName
SQLCHAR *	SchemaName	Вход	Образец строки поиска для имен собственников.
SQLSMALLINT	NameLength2	Вход	Длина *SchemaName
SQLCHAR *	TableName	Вход	Образец строки поиска для имен таблиц. Если драйвер поддерживает собственников только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют собственников.
SQLSMALLINT	NameLength3	Вход	Длина *TableName
SQLCHAR *	TableType	Вход	Список типов таблиц для выбора.
SQLSMALLINT	NameLength4	Вход	Длина *TableType

**SQLColumns** - возвращает список имен столбцов заданной таблицы

Синтаксис:

**SQLRETURN SQLColumns(SQLHSTMT StatementHandle, SQLCHAR \*CatalogName, SQLSMALLINT NameLength1, SQLCHAR \*SchemaName, SQLSMALLINT NameLength2, SQLCHAR \*TableName, SQLSMALLINT NameLength3, SQLCHAR \*ColumnName, SQLSMALLINT NameLength4)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLCHAR *	CatalogName	Вход	Имя квалификатора (каталога). Если драйвер поддерживает квалификаторы только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют квалификаторов.
SQLSMALLINT	NameLength1	Вход	Длина CatalogName
SQLCHAR *	SchemaName	Вход	Образец строки поиска для имен собственников. Если драйвер поддерживает собственников только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют собственников.
SQLSMALLINT	NameLength2	Вход	Длина SchemaName
SQLCHAR *	TableName	Вход	Образец строки поиска для имен таблиц
SQLSMALLINT	NameLength3	Вход	Длина TableName
SQLCHAR *	ColumnName	Вход	Образец строки поиска для имен столбцов
SQLSMALLINT	NameLength4	Вход	Длина ColumnName

**SQLStatistics** - выбирает список статистических данных об отдельной таблице и список индексов, которые связаны с этой таблицей

Синтаксис:

**SQLRETURN SQLStatistics(SQLHSTMT StatementHandle, SQLCHAR \*CatalogName, SQLSMALLINT NameLength1, SQLCHAR \*SchemaName, SQLSMALLINT NameLength2, SQLCHAR \*TableName, SQLSMALLINT NameLength3, SQLUSMALLINT Unique, SQLUSMALLINT Reserved)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLCHAR *	CatalogName	Вход	Имя квалификатора (каталога). Если драйвер поддерживает квалификаторы только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют квалификаторов.
SQLSMALLINT	NameLength1	Вход	Длина *CatalogName
SQLCHAR *	SchemaName	Вход	Образец строки поиска для имен собственников. Если драйвер поддерживает собственников только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("" ) означает, что эти таблицы не имеют собственников.
SQLSMALLINT	NameLength2	Вход	Длина *SchemaName
SQLCHAR *	TableName	Вход	Образец строки поиска для имен таблиц
SQLSMALLINT	NameLength3	Вход	Длина *TableName
SQLUSMALLINT	Unique	Вход	Тип индекса: SQL_INDEX_UNIQUE или SQL_INDEX_ALL.
SQLUSMALLINT	Reserved	Вход	Указывает значение кол-ва строк и столбцов в результирующем множестве. Следующие опции оказывают влияние только на возвращаемые строки и столбцы (информация об индексах возвращается даже если не возвращаются строки и столбцы): SQL_ENSURE требует безусловной выборки статистики драйвера. SQL_QUICK требует, чтобы драйвер выбрал результаты только в том случае, если они полностью готовы на сервере. В этом случае драйвер не гарантирует, что все значения являются текущими.



**SQLSpecialColumns** - выбирает следующую информацию о столбцах заданной таблицы:

- а) оптимальный набор столбцов, который однозначно идентифицирует строку в таблице
- б) столбцы, которые обновляются автоматически при изменении любого значения в строке

Синтаксис:

**SQLRETURN SQLSpecialColumns(SQLHSTMT StatementHandle,  
SQLSMALLINT IdentifierType,  
SQLCHAR \*CatalogName,  
SQLSMALLINT NameLength1,  
SQLCHAR \*SchemaName,  
SQLSMALLINT NameLength2, SQLCHAR \*TableName,  
SQLSMALLINT NameLength3, SQLSMALLINT Scope,  
SQLSMALLINT Nullable)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLSMALLINT	IdentifierType	Вход	Тип возвращаемого столбца. Может принимать одно из следующих значений: SQL_BEST_ROWID – возвращает оптимальный столбец или набор столбцов, которые при выборке значения из столбца или столбцов, позволяют однозначно определить строку в заданной таблице. Столбец может быть или псевдостолбцом, который был создан специально для этой цели (такой как ROWID в ORACLE или TID в INGRES) или столбцом любого уникального индекса этой таблицы. SQL_ROWVER – возвращает столбец или столбцы в заданной таблице, которые автоматически обновляются источником данных при изменении значения любого столбца (подобно ROWID из SQLBase или TIMESTAMP из Sybase).
SQLCHAR *	CatalogName	Вход	Имя квалификатора (каталога). Если драйвер поддерживает квалификаторы только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД, пустая строка ("") означает, что эти таблицы не имеют квалификаторов.
SQLSMALLINT	NameLength1	Вход	Длина *CatalogName
SQLCHAR *	SchemaName	Вход	Образец строки поиска для имен собственников. Если драйвер поддерживает собственников только для некоторых таблиц, а также при выборке драйвером данных из различных СУБД,



Тип	Аргумент	Использование	Описание
			пустая строка ("") означает, что эти таблицы не имеют собственников.
SQLSMALLINT	NameLength2	Вход	Длина *SchemaName
SQLCHAR *	TableName	Вход	Имя таблицы.
SQLSMALLINT	NameLength3	Вход	Длина *TableName
SQLSMALLINT	Scope	Вход	Минимально необходимая область действия rowid. Возвращаемый rowid может иметь большую область действия. Может принимать одно из следующих значений: SQL_SCOPE_CURROW – rowid будет действительным только на время позиционирования на текущей строке. Последующий повторный набор, использующий rowid, может не вернуть строку, если эта строка была обновлена или удалена другой транзакцией. SQL_SCOPE_TRANSACTION – rowid будет действительным в течении текущей транзакции. SQL_SCOPE_SESSION – rowid будет действительным в течении текущего сеанса.
SQLSMALLINT	Nullable	Вход	Определяет, возвращать ли специальные столбцы, которые могут иметь значение NULL. Может принимать одно из следующих значений: - SQL_NO_NULLS: исключает специальные столбцы, которые могут иметь значение NULL; - SQL_NULLABLE: возвращает специальные столбцы, даже, если они могут иметь значения NULL.

**SQLCancel** - прерывает действие или действия, связанные с предварительно назначенным идентификатором оператора, который в свою очередь был связан с источником данных через идентификатор соединения.

Синтаксис: **SQLRETURN SQLCancel(StatementHandle)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора

**SQLGetCursorName** - возвращает имя курсора, которое ассоциируется с определенным идентификатором оператора

Синтаксис:

**SQLRETURN SQLGetCursorName(SQLHSTMT StatementHandle, SQLCHAR \*CursorName, SQLSMALLINT BufferLength, SQLSMALLINT \*NameLengthPtr)**

Параметры:

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLCHAR *	CursorName	Выход	Указатель области хранения имени курсора.
SQLSMALLINT	BufferLength	Вход	Длина CursorName
SQLSMALLINT *	NameLengthPtr	Выход	Общее число байт (исключая байт нулевого окончания), которое может быть возвращено в CursorName. Если это число больше или равно BufferLength, то имя курсора в CursorName усекается до числа байт, равного BufferLength-1 байт.

