

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 20

Мониторинг нагрузки сервера

Цель: изучение мониторинга работы сервера

Теоретические сведения

Наблюдение за базами данных выполняется с целью оценки производительности сервера. Эффективное наблюдение подразумевает регулярное создание моментальных снимков текущей производительности для обнаружения процессов, вызывающих неполадки, и постоянный сбор данных для отслеживания тенденций роста или изменения производительности.

Постоянная оценка производительности базы данных помогает добиться оптимальной производительности путем минимизации времени ответа и максимального увеличения пропускной способности. Приблизительный сетевой трафик, дисковый ввод-вывод и загрузка ЦП — ключевые факторы, влияющие на производительность. Следует тщательно проанализировать требования приложения, понять логическую и физическую структуру данных, оценить использование базы данных и добиться компромисса между такими конфликтующими нагрузками, как оперативная обработка транзакций (OLTP) и поддержка решений.

Мониторинг и настройка производительности баз данных

В состав Microsoft SQL Server и операционной системы Microsoft Windows входят служебные программы, позволяющие следить за текущим состоянием базы данных и измерять производительность, если это состояние меняется. Для наблюдения за Microsoft SQL Server можно использовать целый ряд средств и методик. Наблюдение за SQL Server позволяет решать следующие задачи:

1. Определять возможности увеличения производительности. Например, выполняя мониторинг времени ответа для часто используемых запросов, можно определить, требуется ли изменить текст запроса или индексы таблицы.

2. Оценивать активность пользователей. Например, выполняя мониторинг пользователей, которые подключаются к экземпляру SQL Server, можно определить, правильно ли настроены параметры безопасности, и проверить работу приложений и систем разработки. Контролируя выполнение SQL-запросов, можно определить, правильно ли они написаны, и проверить результаты, которые они возвращают.

3. Устранять проблемы или отлаживать компоненты приложений, например, хранимые процедуры.

Мониторинг в динамической среде

Изменение этих условий приведет к изменению производительности. По результатам оценки можно заметить изменения производительности при увеличении числа пользователей, изменении методов доступа пользователей

и методов соединения, при увеличении объема содержимого базы данных, изменении клиентского приложения и данных в приложении, а также при усложнении запросов и увеличении объема сетевого трафика. С помощью средств контроля производительности можно связывать изменения отдельных показателей производительности с изменениями условий и сложных запросов. Примеры:

1. Отслеживая время отклика на часто используемые запросы, можно определить, нужно ли изменять запросы или индексы опрашиваемых таблиц.

2. Отслеживая выполнение запросов Transact-SQL можно определить правильность их написания, а также соответствие ожидаемым результатам.

3. Отслеживая пользователей, пытающихся подключиться к экземпляру SQL Server, можно проверить надежность защиты и протестировать приложения или системы разработки.

Время отклика — это время ожидания возврата пользователю первой строки результирующего набора в форме визуального подтверждения обработки запроса. Пропускная способность — это общее количество запросов, которые сервер может обработать за единицу времени.

С увеличением числа пользователей растет соперничество за ресурсы сервера, что в свою очередь увеличивает время ответа и уменьшает общую пропускную способность.

Задачи наблюдения и настройки производительности:

1. Мониторинг компонентов SQL Server. Необходимые действия для мониторинга компонентов SQL Server, такие как монитор активности, расширенные события, динамические административные представления и функции и т. д.

2. Средства контроля и настройки производительности. Список средств наблюдения и настройки, доступных в SQL Server, например, статистики динамических запросов и помощник по настройке ядра СУБД.

3. Обновление баз данных с помощью помощника по настройке запросов. Поддержание стабильной производительности рабочей нагрузки во время обновления до нового уровня совместимости базы данных.

4. Мониторинг производительности с использованием хранилища запросов. Использование хранилища запросов для автоматической регистрации журнала запросов, планов и статистики выполнения и сохранение этих данных для просмотра.

5. Формирование базовых показателей производительности. Инструкции по формированию базовых показателей производительности.

6. Локализация проблем производительности. Локализация проблем производительности базы данных.

7. Выявление узких мест. Наблюдение за производительностью сервера и отслеживание его работы для выявления узких мест.

