

УТВЕРЖДЕН
643.72410666.00067-07 98 01-ЛУ

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 20. Обеспечение работы с
СУБД MS SQL Server.
Компонент «TDS_FDW»

643.72410666.00067-07 98 01-20

Листов 17

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонента «TDS_FDW» (далее по тексту – «компонент» или TDS_FDW), предназначенного для миграции БД на платформе Microsoft SQL Server в СУБД «Jatoba» и прямой доступ к СУБД на платформе Microsoft SQL Server (далее – MS SQL) из СУБД «Jatoba».

Настоящее руководство предназначено для администраторов СУБД.

Версия компонента — 2.0.3.

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием

СОДЕРЖАНИЕ

1. Назначение компонента	4
1.1. Условия применения.....	5
2. Установка и настройка	6
2.1. Установка компонента «TDS_FDW» в ОС GNU/Linux.....	6
2.2. Установка расширения «TDS_FDW».....	8
3. Функциональные возможности компонента	9
3.1. Создание внешнего сервера	9
3.2. Создание внешней таблицы	10
3.3. Задание сопоставления пользователей.....	12
3.4. Импорт внешней схемы	12
3.5. Определение локальных переменных	12
3.6. Пример использования компонента	13
4. Удаление компонента	15
Перечень сокращений.....	16

1. НАЗНАЧЕНИЕ КОМПОНЕНТА

Компонент «TDS_FDW» предназначен для импорта данных из БД MS SQL в СУБД «Jatoba», который помогает сопоставить типы данных.

Сопоставляемые типы данных приведены в таблице 1.1.

Таблица 1.1 – Сопоставляемые типы данных

MS SQL	СУБД «Jatoba»
bigint	bigint
integer	integer
smallint	Smallint
Real	Real
Numeric(18,0)	Numeric(18,0)
Binary(10)	Bytea
Bit	Smallint
Char(8)	Character(8)
nchar(8)	Character(8)
Float	Double precision
Decimal(18,0)	Numeric(18,0)
Ntext	Text
Text	Text
Nvarchar(8)	Character varying(8)
Date	Date
Datetime	Timestamp without time zone
Time	Time without time zone
Timestamp	Bytea
Money	Money
Tinyint	Smallint
Smallmoney	money

1.1. Условия применения

Компонент «TDS_FDW» может использоваться с СУБД «Jatoba» версий 5.x и выше, под управлением операционных систем GNU/Linux.



Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!



В текущей реализации компонента не поддерживается управление через компонент пользовательского веб-интерфейса для администраторов «Jatoba data safe».

Ограничений по совместимости с другими компонентами нет.

2. УСТАНОВКА И НАСТРОЙКА

Установка компонента должна производиться от имени пользователя, обладающего административными привилегиями в системе. Данный компонент штатным образом может быть установлен только с СУБД «Jatoba» (см. документ «Защищенная система управления базами данных «Jatoba». Руководство по установке).

2.1. Установка компонента «TDS_FDW» в ОС GNU/Linux

Компонент устанавливается в составе СУБД «Jatoba». Его возможно установить при первичной установке либо доустановить.

Установку компонента возможно провести двумя способами:

- 1) установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него;
- 2) установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.



Перед установкой компонента «TDS_FDW» в ОС GNU/Linux на базе Debian 10 необходимо добавить архивный репозиторий в список доступных репозиториях ОС при помощи команды:

```
echo "deb http://archive.debian.org/debian buster  
main" > /etc/apt/sources.list
```

Перед установкой компонента необходимо установить библиотеки:

- «libsybdb5»;
- «freetds-dev»;
- «freetds-common».

Для систем на основе пакетного менеджера APT команда установки библиотек следующая:

```
apt-get install libsybdb5 freetds-dev freetds-common
```

Компонент выполнен в виде отдельного deb или rpm-пакета. Установка компонента осуществляется средствами пакетного менеджера ОС. Для разных типов пакетных менеджеров команда установки немного отличается. Ниже приведены основные типы:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

– для систем на основе пакетного менеджера APT (к таким системам относятся все ОС семейства Debian, использующие deb-пакеты) команда установки следующая:

```
apt-get install jatoba5-tds-fdw
```