**SQLSetCursorName** - присваивает имя курсору

Синтаксис:

**SQLRETURN SQLSetCursorName(SQLHSTMT StatementHandle, SQLCHAR \*CursorName, SQLSMALLINT NameLength)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора. StatementHandle не может соответствовать оператору, который находится в выполняемом или позиционируемом состоянии.
SQLCHAR *	CursorName	Вход	Имя курсора. Для эффективной обработки в имени курсора не должно быть ведущих или конечных пробелов.
SQLSMALLINT	NameLength	Вход	Длина *CursorName.

**SQLGetTypeInfo** - возвращает информацию о типах данных, которые поддерживаются источником данных. Драйвер возвращает информацию в виде результирующего множества.

Синтаксис:

**SQLRETURN SQLGetTypeInfo(SQLHSTMT StatementHandle, SQLSMALLINT DataType)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLSMALLINT	DataType	Вход	SQL-тип данных. При DataType равном SQL_ALL_TYPES, должна быть возвращена информация о всех типах данных.

**SQLGetConnectOption** - возвращает текущую установку атрибута соединения

Синтаксис:

**SQLRETURN SQLGetConnectOption(SQLHDBC ConnectionHandle, SQLINTEGER Attribute, SQLPOINTER ValuePtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLINTEGER	Attribute	Вход	Извлекаемый атрибут.
SQLPOINTER	ValuePtr	Выход	Указатель на область, в которую возвращается текущее значение

			атрибута Attribute
--	--	--	--------------------

**SQLSetConnectOption** - устанавливает атрибуты соединения

Синтаксис:

**SQLRETURN SQLSetConnectOption(SQLHDBC ConnectionHandle, SQLINTEGER Attribute, SQLINTEGER Value)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHDBC	ConnectionHandle	Вход	Идентификатор соединения
SQLINTEGER	Attribute	Вход	Устанавливаемый атрибут.
SQLINTEGER	Value	Вход	Значение атрибута.

**SQLGetStmtOption** - возвращает текущую установку атрибута оператора

Синтаксис:

**SQLRETURN SQLGetStmtOption(SQLHSTMT StatementHandle, SQLINTEGER Attribute, SQLPOINTER ValuePtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLINTEGER	Attribute	Вход	Извлекаемый атрибут.
SQLPOINTER	ValuePtr	Выход	Указатель на область, в которую возвращается текущее значение атрибута Attribute

**SQLSetStmtOption** - устанавливает атрибуты, связанные с оператором

Синтаксис:

**SQLRETURN SQLSetStmtOption(SQLHSTMT StatementHandle, SQLINTEGER Attribute, SQLINTEGER Value)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLINTEGER	Attribute	Вход	Устанавливаемый атрибут.
SQLINTEGER	Value	Вход	Значение атрибута.

**SQLColAttributes** - возвращает информацию о столбце результирующего множества

Синтаксис:

**SQLRETURN SQLColAttributes(StatementHandle, ColumnNumber, FieldIdentifier, CharacterAttributePtr, BufferLength, StringLengthPtr, NumericAttributePtr)**

Параметры:

Тип	Аргумент	Использование	Описание
SQLHSTMT	StatementHandle	Вход	Идентификатор оператора
SQLUSMALLINT	ColumnNumber	Вход	Номер столбца результирующего множества, упорядоченных слева направо, начиная с 1.
SQLUSMALLINT	FieldIdentifier	Вход	Определяет, какая информация должна быть возвращено из

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

Тип	Аргумент	Использование	Описание
			колонки ColumnNumber
SQLPOINTER	CharacterAttributePtr	Выход	Указатель на область памяти, куда возвращается информация, соответствующая FieldIdentifier
SQLSMALLINT	BufferLength	Вход	Максимальная длина буфера CharacterAttributePtr
SQLSMALLINT *	StringLengthPtr	Выход	Указатель на буфер, в котором возвращается общее число байт (за исключением байта нулевого окончания), которые могут быть возвращены в *CharacterAttributePtr. Для символьных данных, если кол-во возвращаемых байт больше или равно BufferLength, то *CharacterAttributePtr усекается до числа байт, равного BufferLength-1 байт. Для всех других типов данных BufferLength игнорируется.
SQLPOINTER	NumericAttributePtr	Выход	Указатель на целое, по которому возвращается значение FieldIdentifier столбца ColumnNumber, для типов данных таких как SQL_COLUMN_LENGTH, в противном случае поле не используется.

### 1.5 Ограничения

На работу ODBC-драйвера в реальном времени накладывается ряд ограничений, которые контролируются Сервером базы данных:

	Ограничение	Значение	Описание
1	Количество внешних SQL-клиентов	3	Максимальное количество компьютеров – SQL-клиентов – одновременно подключенных к Серверу базы данных
2	Размер буфера для размещения результата SQL-запроса	300000 байт	Если объем данных, возвращаемых в качестве результата SQL-запроса, превысит указанное значение, то SQL-обработчик отбросит «лишние» данные
3	Количество атрибутов в SQL-запросе	50	SQL-запрос не будет выполняться, если количество атрибутов таблиц в запросе будет больше указанного значения
4	Количество SQL-запросов в минуту	20	Если в минуту на сервер БД будет приходить больше SQL-запросов, чем указанное значение, то серверная часть ODBC в системном журнале событий в раздел «Приложения» будет помещать предупреждающее сообщение с идентификатором 12288: Источник <b>KrugODBC «Превышено допустимое количество SQL-запросов к серверу БД».</b> <b>В этом случае, Серверная часть ODBC будет возвращать ошибку ODBC-драйверу до истечения минуты. Т.е. запросы ODBC-драйвера на сервере БД обрабатываться не будут.</b>

### 1.6 Пример импорта данных в MS Access

Для осуществления импорта данных следует:

- настроить ODBC-драйвер (описание в разделе 1.3). В примере используется сервер с именем «**OBD**», созданный при настройке ODBC-драйвера
- запустить Сервер БД SCADA КРУГ-2000.

Порядок действий:

- 1 Открыть **MS Access** из системного меню «**Пуск/Все программы/Microsoft Office/Microsoft Office Access**» и выбрать пункт «**Новая база данных...**» (рисунок 1.6.1)
- 2 В появившемся после создания базы данных окне «**....: база данных (формат Access 2007)**» (рисунок 1.6.2) перейти на вкладку «**Внешние данные**» и в группе «**Импорт**» выбрать пункт «**Дополнительно/ База данных ODBC**»