8. Использование динамических административных представлений для определения статистики использования и производительности представлений. Рассматриваются методы и скрипты, используемые для получения информации о производительности запросов.

9. Мониторинг производительности и действий сервера. Использование средств наблюдения за производительностью и активностью SQL Server и Windows.

10. Отслеживание использования ресурсов. Использование системного монитора (также известного как perfmon) для измерения производительности SQL Server с помощью счетчиков производительности.

В SQL Server встроено два инструмента: Activity Monitor (монитор активности) и Data Collector (сборщик данных). С их помощью выполняются практически все задачи диагностики.

Activity Monitor – это утилита, позволяющая оценивать активность пользователей приложения или сети. Она показывает текущее состояние SQL Server, осуществляемые на момент проверки процессы и то, как они отражаются на производительности СУБД.

Activity Monitor выглядит как окно с несколькими вкладками.

Администратор базы данных может открыть такие панели:

1. Overview (обзор). На этой панели демонстрируется время обработки запросов процессором, количество ожидающих запросов, количество запросов в секунду, ввод и вывод данных.

2. Processes (процессы). На этой панели отражаются все активные процессы и подробная информация по ним. В Processes также можно запустить скрипт, который автоматически анализирует выбранный процесс.

3. Resource Waits (ожидающие ресурсы). На этой панели отображается, какие ресурсы необходимы СУБД для выполнения заданных функций. В перечень ресурсов входит объем оперативной памяти и сервера, сети, компиляция и др. В этой же панели администратор базы данных может просмотреть общий и средний промежуток времени ожидания ресурсов.

4. Data File I/O (ввод-вывод данных). На этой панели отражаются все операции, связанные с внесением изменений в файлы БД, а также полная информация об этих файлах.

5. Recent Expensive Queries (последние ресурсоемкие запросы). На этой панели отражаются те запросы, которые были выполнены в течение ближайших 30 секунд, и обработка которых затребовала наибольшего числа ресурсов. В некоторых версиях SQL Server эта панель называется Activity Expensive Queries (активные ресурсоемкие запросы).

Сбор и обработка данных с помощью Activity Monitor (рисунок 1) ведется в режиме реального времени и только при условии развертывания панели. Если администратор базы данных сворачивает панель, обработка информации прекращается. При этом на экране можно развернуть все пять панелей, чтобы оценить активность пользователей по разным параметрам.