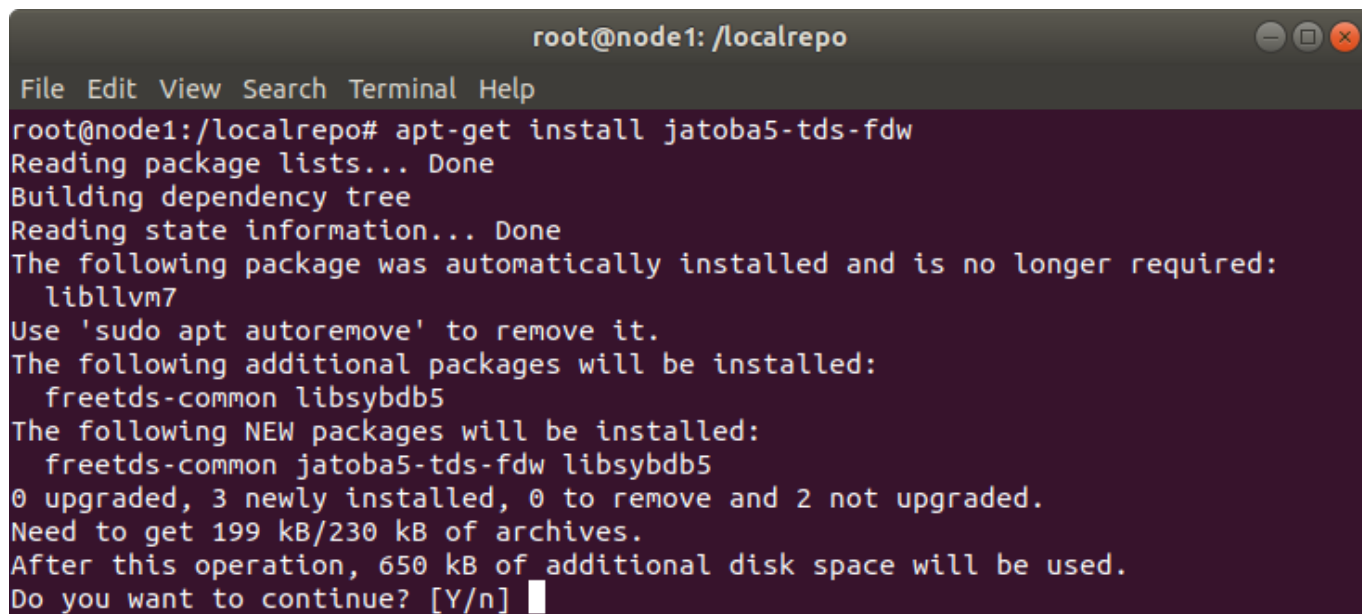
– для систем на основе пакетных менеджеров YUM/DNF (к таким системам относятся все ОС семейства RedHat и вышедшие из нее, использующие rpm-пакеты) команда установки следующая:

```
yum install jatoba5-tds-fdw
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

– ALT Linux использует пакетный менеджер APT, но распространяется в виде rpm-пакетов и для нее команда установки выглядит аналогично Debian:

```
apt-get install jatoba5-tds-fdw
```



```
root@node1: /localrepo
File Edit View Search Terminal Help
root@node1:/localrepo# apt-get install jatoba5-tds-fdw
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  freetds-common libsybdb5
The following NEW packages will be installed:
  freetds-common jatoba5-tds-fdw libsybdb5
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 199 kB/230 kB of archives.
After this operation, 650 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