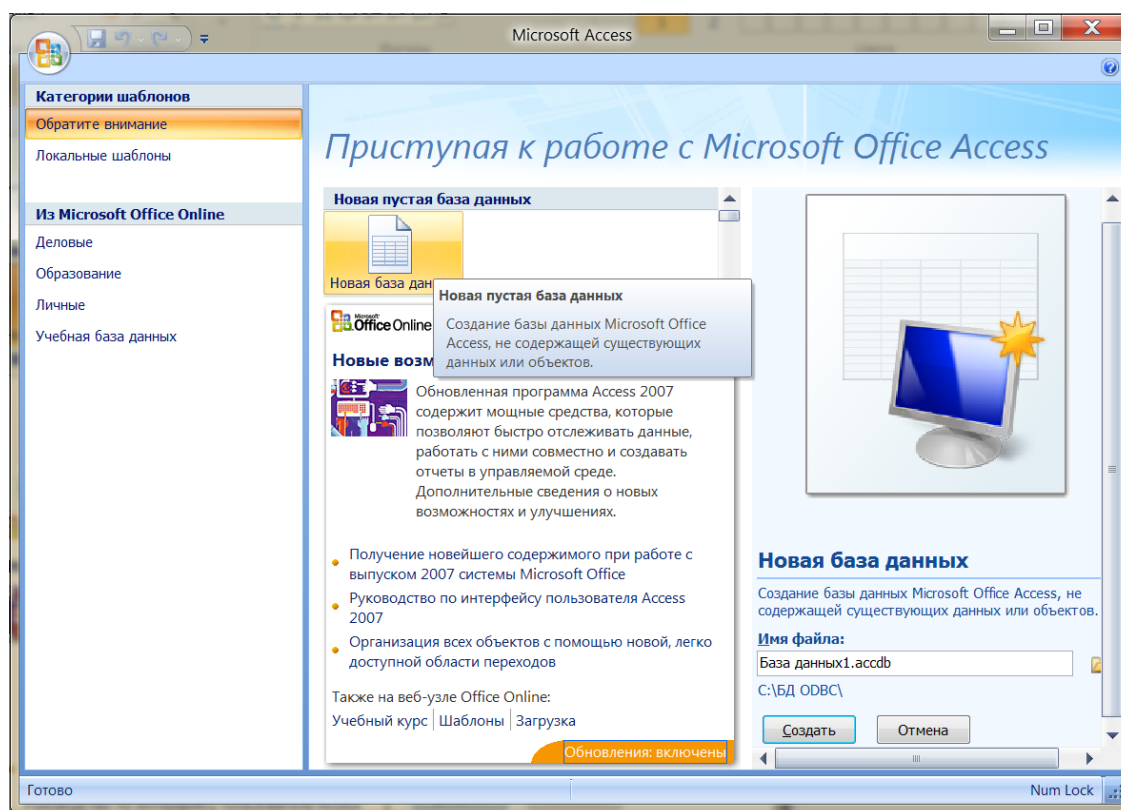


Рисунок 1.6.1 – Новая база данных

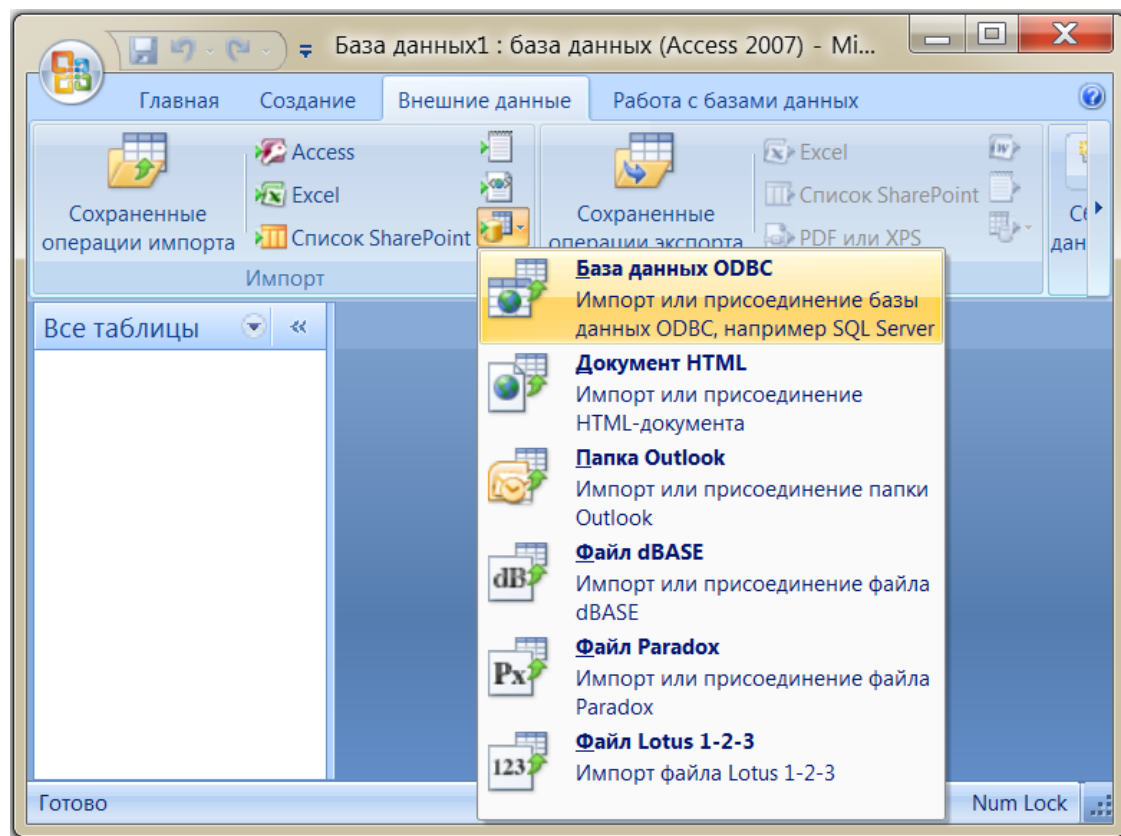


Рисунок 1.6.2 – Новая база данных

- 3 В открывшемся окне (рисунок 1.6.3) выбрать «Импортировать данные источника в новую таблицу в текущей базе данных» и нажать на кнопку «ОК»

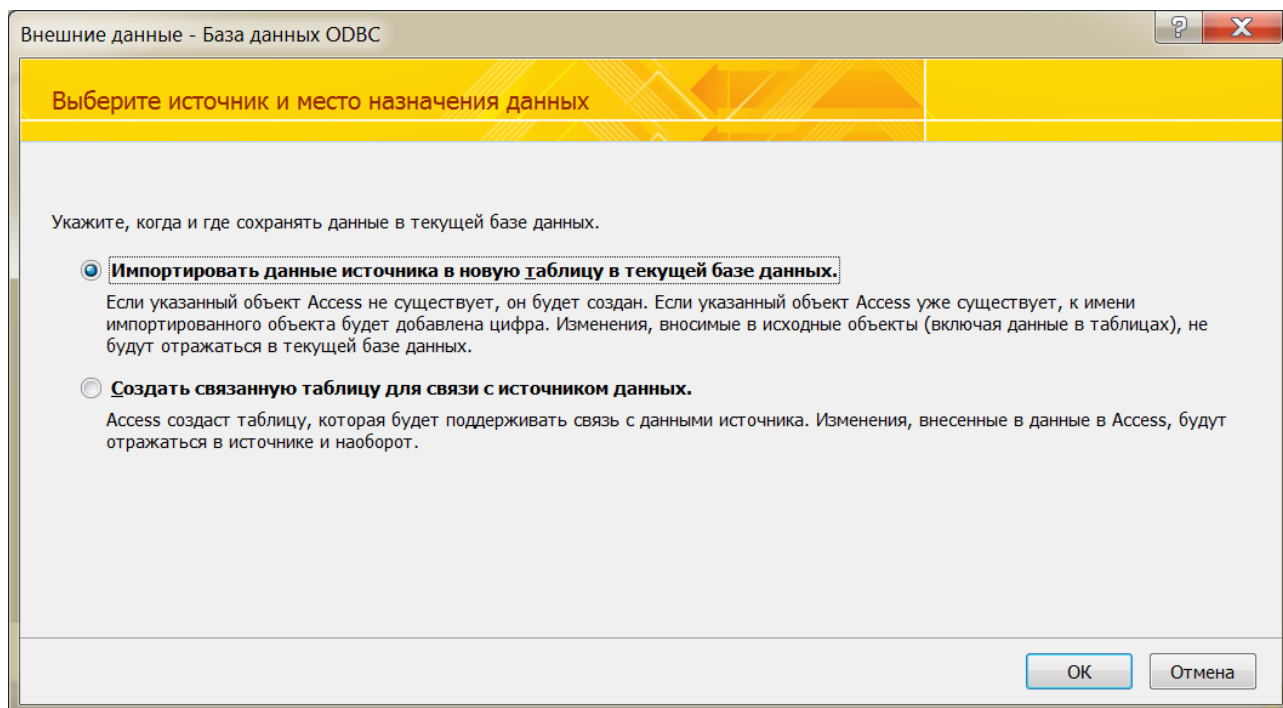


Рисунок 1.6.3 – Импорт данных

- 4 После этого открывается окно со списком источников данных, доступных на данный момент (рисунок 1.6.4). Выбрать в качестве источника данных «KrugODBC» и нажать на кнопку «ОК»