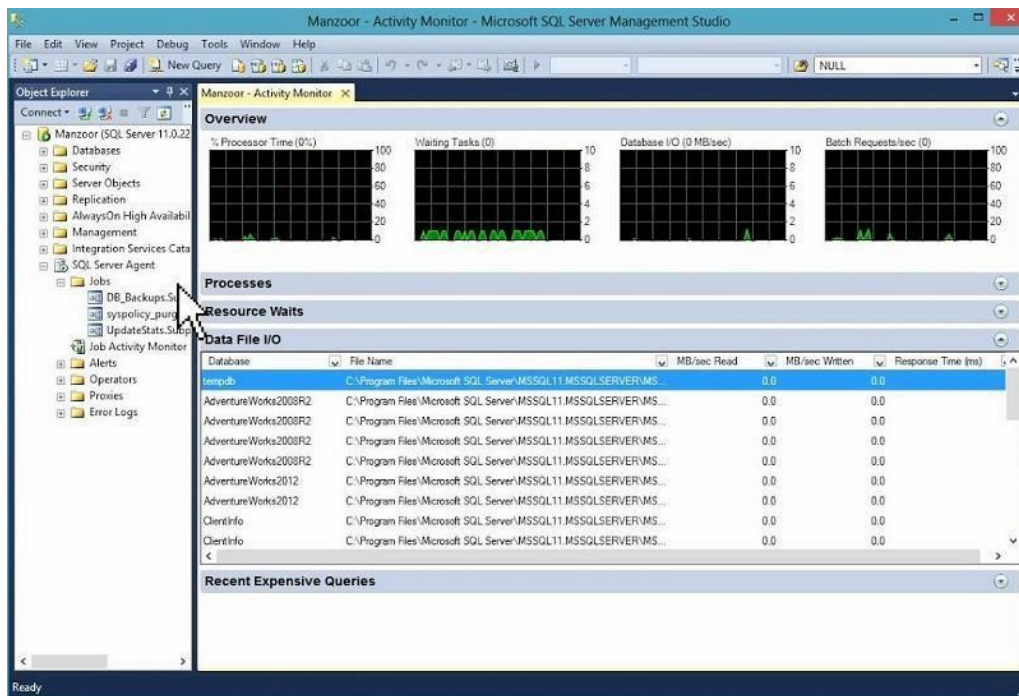


Рисунок 1- Сбор и обработка данных с помощью Activity Monitor

Для облегчения процесса использования Activity Monitor можно фильтровать, сортировать и менять местами столбцы с данными диагностики. Это делается с помощью компьютерной мыши прямо на развернутой панели.

Системная процедура **sp_who**.

Предоставляет сведения о текущих пользователях, сеансах и процессах в экземпляре Microsoft Компонент SQL Server Database Engine. Данные могут быть отфильтрованы, чтобы возвращать только те процессы, которые не простаивают, принадлежат конкретному пользователю или принадлежат определенному сеансу.

Общий формат запуска процедуры

sp_who [[@loginame =] «login» | session ID | «ACTIVE»]

Аргументы

[@loginame =] «login» – определяет процессы, принадлежащие конкретному имени входа.

Session ID – идентификатор сеанса является идентификационным номером сеанса, принадлежащего SQL Server экземпляру.

«ACTIVE» – исключает сеансы, ожидающие следующей команды от пользователя.

Если значение не указано, эта процедура возвращает все сеансы, принадлежащие экземпляру.

Пример использования:

```
USE master; GO
```

```
EXEC sp_who «10» --specifies the process_id;
```

```
GO
```

У хранимой процедуры `sp_who` есть недокументированный вариант – `sp_who2`. Эта хранимая процедура запускается на выполнение точно так же, но возвращает более подробную информацию.

Преимуществом использования Activity Monitor является то, что использование этого инструмента практически не отражается на производительности СУБД.

Что касается недостатков этого инструмента, в их число можно включить:

- ограниченное количество параметров, по которым ведется диагностика;

- отсутствие эталона, то есть шаблона с указанием допустимых значений параметров;

- отсутствие возможности формирования отчета о результатах проверки для их последующего анализа.

Из этого можно сделать вывод, что Activity Monitor – это прекрасный инструмент для проведения быстрой диагностики и поиска запросов, затратных с точки зрения потребления ресурсов.

Ход работы:

1. Запустить мониторинг активности.

Шаги:

1. Открыть SQL Server Management Studio (SSMS).
2. Подключиться к нужному экземпляру SQL Server.
3. В Обозревателе объектов щелкнуть правой кнопкой мыши по серверу.

