

Перенаправление сложных Select-запросов

Настройка отдельного пула запросов

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Настройка отдельного пула запросов	4
1. Настроить соединение отдельного пула запросов	4
2. Включить функциональность отдельного пула запросов	5
3. Настроить инструмент Resource Governor	5
4. Выполнить запрос в отдельном пуле запросов	6

Настройка отдельного пула запросов



Некоторые сложные запросы к базе данных могут нагружать ресурсы сервера базы данных на 100% в течение продолжительного времени. Это приводит к затруднению или невозможности работы других пользователей.

Виды сложных запросов:

- Неоптимально составленные запросы в динамических группах, блоках итогов.
- Сложные аналитические выборки в блоках итогов.

Назначение отдельного пула запросов — обработка сложных `Select`-запросов, которые не являются частью транзакции и вынесены в отдельный пул.

Использование отдельного пула запросов позволяет:

- Ограничить ресурсы, которые выделяются сервером базы данных, на обработку сложных `Select`-запросов.
- Уменьшить влияние обработки сложных `Select`-запросов на работу других пользователей и частей системы.

1. Настроить соединение отдельного пула запросов

MS SQL Server позволяет ограничивать выделяемые ресурсы с помощью встроенного инструмента — [Resource Governor](#). Ранжирование соединений в Resource Governor базируется на информации о подключении, а не о конкретном запросе. Работу инструмента сложно увидеть на незагруженном сервере и на "коротких" запросах. Эффект от использования Resource Governor наблюдается, когда сервер баз данных загружен на 100%, а сложный запрос выполняется продолжительное время.

Чтобы **настроить соединение отдельного пула запросов**, откройте конфигурационный файл `ConnectionStrings.config` и для свойства `App` или `Application Name` допишите суффикс `_Limited`. Эта настройка позволяет использовать специальное соединение, которое разделит запросы на простые и потенциально сложные.

Виды соединений:

- Если в строке соединения конфигурационного файла `ConnectionStrings.config` для свойства `App` **не указано значение**, то для соединений будут использованы значения по умолчанию. Значение по умолчанию для общего соединения — `".Net SqlClient DataProvider"`, значение по умолчанию для соединения отдельного пула запросов — `".Net SqlClient DataProvider_Limited"`.
- Если в строке соединения конфигурационного файла `ConnectionStrings.config` для свойства `App` **указано значение** `"creatio"`, то значение свойства для соединения отдельного пула запросов будет заменено значением `"creatio_Limited"`.

Пример настройки свойства `App` строки соединения

```
<add name="db" connectionString="App=creatio; Data Source=dbserver\mssql2016; Initial Catalog
```

Таким образом, при загрузке дашбордов или фильтрации разделов с помощью динамических групп приложение создает дополнительные соединения с базой данных. В отличие от основных соединений, эти соединения в названиях содержат суффикс `_Limited`.

Важно. При использовании отдельного пула запросов не выполняется ограничение ресурсов. Использование суффикса `_Limited` позволяет выполнять ранжирование соединений средствами Resource Governor.

2. Включить функциональность отдельного пула запросов

Чтобы **включить функциональность отдельного пула запросов**, в файле

`..\Terrasoft.WebApp\Web.config` установите значение `true` для ключа `UseQueryKinds` элемента `<appSettings>`. Ключ `UseQueryKinds` обеспечивает отправку запросов из дашбордов и динамических групп в соединения, которые в названии содержат суффикс `_Limited`.

```
..\Terrasoft.WebApp\Web.config
```

```
<add key="UseQueryKinds" value="true" />
```

3. Настроить инструмент Resource Governor

Настройка инструмента Resource Governor подразумевает настройку групп и пула.

Чтобы **настроить инструмент Resource Governor**, выполните SQL-скрипт.

Пример настройки групп и пула

```
ALTER RESOURCE POOL poolLimited WITH (
    MAX_CPU_PERCENT = 20,
    MIN_CPU_PERCENT = 0
    -- REQUEST_MAX_MEMORY_GRANT_PERCENT = value
    -- REQUEST_MAX_CPU_TIME_SEC = value
    -- REQUEST_MEMORY_GRANT_TIMEOUT_SEC = value
    -- MAX_DOP = value
    -- GROUP_MAX_REQUESTS = value
);
GO
--- Create a workload group for off-hours processing
--- and configure the relative importance.
CREATE WORKLOAD GROUP groupLimited WITH (IMPORTANCE = LOW) USING poolLimited
```

```
GO ALTER RESOURCE GOVERNOR RECONFIGURE;
GO
```

Настройка инструмента подробно описана в официальной [документации Resource Governor](#).

Для каждого нового соединения выполняется функция-классификатор, которая возвращает название группы.

Пример функции-классификатора

```
USE [master]
GO

ALTER FUNCTION [dbo].[fnProtoClassifier]()
    RETURNS sysname
    WITH SCHEMABINDING
AS
BEGIN
    IF(app_name() like '%_Limited')
    BEGIN
        RETURN N'groupLimited'
    END
    RETURN N'default'
END;
```

4. Выполнить запрос в отдельном пуле запросов

Чтобы **выполнить запрос в отдельном пуле запросов**, получите специальный `DBExecutor`, передав в качестве дополнительного параметра значение `Limited` из перечисления `QueryKind`.

В приведенном ниже примере `QueryKind` — это аргумент метода `EnsureDBConnection()`. Значение аргумента приходит в клиентском `ESQ`-запросе, устанавливается в серверный `ESQ`-запрос и в `Select`-запрос.

Получение `DBExecutor` в зависимости от полученного `QueryKind`

```
using (DBExecutor executor = userConnection.EnsureDBConnection(QueryKind)) {
    // ...
};
```

Вызов `EnsureDBConnection(QueryKind.General)` эквивалентен вызову `EnsureDBConnection()` без указания `QueryKind`.

Таким образом, если при создании экземпляра класса `Terrasoft.EntitySchemaQuery` во front-end части приложения установить признак `QueryKind.Limited`, то это значение будет передано на сервер и запросу будет обеспечен специальный `DBExecutor`, который использует отдельный пул запросов.

Пример установки признака `QueryKind.Limited` клиентскому ESQ -запросу в схеме `ChartModule`

```
...
getChartDataESQ: function() {
    return this.Ext.create("Terrasoft.EntitySchemaQuery", {
        rootSchema: this.entitySchema,
        queryKind: Terrasoft.QueryKind.LIMITED
    });
},
...
```

Важно. Если в программном коде присутствуют вложенные вызовы `userConnection.EnsureDBConnection(QueryKind)`, то необходимо убедиться, что на всех уровнях вложенности используется одно и то же значение `QueryKind`.