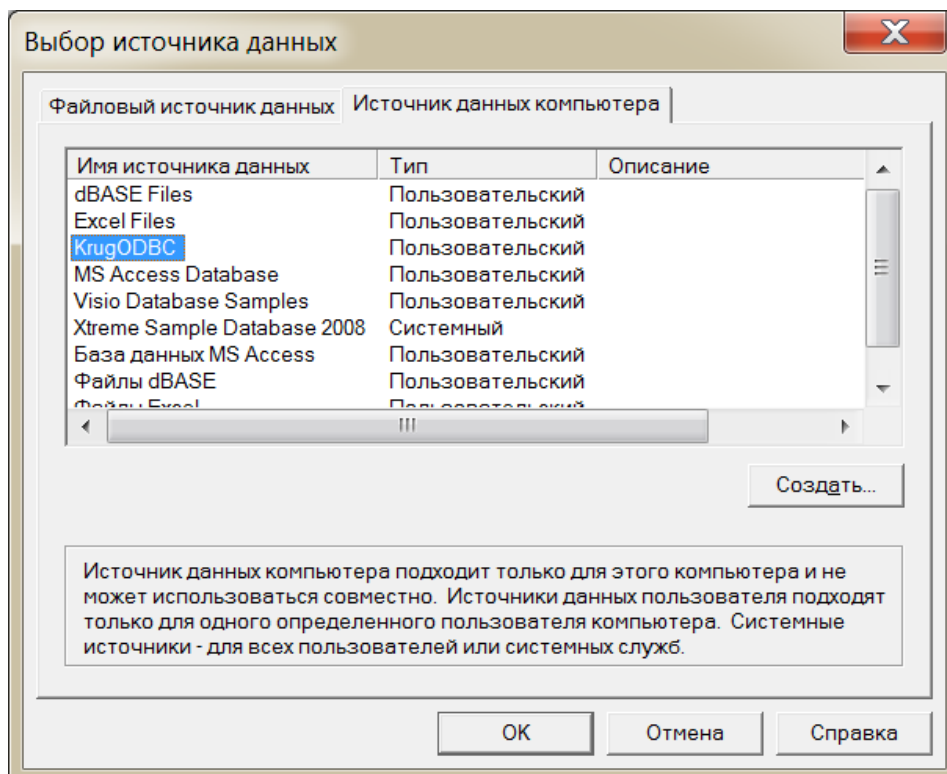


Рисунок 1.6.4 – Источник данных

- 5 В появившемся окне (рисунок 1.6.5) в поле ввода «Сервер» ввести имя сервера «OBD» и нажать на кнопку «ОК»

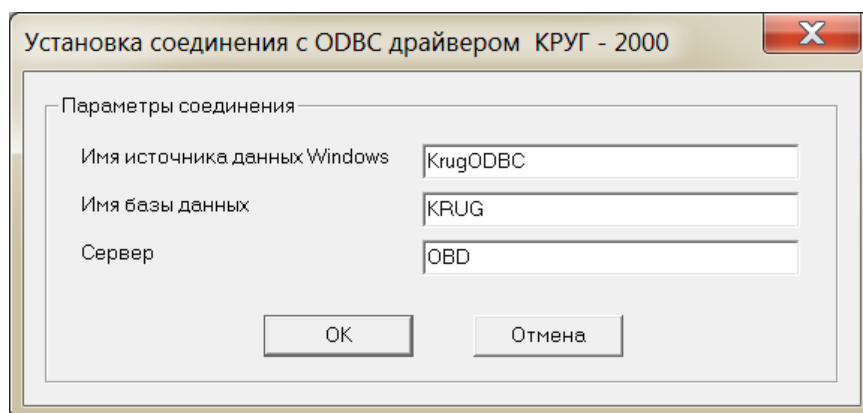


Рисунок 1.6.5 – Установка соединения

- 6 Выбрать в окне «Импорт объектов» (рисунок 1.6.6) таблицы, которые необходимо импортировать, и нажать на кнопку «ОК»

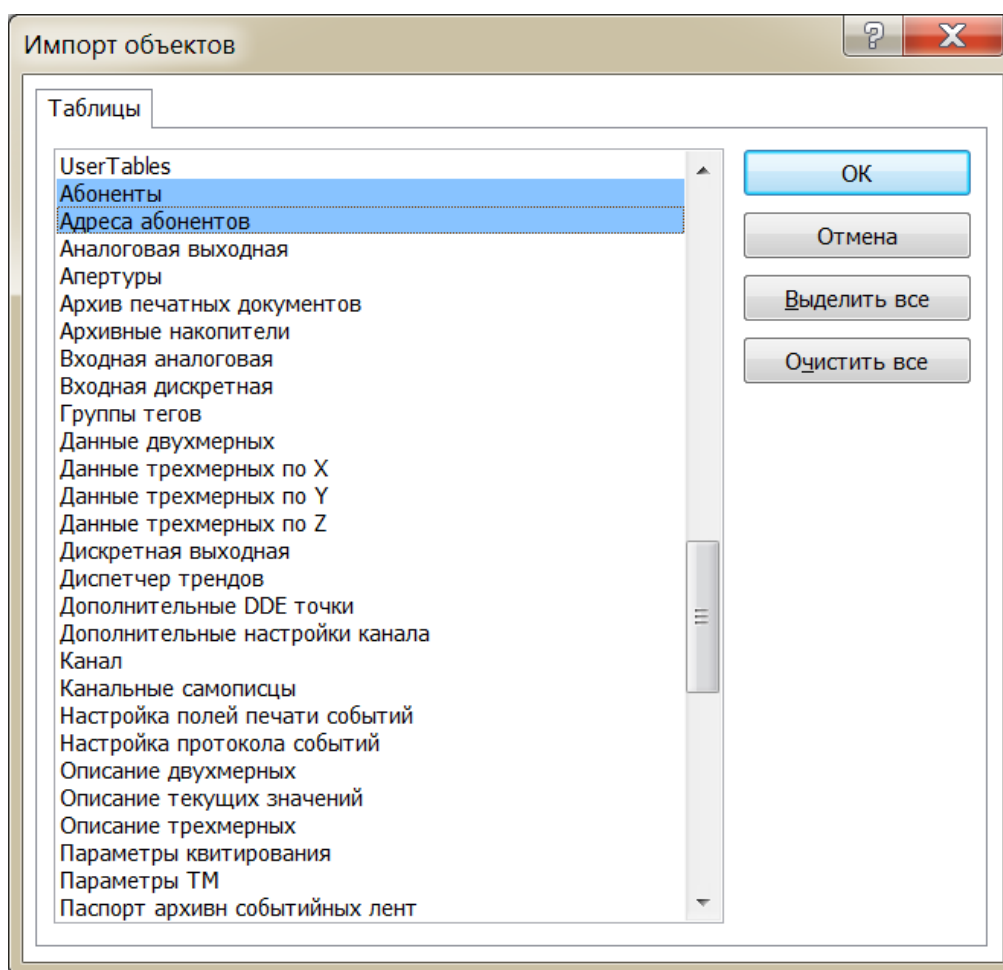


Рисунок 1.6.6 – Выбор таблиц для импорта



- 7 Выбранные таблицы добавлены в базу данных Access (рисунок 1.6.7) и могут быть использованы для обработки данных.