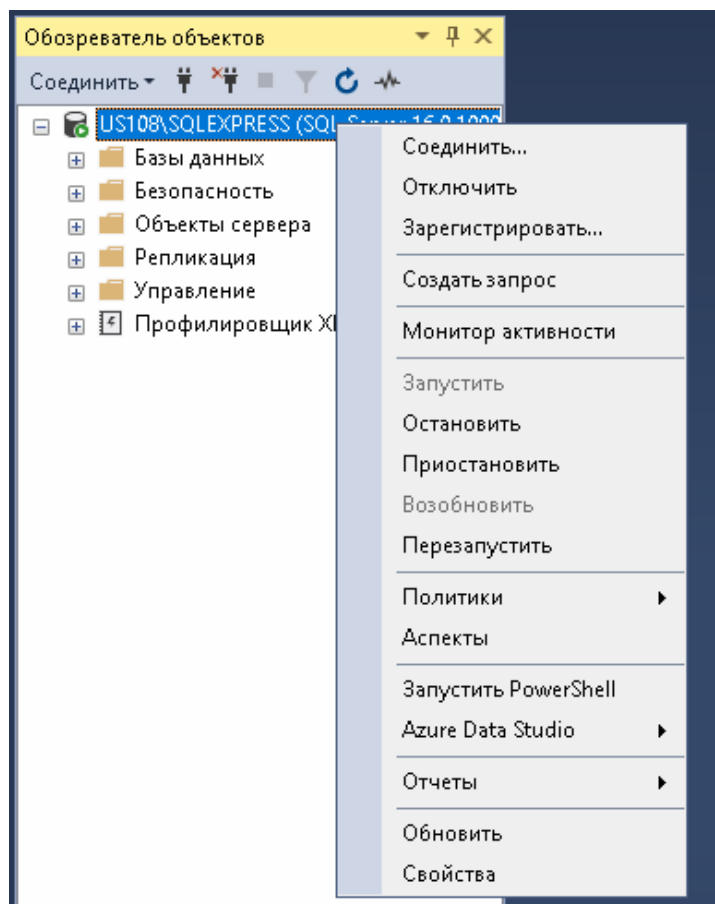


Рисунок 2 – Контекстное меню

4. Выбрать Монитор активности.
5. Дождаться загрузки информации о сервере.

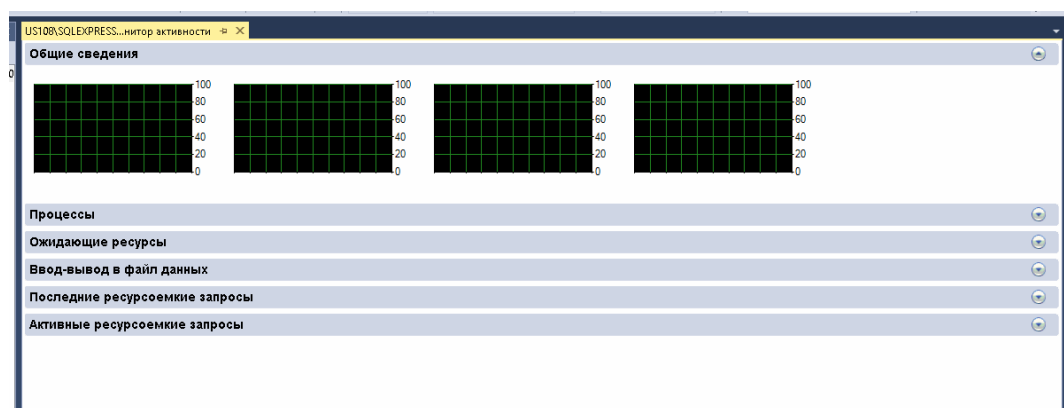


Рисунок 3 – Монитор активности

2. Выявить выполняемые процессы, время ожидания ресурсов и другие отображаемые параметры.

Шаги:

1. В окне Монитор активности открыть вкладку Процессы.

ID	PID	Имя	Баз	Сос	Ком	При	Время о...	Тип	Ожи	Б...	Г...	Заг...	Имя...	Гру...
52	1	NT SER...	master			SQLServ...	0				32	US108	internal	
53	1	US108\va...	tempdb	RUNNING	SELECT	Microsoft...	0				32	US108	internal	
56	1	US108\va...	master			Microsoft...	0				0	US108	internal	
65	1	US108\va...	master			Microsoft...	0				32	US108	internal	

Рисунок 4 – Вкладка Процессы

2. Перейти во вкладку Ожидающие ресурсы и изучить параметры ожидания ресурсов.

Категория ожидания	Время ожидания (мс в...)	Последнее время ожид...	Среднее количество ож...	Совокупное время ожидани...
Buffer I/O	0	0	0,0	130
Buffer Latch	0	0	0,0	1
Latch	0	0	0,0	0
Lock	0	0	0,0	4
Logging	0	0	0,0	24
Memory	0	0	0,0	0
Network I/O	0	0	0,0	0
Other	0	0	0,0	6
SQLCLR	0	0	0,0	0

Рисунок 5 – Вкладка Ожидающие ресурсы

3. Вкладка Ввод-вывод данных в файл позволяет отследить операции ввода-вывода данных.