– openSUSE также распространяется в виде rpm-пакетов, но использует собственный пакетный менеджер zypper, для нее команда установки выглядит следующим образом:

```
zypper install jatoba5-tds-fdw
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется.

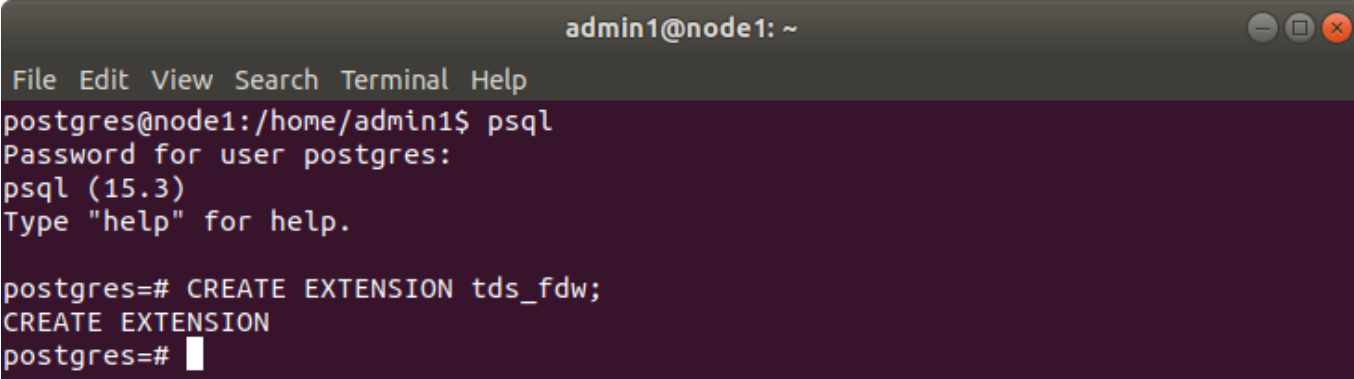
Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды `install` нужно использовать соответствующую данному пакетному менеджеру команду удаления (`remove`, `purge`, `erase` и т.п.).

Для получения детальной информации по пакетному менеджеру рекомендуется обратиться к документации по ОС.

2.2. Установка расширения «TDS_FDW»

Расширение устанавливается при помощи SQL-команды:

```
CREATE EXTENSION tds_fdw;
```



```
admin1@node1: ~  
File Edit View Search Terminal Help  
postgres@node1:/home/admin1$ psql  
Password for user postgres:  
psql (15.3)  
Type "help" for help.  
  
postgres=# CREATE EXTENSION tds_fdw;  
CREATE EXTENSION  
postgres=#
```

Рисунок 2.1 – Команда установки расширения

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ КОМПОНЕНТА

3.1. Создание внешнего сервера

Внешний сервер создается командой:

```
CREATE SERVER mssql_svr  
    FOREIGN DATA WRAPPER tds_fdw  
    OPTIONS (servername '127.0.0.1', port '1433', database  
'tds_fdw_test', tds_version '7.1');
```

Параметры команды приведены в таблице 3.1.

Таблица 3.1 – Параметры команды создания внешнего сервера

Название параметра	Обязательный	Описание
servername	+	адрес сервера внешней базы данных
port	+	порт внешней базы данных
database	–	название внешней базы данных
dbuse	–	если параметр установлен в «0» (значение по умолчанию), подключение идет к базе данных, указанной в параметр «database»; если в «1» — подключение идет к базе данных, полученной через вызов метода «dbuse()»
language	–	язык сообщений и локаль для даты (по умолчанию — «us_english»)
tds_version	–	версия протокола TDS
msg_handler	–	функция, используемая обработчиком сообщений TDS: «blackhole» (значение по умолчанию, сообщения TDS будут игнорироваться), «notice» (сообщения TDS будут выводиться как уведомления СУБД «Jatoba»)
fdw_startup_cost	–	числовое значение стоимости запуска чтения информации с внешнего сервера

Название параметра	Обязательный	Описание
fdw_tuple_cost	—	числовое значение стоимости чтения строк на внешнем сервере

3.2. Создание внешней таблицы

Внешнюю таблицу можно определить четырьмя способами:

- 1) С помощью названия таблицы:

```
CREATE FOREIGN TABLE mssql_table (  
    id integer,  
    data varchar)  
SERVER mssql_svr  
OPTIONS (table_name 'dbo.mytable', row_estimate_method  
'showplan_all');
```

- 2) С помощью названия схемы и названия таблицы:

```
CREATE FOREIGN TABLE mssql_table (  
    id integer,  
    data varchar)  
SERVER mssql_svr  
OPTIONS (schema_name 'dbo', table_name 'mytable',  
row_estimate_method 'showplan_all');
```

- 3) С помощью запроса:

```
CREATE FOREIGN TABLE mssql_table (  
    id integer,  
    data varchar)  
SERVER mssql_svr  
OPTIONS (query 'SELECT * FROM dbo.mytable',  
row_estimate_method 'showplan_all');
```

- 4) Указывая название столбца:

```
CREATE FOREIGN TABLE mssql_table (  

```

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
id integer,
col2 varchar OPTIONS (column_name 'data'))
SERVER mssql_svr
OPTIONS (schema_name 'dbo', table_name 'mytable',
row_estimate_method 'showplan_all');
```

Параметры команды приведены в таблице 3.2.