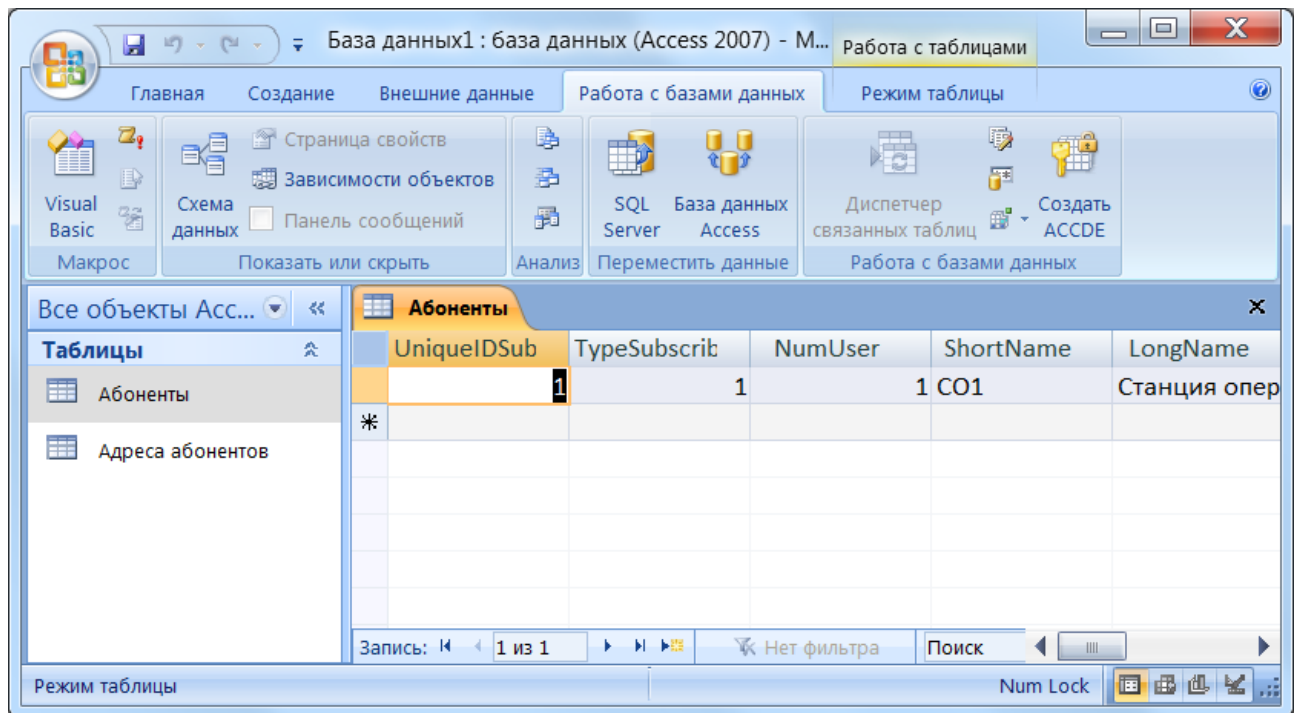


Рисунок 7– База данных после добавления таблиц

### 1.7 Пример импорта данных в MS Excel

Для импорта данных в Microsoft Excel необходимо использовать надстройку **Power Query**. Системные требования **Power Query** и принципы работы смотрите на официальном сайте Microsoft.

Для осуществления импорта данных следует:

- настроить ODBC-драйвер (описание в разделе 1.3),
- запустить Сервер БД SCADA КРУГ-2000.

Порядок действий:

- 1 Перейти в **PowerQuery**. (При установленном **PowerQuery** появляется соответствующий Пункт меню в Microsoft Excel).
- 2 Выбрать пункт меню «Из других источников».

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

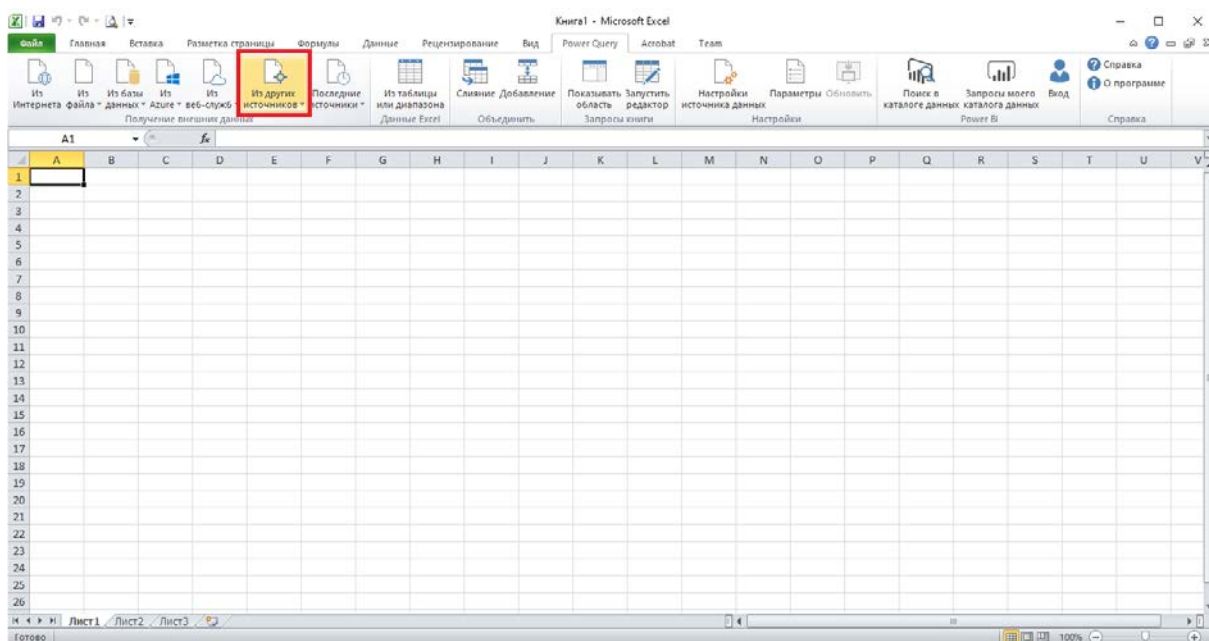


Рисунок 1.7.1 – Пункт меню PowerQuery в Microsoft Excel

- 3 Выбрать пункт «Из ODBC импорт данных из ODBC»
- 4 Выбрать источник данных «KrugODBC».