База данных	Имя файла	МБ/с считано	МБ/с записано	Время ответа (мс)
tempdb	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	1
""	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
""	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
COMPANIA	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
COMPANIA	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
master	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
master	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
model	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
model	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0
model	C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSS...	0,0	0,0	0

Рисунок 6 – Вкладка Ввод-вывод в файл данных

4. Вкладка Последние ресурсоёмкие запросы поможет выявить ресурсоёмкие запросы

Запрос	Число в...	ЦП (мс/...	Число...	Число л...	Число л...	Средня...	Число п...	Баз...
SELECT db.name AS [Name], db.database_id AS [Database ID]	0	0	0	0	0	4228	1	master
SELECT TOP 1 @previous_collection_time = c...	6	0	0	0	0	0	1	tempdb
INSERT INTO #am_wait_stats_snapshots SEL...	6	0	0	0	20	2	1	tempdb
WITH interval_waitstats AS (-- First get resou...	6	0	0	0	6	1	1	tempdb
DELETE FROM #am_wait_stats_snapshots W...	6	0	0	0	2	0	1	tempdb
DELETE FROM #am_dbfileio WHERE collecti...	6	0	0	0	0	0	1	tempdb
SELECT SUM (interval_wait_time_per_sec) ...	6	0	0	0	0	0	1	tempdb
SELECT TOP 1 @previous_collection_time = I...	6	0	0	0	0	0	1	tempdb
DELETE FROM #am_resource_mon_snap WH...	12	0	0	0	0	0	1	tempdb
INSERT INTO #am_request_count_collection	6	0	0	0	0	0	1	tempdb

Рисунок 7 – Вкладка Последние ресурсоёмкие запросы

3. Запустить процедуру «sp_who»:
 - без параметров;
 - с параметром, указывающим на конкретное имя входа;
 - с параметром, указывающим на идентификатор процесса.

Шаги:

1. Открыть SQL Server Management Studio (SSMS).
2. В верхней панели выбрать Создать запрос.
3. Ввести и выполнить команду:

1. Без параметров:

USE master;

EXEC sp_who;

spid	ecid	status	loginame	hostname	blk	dbname	cmd	request_id
1	0	sleeping	sa		0	master	TASK MANAGER	0
2	4	sleeping	sa		0	master	TASK MANAGER	0
3	8	sleeping	sa		0	master	TASK MANAGER	0
4	9	background	sa		0	NULL	XIO_LEASE_RENEWAL_WORKER	0
5	10	background	sa		0	NULL	XIO_RETRY_WORKER	0
6	11	background	sa		0	NULL	XTP_CKPT_AGENT	0
7	12	background	sa		0	NULL	RECOVERY WRITER	0
8	13	background	sa		0	NULL	PVS_PREALLOCATOR	0
9	14	background	sa		0	NULL		0
10	15	background	sa		0	NULL	LAZY WRITER	0
11	16	background	sa		0	NULL	LOG WRITER	0
12	17	background	sa		0	NULL	LOG WRITER	0
13	18	background	sa		0	master	SIGNAL HANDLER	0
14	19	background	sa		0	NULL	LOCK MONITOR	0
15	20	background	sa		0	master	BRKR TASK	0
16	21	background	sa		0	NULL	XE TIMER	0
17	22	background	sa		0	NULL	RESOURCE MONITOR	0
18	23	background	sa		0	NULL	XE DISPATCHER	0
19	24	sleeping	sa		0	master	TASK MANAGER	0
20	25	sleeping	sa		0	master	TASK MANAGER	0
21	27	sleeping	sa		0	master	TASK MANAGER	0
22	29	sleeping	sa		0	master	TASK MANAGER	0
23	30	sleeping	sa		0	master	TASK MANAGER	0
24	31	sleeping	sa		0	master	TASK MANAGER	0
25	34	sleeping	sa		0	master	TASK MANAGER	0
26	41	sleeping	sa		0	master	TASK MANAGER	0
27	44	sleeping	sa		0	master	TASK MANAGER	0
28	46	background	sa		0	NULL	PARALLEL REDO TASK	0
29	47	background	sa		0	NULL	PARALLEL REDO TASK	0

Рисунок 8 – Запрос без параметров

2. С параметром имени пользователя:

USE master;

EXEC sp_who 'sa';

SQLQuery1.sql - US...US108\admin1 (70)* X US108\SQLEXPRESS...нитор активности

```
USE master;
EXEC sp_who 'sa';
```