Таблица 3.2 – Параметры команды создания внешней таблицы

Название параметра	Обязательный	Описание
query	+ (если не указан параметр «table_name»)	запрос, выполняющийся на внешней таблице
schema_name	–	название схемы (не обязательно указывать, если схема была указана в названии таблицы в параметре «query» или «table_name»)
table_name	+ (если не указан параметр «query»)	название внешней таблицы
match_column_names	–	сопоставлять ли названия столбцов таблицы Jatoba и внешней таблицы по названию (по умолчанию сопоставление идет по порядку)
use_remote_estimate	–	оценивать ли размер внешней таблицы через операцию, определяемую параметром «row_estimate_method» (если нет, то используется параметр «local_tuple_estimate»)
local_tuple_estimate	–	числовая оценка количества строк во внешней таблице
row_estimate_method	–	метод, использующийся для оценки размера внешней таблицы: «execute» (по умолчанию) выполняет запрос к внешней таблице, «showplan_all» получает примерное количество строк используя функцию MS SQL «SET SHOWPLAN_ALL»

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

3.3. Задание сопоставления пользователей

Соответствие пользователя задается командой:

```
CREATE USER MAPPING FOR postgres
    SERVER mssql_svr
    OPTIONS (username 'sa', password '');
```

Параметры команды приведены в таблице 3.3.

Таблица 3.3 – Параметры команды задания сопоставления пользователей

Название параметра	Обязательный	Описание
username	+	имя пользователя внешнего сервера
password	+	пароль пользователя внешнего сервера

3.4. Импорт внешней схемы

Используется команда

```
IMPORT FOREIGN SCHEMA dbo
    EXCEPT (mssql_table)
    FROM SERVER mssql_svr
    INTO public
    OPTIONS (import_default 'true');
```

Параметры команды приведены в таблице 3.4.

Таблица 3.4 – Параметры команды импорта внешней схемы

Название параметра	Обязательный	Описание
import_default	–	включать ли выражения столбца DEFAULT в определении внешних таблиц (по умолчанию выключено)
import_not_null	–	включать ли ограничения столбца NOT NULL в определении внешних таблиц (по умолчанию включено)

3.5. Определение локальных переменных

Используется следующая команда:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
SET tds_fdw.show_finished_memory_stats=1;
```

Локальные переменные, используемые компонентом, приведены в таблице 3.5.

Таблица 3.5 – Локальные переменные компонента «TDS_FDW»

Название	Описание
tds_fdw.show_before_row_memory_stats	выводить статистику контекста памяти в журнал СУБД «Jatoba» перед чтением каждой строки
tds_fdw.show_after_row_memory_stats	выводить статистику контекста памяти в журнал СУБД «Jatoba» после чтения каждой строки
tds_fdw.show_finished_memory_stats	выводить статистику контекста памяти в журнал СУБД «Jatoba» после завершения запроса

3.6. Пример использования компонента

1. Создать расширение с помощью команды:

```
CREATE EXTENSION tds_fdw;
```

2. В СУБД «Jatoba» создать объект типа «FOREIGN SERVER».

Это специальный объект, который позволяет через расширение tds_fdw получать доступ к данным MS SQL.

```
CREATE SERVER mssql_srv  
FOREIGN DATA WRAPPER tds_fdw  
OPTIONS (servername '192.168.243.223'), port '1433', database  
'aszp';
```

3. Создать маппинг для созданного сервера, т.е. от имени и с правами какого пользователя будет проводиться подключение к MS SQL:

```
CREATE USER MAPPING FOR postgres  
SERVER mssql_srv  
OPTIONS (user 'slog', password 'p@ssword');
```

4. Импортировать исходную схему «dbo» MS SQL в схему «public» СУБД «Jatoba»:

```
IMPORT FOREIGN SCHEMA dbo FROM SERVER "mssql_srv" INTO public;
```

В результате станет доступным просмотр созданных таблиц и импорт данных.

4. УДАЛЕНИЕ КОМПОНЕНТА

Удаление компонента проводится SQL-командой:

```
DROP EXTENSION tds_fdw;
```

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

MS SQL	–	Microsoft SQL Server
SQL	–	Structured Query Language
TDS	–	Tabular Data Stream
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

Лист регистрации изменений

№ изменения: _____

Подпись отв. лица: _____

Дата внесения изм: _____