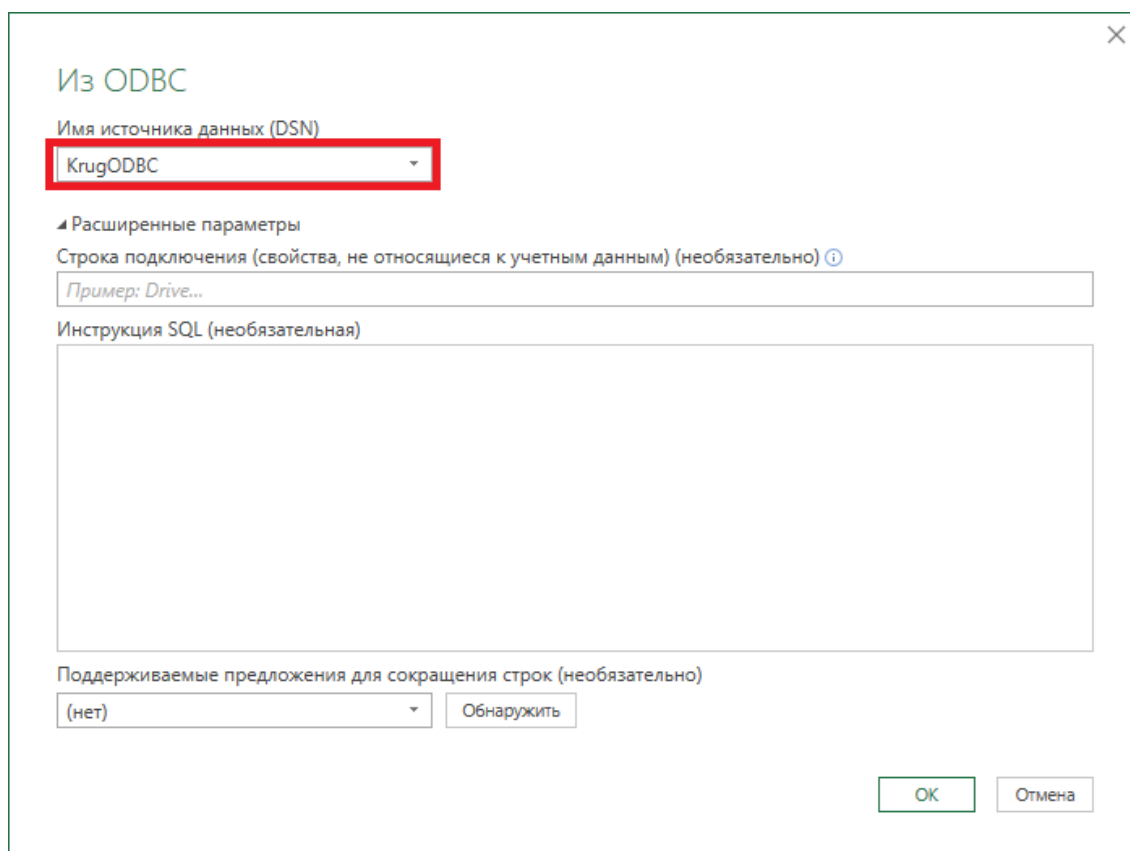


Рисунок 1.7.2 – Выбор источника данных

- 5 Раскрыть поле «расширенные параметры».

- 6 В поле «Инструкция SQL» написать SQL запрос. Нажать кнопку ОК.

Из ODBC

Имя источника данных (DSN)  
KrugODBC

▲ Расширенные параметры

Строка подключения (свойства, не относящиеся к учетным данным) (необязательно) ⓘ  
Пример: Drive...

Инструкция SQL (необязательная)  
SELECT \* FROM Абоненты

Поддерживаемые предложения для сокращения строк (необязательно)  
(нет)    Обнаружить

ОК    Отмена

Рисунок 1.7.3 –SQL-запрос

- 7 После получения результата SQL-запроса нажмите кнопку Загрузить.

ODBC (dsn=KrugODBC SELECT \* FROM Абоненты)

UniqueIDSubscriber	TypeSubscriber	NumUser	ShortName	LongName	NameComputer	bReservOn	bTransitOn	ReservedWord0
1	4	1	САБД1	Сервер АБД 1	tolstych	0	0	
2	7	1	ОПСЕРВ1	ОПС-сервер 1	localhost	0	0	
3	7	2	ОПСЕРВ2	ОПС-сервер 2	localhost	0	0	
4	4	2	САБД2	Сервер АБД 2	SABD2	0	0	

Загрузить Изменить Отмена

Рисунок 1.7.4 – Результат SQL-запроса

### 8 Результат

Книга1 - Microsoft Excel

Таблица: Таблица1

Запрос1

Загружено строк: 4.

UniqueIDSubscriber	TypeSubscriber	NumUser	ShortName	LongName	NameComputer	bReservOn	bTransitOn	ReservedWord0
1	4	1	САБД1	Сервер АБД 1	SABD1	0	0	
2	7	1	ОПСЕРВ1	ОПС-сервер 1	localhost	0	0	
3	7	2	ОПСЕРВ2	ОПС-сервер 2	localhost	0	0	
4	4	2	САБД2	Сервер АБД 2	SABD2	0	0	

Рисунок 1.7.5 - Импортированные в Microsoft Excel данные

При получении таблицы «Входная аналоговая» с помощью ODBC драйвера в MS Access может возникнуть ошибка выполнения SQL-запроса (рисунок 1.7.6).

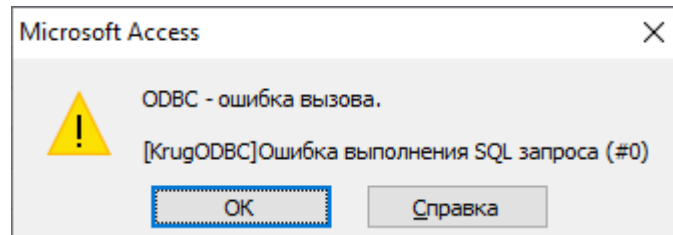


Рисунок 1.7.6 – Ошибка выполнения SQL-запроса

Данная ошибка вызвана нарушением ограничения количества столбцов в SQL-запросе (см. раздел 1.5 Ограничения). В этом случае необходимо уменьшить количество запрашиваемых столбцов. Один из вариантов решения проблемы:

- 1) В MS Access выбрать пункт меню «Внешние данные» -> «База данных ODBC» -> «Создать связанную таблицу» -> «Создать связанную таблицу для связи с источником данных».
- 2) В диалоговом окне «Выбор источника данных» перейти на вкладку «Источник данных компьютера».
- 3) Выбрать источник данных KrugODBC.
- 4) В диалоговом окне «Связь с таблицей» выбрать таблицу «Входная аналоговая».
- 5) В диалоговом окне «Выбор однозначного индекса» выбрать «Отмена».
- 6) В MS Access выбирать пункт меню «Создание» -> «Конструктор запросов».
- 7) В диалоговом окне «Добавление таблицы» выбрать таблицу «Входная аналоговая».
- 8) Выбрать необходимые атрибуты. (Например, Позиция, Имя 1, Имя 2, Единица измерения).
- 9) Сохранить запрос путем выбора соответствующего пункта контекстного меню на заголовке окна запроса.
- 10) На панели инструментов выбрать пункт «Выполнить».



## **2 SQL ГРАММАТИКА СИСТЕМЫ КРУГ-2000**

SELECT [TOP <целая константа> [PERCENT]] <список полей>  
FROM <список таблиц>  
[WHERE <условие выборки>]  
[ORDER BY <спецификация сортировки>]  
[<признак окончания запроса>]

<признак окончания запроса>::= ;

<список полей>::=

\*|<элемент списка полей> [, <элемент списка полей>]

<элемент списка полей>::=

<имя таблицы>.\* | <идентификатор столбца>[ AS <псевдоним столбца>]

<список таблиц>::=

<элемент списка таблиц> [, <элемент списка таблиц>]

<элемент списка таблиц>::=

<имя таблицы>[ AS <псевдоним таблицы>]

<идентификатор столбца>::=

<имя таблицы>.<имя столбца>|<имя столбца>

<имя таблицы>::=<имя объекта>

<псевдоним таблицы>::=<имя объекта>

<имя столбца>::=<имя объекта>

<псевдоним столбца>::=<имя объекта>

<имя объекта>::=

<левая скобка><расширенное имя><правая скобка>|

“<расширенное имя>”|

<идентификатор>

<левая скобка>::=[

<правая скобка>::=]

<условие выборки>::=

<лог. терм>[OR<условие выборки>]

<лог. терм>::=

<лог. терм 1>[AND<лог. терм>]

<лог. терм 1>::= [NOT]{<предикат сравнения>|<условие выборки>}

<предикат сравнения>::=

<выражение><оператор сравнения><выражение> | <идентификатор столбца> LIKE  
<шаблон>

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

---

<шаблон> - строковая константа, в которой могут использоваться специальные символы:

\* - замена 0 или более символов;

? - замена 1 символа;

# - замена одной цифры;

[a-b] - замена одного из указанных символов.