90 %

	spid	ecid	status	loginame	hostname	blk	dbname	cmd	request_id
1	1	0	sleeping	sa		0	master	TASK MANAGER	0
2	2	0	sleeping	sa		0	master	TASK MANAGER	0
3	3	0	sleeping	sa		0	master	TASK MANAGER	0
4	4	0	sleeping	sa		0	master	TASK MANAGER	0
5	5	0	sleeping	sa		0	master	TASK MANAGER	0
6	6	0	sleeping	sa		0	master	TASK MANAGER	0
7	7	0	sleeping	sa		0	master	TASK MANAGER	0
8	8	0	sleeping	sa		0	master	TASK MANAGER	0
9	9	0	background	sa		0	NULL	XIO_LEASE_RENEWAL_WORKER	0
10	10	0	background	sa		0	NULL	XIO_RETRY_WORKER	0
11	11	0	background	sa		0	NULL	XTP_CKPT_AGENT	0
12	12	0	background	sa		0	NULL	RECOVERY WRITER	0
13	13	0	background	sa		0	NULL	PVS_PREALLOCATOR	0
14	14	0	background	sa		0	NULL		0
15	15	0	background	sa		0	NULL	LAZY WRITER	0
16	16	0	background	sa		0	NULL	LOG WRITER	0
17	17	0	background	sa		0	NULL	LOG WRITER	0
18	18	0	background	sa		0	master	SIGNAL HANDLER	0
19	19	0	background	sa		0	NULL	LOCK MONITOR	0
20	20	0	background	sa		0	master	BRKR TASK	0
21	21	0	background	sa		0	NULL	XE TIMER	0
22	22	0	background	sa		0	NULL	RESOURCE MONITOR	0
23	23	0	background	sa		0	NULL	XE DISPATCHER	0
24	24	0	sleeping	sa		0	master	TASK MANAGER	0
25	25	0	sleeping	sa		0	master	TASK MANAGER	0
26	27	0	sleeping	sa		0	master	TASK MANAGER	0
27	29	0	sleeping	sa		0	master	TASK MANAGER	0
28	30	0	sleeping	sa		0	master	TASK MANAGER	0
29	31	0	sleeping	sa		0	master	TASK MANAGER	0

Запрос успешно выполнен. US108\SQLEXPRESS (16.0 RTM) US108\admin1 (70) master 00:00:00 45 строки

Рисунок 9 – Запрос с параметром имени пользователя

3. С параметром идентификатора процесса:

USE master;

EXEC sp_who '10';

SQLQuery1.sql - US...US108\admin1 (70)* X US108\SQLEXPRESS...нитор активности

```
USE master;
EXEC sp_who '10';
```

90 %

	spid	ecid	status	loginame	hostname	blk	dbname	cmd	request_id
1	10	0	background	sa		0	NULL	XIO_RETRY_WORKER	0

Запрос успешно выполнен. US108\SQLEXPRESS (16.0 RTM) US108\admin1 (70) master 00:00:00 1 строки

Рисунок 10 – Запрос с параметром идентификатора процесса

4. Аналогично выполнить запуск процедуры «sp_who2».

Шаги:

1. Открыть SQL Server Management Studio (SSMS).
2. Создать новый запрос и выполнить команду:

1. Без параметров:
USE master;
EXEC sp_who2;

SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID
1	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:23:47		1	0
2	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		2	0
3	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		3	0
4	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:24:46		4	0
5	sleeping	sa	.	.	master	TASK MANAGER	0	0	01/29 10:30:12		5	0
6	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		6	0
7	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		7	0
8	sleeping	sa	.	.	master	TASK MANAGER	0	0	01/29 10:24:46		8	0
9	BACK...	sa	.	.	NULL	XIO_LEASE_R...	15	0	01/29 08:26:50		9	0
10	BACK...	sa	.	.	NULL	XIO_RETRY_...	0	0	01/29 08:26:50		10	0
11	BACK...	sa	.	.	NULL	XTP_CKPT_AG...	0	0	01/29 08:26:53		11	0
12	BACK...	sa	.	.	NULL	RECOVERY W...	0	0	01/29 08:26:53		12	0
13	BACK...	sa	.	.	NULL	PVS_PREALLO...	0	0	01/29 08:26:53		13	0
14	BACK...	sa	.	.	NULL		0	0	01/29 08:26:53		14	0
15	BACK...	sa	.	.	NULL	LAZY WRITER	15	0	01/29 08:26:53		15	0
16	BACK...	sa	.	.	NULL	LOG WRITER	140	0	01/29 08:26:53		16	0
17	BACK...	sa	.	.	NULL	LOG WRITER	171	0	01/29 08:26:53		17	0
18	BACK...	sa	.	.	master	SIGNAL HAND...	0	0	01/29 08:26:53		18	0
19	BACK...	sa	.	.	NULL	LOCK MONITOR	0	0	01/29 08:26:53		19	0
20	BACK...	sa	.	.	master	BRKR TASK	0	0	01/29 08:26:58		20	0
21	BACK...	sa	.	.	NULL	XE TIMER	0	0	01/29 08:26:54		21	0
22	BACK...	sa	.	.	NULL	RESOURCE M...	125	0	01/29 08:26:54		22	0
23	BACK...	sa	.	.	NULL	XE DISPATCHER	0	0	01/29 08:26:54		23	0
24	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:24:56		24	0
25	sleeping	sa	.	.	master	TASK MANAGER	0	0	01/29 08:26:55		25	0
26	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:42		26	0
27	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:25:11		27	0
28	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:42		28	0