<оператор сравнения>::= < | > | <= | >= | = | <>

<выражение>::=

<терм>|<выражение>{+|-}<терм>

<терм>::=<элемент выражения>|<терм>{\*|/}<элемент выражения>

<элемент выражения>::=

<идентификатор столбца>|(<выражение>)|<константа>

<константа>::=

<строковая константа>|

<целая константа>|

<вещественная константа>|

<логическая константа>|

<константа времени>

<строковая константа>::=

'<расширенное имя>'

<целая константа>::=

[-]<целая константа без знака>

<целая константа без знака>::=

<цифра>[<цифра>...]

<цифра>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<вещественная константа>::=

[-]<целая константа без знака>.<целая константа без знака>

<логическая константа>::= 0|1

<константа времени>::=

<целая константа без знака>

<шаблон> - строковая константа, в которой могут использоваться специальные символы:

\* - замена 0 или более символов;

? - замена 1 символа;

# - замена одной цифры;

[a-b] - замена одного из указанных символов.

<спецификация сортировки>::=

<спецификатор сортировки>[,<спецификатор сортировки>...]

<спецификатор сортировки>::=

<целая константа без знака>[ASC|DESC]



<расширенное имя>::= строка длиной не более 64 символов, содержащая любые символы, кроме '.', '[', ']', '!', '"', '\', а также символов с кодами 0..31. Первым символом не может быть пробел.

<идентификатор>::= строка длиной не более 64 символов, содержащая латинские буквы.

В выражении WHERE можно использовать функцию GetCurDate. Можно вызывать как GetCurDate, так и GetCurDate(). Эта функция возвращает текущее время в формате ДАТА/ВРЕМЯ.

В выражении, содержащем функцию GetCurDate, допускается использовать операции сложения, вычитания, сравнения (<, <=, =, <>, >, >=) даты и времени.

Минимальный интервал времени – 1 миллисекунда. Поэтому выражение «(GetCurDate - [Дата и время события]) < 1000» означает, что отрезок времени должен быть меньше одной секунды.

Максимально число миллисекунд, используемое в операциях сравнения – 2147483648.

Можно задать дату и время в формате «ЧАСЫ:МИНУТЫ:СЕКУНДЫ[.МИЛИСЕК] ДЕНЬ.МЕСЯЦ.ГОД», например, '09:18:45.345 08.03.2006'.



### 3 СПИСОК АТТРИБУТОВ БД

#### СПИСОК АТТРИБУТОВ БД СИСТЕМЫ КРУГ-2000 ДОСТУПНЫХ НА ЧТЕНИЕ ЧЕРЕЗ ДРАЙВЕР-ODBC

Словарь логических состояний
[1] Код
[2] Имя
Словарь типов линеаризации
[2] Тип
[3] Имя
[4] Номер
Словарь единиц измерений
[1] Код
[2] Имя
Словарь типов контроллеров
[1] Код
[2] Имя
Словарь типов обработок
[1] Код
[2] Название
Словарь типов абонентов в ПТК
[1] TypeSubscriber
[4] LongName
Словарь цветов
[1] Код
[2] RGB
[3] Name
Словарь типов строк заголовка
[1] Kod
[2] Name
Словарь типов датчиков
[1] Code
[2] Name
[2] Variable
Входная аналоговая
[1] Номер канала
[2] Номер переменной в УСО
[3] Номер платы
[4] Номер входа
[5] Позиция
[6] Имя 1
[7] Имя 2
[8] Единица измерения
[9] Тип датчика
[10] Тип линеаризации шкалы
[11] Начало шкалы
[12] Конец шкалы
[13] Нижняя предупредительная граница
[14] Верхняя предупредительная граница
[15] Нижняя предаварийная граница

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

[16]	Верхняя предаварийная граница
[17]	Гистерезис сигнализации
[18]	Время определения скорости изменения параметра
[19]	Граница сигнализации по скорости роста
[20]	Граница сигнализации по скорости падения
[21]	Постоянная фильтра от 0,000 до 1,000
[23]	Тип замены недостоверного значения
[24]	Рабочее значение переменной
[25]	Отсечка нуля от 0 до 30% шкалы
[26]	Апертура
[29]	Текущее значение после преобразования (контроллер)
[30]	Снятие переменной с опроса
[31]	Снятие переменной с сигнализации
[34]	Неисправность канала
[35]	Новое нарушение <Неисправность канала>
[38]	Нарушена нижняя предаварийная граница
[39]	Новое нарушение нижней предаварийной границы
[40]	Нарушена верхняя предаварийная граница
[41]	Новое нарушение верхней предаварийной границы
[42]	Нарушена нижняя предупредительная граница
[43]	Новое нарушение нижней предупредительной границы
[44]	Нарушена верхняя предупредительная граница
[45]	Новое нарушение верхней предупредительной границы
[46]	Нарушена граница по скорости роста
[47]	Новое нарушение по скорости роста
[48]	Нарушена граница по скорости падения
[49]	Новое нарушение по скорости падения
[50]	Переменная в норме
[51]	Переход переменной в нормальное состояние
[52]	Сигнализация по достоверности
[53]	Новая сигнализация по достоверности
[55]	Значение сигнала ниже начала шкалы
[56]	Значение сигнала ниже начала шкалы (новое)
[57]	Значение сигнала выше конца шкалы
[58]	Значение сигнала выше конца шкалы (новое)
[59]	Нарушение <Обрыв (<3,5 mA) >
[60]	Новое нарушение <Обрыв>
[61]	Нарушение <Перегрузка (>5,5/20,5 mA)>
[62]	Новое нарушение <Перегрузка>
[65]	Текущее значение после (CO)
[66]	Длинная позиция
[67]	Длинное имя
[73]	№ алгоритма нестандарт обработки 1
[77]	Вкл/выкл нестандарт обработок 1
[82]	Место обработки
[87]	Номер канала в комсервере
[89]	Метка времени
[90]	Снять с опроса в CO
[91]	Снять с сигнализации в CO
[92]	Номер УСО в канале
[94]	ALARM1

[95] ALARM2
[98] Цвет состояния
Аналоговая выходная
[1] Номер канала
[2] Номер УСО
[3] Номер платы
[4] Номер выхода
[5] Позиция
[6] Имя 1
[7] Имя 2
[15] Величина задания
[16] Значение ручного Задатчика / Положение МЭО
[17] Величина коэффиц Пропорцион
[18] Постоянная времени интегриров
[19] Величина коэффиц Дифференц
[20] Зона нечувствительности
[21] Коэф ослабл для зоны нечувств
[22] Верхняя граница отклонения от задания
[23] Нижняя граница отклонения от задания
[24] Верхнее огранич хода ИМ
[25] Нижнее огранич хода ИМ
[26] Верхняя граница сигнализ хода ИМ
[27] Нижняя граница сигнализ хода ИМ
[28] Тактовая частота / длительность импульса
[29] Переход к новому заданию
[30] Постоянная времени по заданию
[31] Коэффициент для форсир перехода (КФ)
[32] Скорость хода ИМ / время полного хода ИМ
[33] Текущее значение параметра
[34] Текущее значение параметра (демасштаб)
[35] Текущее значение задания (демасштаб)
[38] Рассогласование
[45] Значение выходного сигнала (для аналог р-ра)
[46] Значение при ручном дистанц управлении
[47] Текущий режим управления (ТРУ)
[48] Текущий режим ввода задания (ТРВЗ)
[56] Отклонение от верхней границы задания
[57] Новое отклонение от верхней границы задания
[58] Отклонение от нижней границы задания
[59] Новое отклонение от нижней границы задания
[60] Нарушение по верхней границе сигн хода ИМ
[61] Новое нарушение по верхней границе сигн хода ИМ
[62] Нарушение по нижней границе сигн хода ИМ
[63] Новое нарушение по нижней границе сигн хода ИМ
[66] Диагностика ЦАП
[67] Диагностика ЦАП (новая)
[68] Признак обрыва цепи
[74] Вид действия (1-прямой / 0-обратный)
[80] Снятие с сигнализации по заданию
[81] Снятие с сигнализации по положению ИМ
[88] Режим регулятора <Руч аппаратный>