Рисунок 11 – Запрос без

2. С параметром имени пользователя:
USE master;
EXEC sp_who2 'sa';

SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID
1	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:23:47		1	0
2	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		2	0
3	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:30:12		3	0
4	sleeping	sa	.	.	master	TASK MANAGER	0	15	01/29 10:24:46		4	0
5	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:30:12		5	0
6	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:30:12		6	0
7	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:30:12		7	0
8	sleeping	sa	.	.	master	TASK MANAGER	0	5	01/29 10:24:46		8	0
9	BACKGROUND	sa	.	.	NULL	XIO_LEASE_RENEWAL_WORKER	15	0	01/29 08:26:50		9	0
10	BACKGROUND	sa	.	.	NULL	XIO_RETRY_WORKER	0	0	01/29 08:26:50		10	0
11	BACKGROUND	sa	.	.	NULL	XTP_CKPT_AGENT	0	0	01/29 08:26:53		11	0
12	BACKGROUND	sa	.	.	NULL	RECOVERY WRITER	0	0	01/29 08:26:53		12	0
13	BACKGROUND	sa	.	.	NULL	PVS_PREALLOCATOR	0	0	01/29 08:26:53		13	0
14	BACKGROUND	sa	.	.	NULL		0	0	01/29 08:26:53		14	0
15	BACKGROUND	sa	.	.	NULL	LAZY WRITER	15	0	01/29 08:26:53		15	0
16	BACKGROUND	sa	.	.	NULL	LOG WRITER	156	0	01/29 08:26:53		16	0
17	BACKGROUND	sa	.	.	NULL	LOG WRITER	171	0	01/29 08:26:53		17	0
18	BACKGROUND	sa	.	.	master	SIGNAL HANDLER	0	0	01/29 08:26:53		18	0
19	BACKGROUND	sa	.	.	NULL	LOCK MONITOR	0	0	01/29 08:26:53		19	0
20	BACKGROUND	sa	.	.	master	BRKR TASK	0	0	01/29 08:26:58		20	0
21	BACKGROUND	sa	.	.	NULL	XE TIMER	0	0	01/29 08:26:54		21	0
22	BACKGROUND	sa	.	.	NULL	RESOURCE MONITOR	125	0	01/29 08:26:54		22	0
23	BACKGROUND	sa	.	.	NULL	XE DISPATCHER	0	0	01/29 08:26:54		23	0
24	sleeping	sa	.	.	master	TASK MANAGER	0	15	01/29 10:24:56		24	0
25	sleeping	sa	.	.	master	TASK MANAGER	0	0	01/29 08:26:55		25	0
26	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:30:42		26	0
27	sleeping	sa	.	.	master	TASK MANAGER	0	15	01/29 10:25:11		27	0
28	sleeping	sa	.	.	master	TASK MANAGER	0	10	01/29 10:30:42		28	0

Рисунок 12 – Запрос с параметром имени пользователя

3. С параметром идентификатора процесса:
USE master;
EXEC sp_who2 '10';

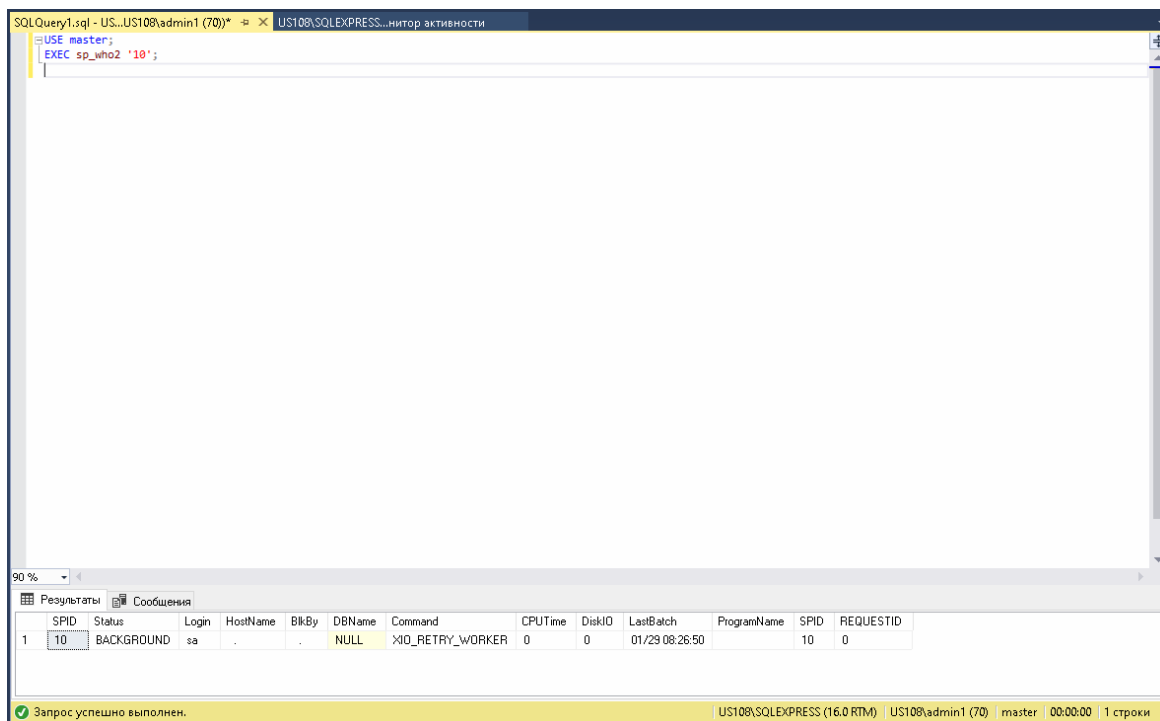


Рисунок 13 – Запрос с параметром идентификатора процесса

5. Ответить на контрольные вопросы.
6. Задokumentировать работу процедур в виде отчета.

Контрольные вопросы

1. Что собой представляет процесс мониторинга, какие его задачи?

Мониторинг – это процесс наблюдения за производительностью сервера. Его задачи:

- Определение возможностей увеличения производительности.
- Оценка активности пользователей.
- Устранение проблем производительности и оптимизация запросов.

2. Какие средства мониторинга рассматриваются в этой работе?

- Activity Monitor (монитор активности)
- Хранимые процедуры `sp_who` и `sp_who2`

3. Какие показатели работы сервера позволяет получать Activity Monitor?

- Время отклика запросов
- Количество активных процессов
- Ожидание ресурсов
- Количество запросов в секунду
- Ввод-вывод данных
- Последние ресурсоемкие запросы

4. Как запускается Activity Monitor?

В SQL Server Management Studio (SSMS):

- Открыть Object Explorer
- Щелкнуть правой кнопкой мыши по серверу
- Выбрать Activity Monitor

5. Какие показатели работы позволяют получать хранимые процедуры «sp_who», «sp_who2»?

- sp_who – информация о пользователях, сеансах и процессах.
- sp_who2 – расширенная информация о процессах, включая использование ЦП, ввод-вывод, блокировки.

6. Какие параметры запуска могут быть использованы для данных процедур?

- Без параметров – отображает все процессы.
- Имя входа (login) – фильтрует процессы по конкретному пользователю.
- Идентификатор сеанса (session ID) – отображает информацию о конкретном процессе.
- ACTIVE – исключает неактивные сеансы.