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

[89] Режим регулятора <Руч дистанционный>
[90] Режим регулятора <Автоматический>
[92] Режим ввода задания <Ручной>
[93] Режим ввода задания <Внешний> (<Каскад>)
[124] Длинная позиция
[125] Длинное имя
[134] Основной/Резерв1/Резерв2/
[140] Метка времени
[143] Номер УСО в канале
[144] Тип регистра УСО
[145] ALARM1
[146] ALARM2
[147] Цвет состояния
Входная дискретная
[1] Номер канала
[2] Номер УСО
[3] Номер платы
[4] Номер входа
[5] Позиция
[6] Имя 1
[7] Имя 2
[8] Код цвета состояния <0>
[9] Код логического состояния <0>
[10] Код цвета состояния <1>
[11] Код логического состояния <1>
[12] Признак необходимости инверсии
[25] Снятие переменной с опроса
[26] Снятие переменной с сигнализации
[27] Текущее значение переменной
[28] Нарушение <Недостоверность>
[29] Новое нарушение <Недостоверность>
[37] Лог признак <Сигнализация>
[38] Лог признак <Норма>
[39] Лог признак <Новая сигнализация>
[40] Лог признак <Новая норма>
[45] Текущее значение в СО
[46] Длинная позиция
[47] Длинное имя
[56] Основной/резерв1/резерв2/
[61] Номер м/схемы (главной)
[62] Метка времени
[65] Номер УСО в канале
[66] Тип регистра УСО
[67] ALARM1
[68] ALARM2
[70] Цвет состояния
Дискретная выходная
[1] Номер канала
[2] Номер УСО
[3] Тип УСО
[4] Номер платы

[5] Номер выхода
[6] Позиция
[7] Имя 1
[8] Имя 2
[9] Код цвета состояния <0>
[10] Код логического состояния <0>
[11] Код цвета состояния <1>
[12] Код логического состояния <1>
[14] Признак необходимости инверсии
[17] Режим ввода символа
[20] Значение выходной переменной в контроллере
[26] Снятие с опроса
[27] Текущее значение переменной в контроллере
[37] Текущее значение переменной
[38] Новая сигнализация <Отказ>
[40] Сигнализация <Отказ>
[46] Текущее значение в СО
[47] Длинная позиция
[48] Длинное имя
[57] Основной/резерв1/резерв2/
[61] Номер канала в комсервере
[63] Метка времени
[66] Номер УСО в канале
[67] Тип регистра УСО
[68] ALARM1
[69] ALARM2
[71] Цвет состояния
РВ Составная
[1] Номер канала
[2] Номер переменной в УСО
[3] Позиция
[4] Имя 1
[5] Имя 2
[6] Тип переменной
[7] Единица измерения
[8] Максимум
[9] Минимум
[10] Начальное значение
[11] Текущее значение
[12] Текущее значение 2
[15] Начальное значение (Строка)
[16] Текущее значение (Строка)
[17] Текущее значение (Длинная строка)
[18] Цвет отображения строки
[19] Начальное значение (логич)
[20] Текущее значение (логич)
[26] Начальное значение 2 (вещественное)
[27] Основной/резерв1/резерв2/
[32] Метка времени
Абоненты
[1] UniqueIDSubscriber

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

[2] TypeSubscriber
[3] NumUser
[4] ShortName
[5] LongName
[6] NameComputer
[7] bReservOn
[8] bTransitOn
Канал
[1] Nkanal
[2] IP address
[3] IP port
[4] Timer
[6] MaxVA
[7] MaxAV
[8] MaxDI
[9] MaxDO
[10] MaxHI
[11] Online
[12] Type
[32] LevelRes
[33] IdAbonent
Описание ФС Передача
[1] CloseOpen
[5] DescribeFileName
[6] DataFileName
[7] OutSystemName
[8] TimeOpros
Описание ФС Прием
[1] CloseOpen
[5] DescribeFileName
[6] DataFileName
[7] OutSystemName
[8] TimeOpros
Адреса абонентов
[1] UniqueIDSubscriber
[2] IPSubscriber
[3] PriorityNetwork
[8] dwTimeDiag
[9] bQuanAttempts
[11] bFault
[15] ftLastFault
[17] bNumSubNet
[18] ftLastRestoring
[20] dwPort
Архивные накопители
[1] Driveld
[2] DriveTypeld
[3] TotalSize
[4] FreeSize
[5] FullnessPercent
[6] DiskExistance



[7] FSType
[8] WriteProtectStatus
[9] ReadyStatus
[10] Label
[11] OverflowStatus
[12] DiskBusyStatus
[13] AutoSaveSign
[16] DriveLetter
[17] DriveName
[18] AutoCopyPeriod
[20] BasePriority
[25] PeriodCheck
[26] Path
Типы архивных накопителей
[1] DriveTypeId
[2] TypeName
[3] VendorName
[6] SizeReserv
Паспорта архивных дисков
[1] DiskId
[2] SystemName
[4] UserDiskName
[5] FirstRecordTime
[6] LastRecordTime
Архив печатных документов
[1] Path
Описание двухмерных
[1] Код функции
[2] Код типа обработки
[5] Число X
Данные двухмерных
[1] Код функции
[2] Номер
[3] Значение X
[4] Значение Y
Описание трехмерных
[1] Код функции
[2] Код типа обработки
[5] Число X
[6] Число Y
Данные трехмерных по X
[1] Код функции
[2] Номер
[3] Значение X
Данные трехмерных по Y
[1] Код функции
[2] Номер
[3] Значение Y
Данные трехмерных по Z
[1] Код функции
[2] Номер

## СРЕДСТВА ИНТЕГРАЦИИ В АСУП

	[3] Значение Z
Диспетчер трендов	
	[1] Количество самописцев
	[3] Количество перьев
	[5] Дата и время первоначального запуска
	[6] Дата и время предыдущего запуска
	[7] Дата и время последнего останова
	[8] Дата и время текущего запуска
	[9] Количество запусков
Самописцы	
	[2] Название
	[3] Количество перьев
	[4] Полный интервал обработки
	[10] Дата и время последней обработки
	[11] Полный интервал записи
	[17] Количество точек
	[19] Номер последней записанной точки
	[20] Дата и время последней записи
Перья	
	[0] RecordID
	[1] Название
	[2] Идентификатор самописца
	[6] Код типа значения
	[7] Размер значения
	[8] Тип обработки
	[10] Номер в самописце
Стандартные интервалы времени	
	[2] Название интервала
	[3] Дата и время события
Ribbon X (где X идентификатор самописца)	
Ribbon X Сводная (значения трендов параметров)	
	[2] Идентификатор записи
	[3] Дата и время записи
	[4...] Массив значений пера. Соответствует количеству перьев