

# Исследование методов доступа к данным MS SQL Server

Стародубцев М.В.  
магистрант кафедры ЭВМ ДонНТУ  
starodubtsev\_max@ukr.net

## **Аннотация**

*Стародубцев М.В. Исследование методов доступа к данным MS SQL Server. Предложены методы для эффективного доступа к данным БД MS SQL Server. Разработаны рекомендации по применению конкретных провайдеров данных для таблиц различного объёма.*

## **Анотація**

*Стародубцев М.В. Дослідження методів доступу до даних MS SQL Server. Запропоновані методи для ефективного доступу до даних MS SQL Server. Розроблені рекомендації по застосуванню конкретних провайдерів даних для таблиць різного об'єму.*

## **Abstract**

*Starodubtsev M. Research the methods of data access MS SQL Server. Methods for effective data access MS SQL Server are proposed. Recommendations on application of concrete providers for the tables of different volume are developed.*

## **Введение**

За последние 30 лет в области теории систем баз данных был проведен ряд исключительно продуктивных исследований. Полученные результаты полностью можно считать самым существенным достижением информатики за этот период. Базы данных стали основой информационных систем и в корне изменили методы работы многих организаций. В частности, за последние годы развитие технологии баз данных привело к созданию очень мощных и удобных в эксплуатации систем. Благодаря этому системы баз данных стали доступными широкому кругу пользователей. Но, к сожалению, кажущаяся простота таких систем способствовала тому, что пользователи стали самостоятельно создавать базы данных и приложения, не имея достаточных знаний о методах проектирования эффективно работающих систем, что часто приводило к непродуктивным затратам ресурсов и некачественным результатам [4].

Объём данных ежегодно увеличивается с невероятной скоростью. Сейчас очень актуальным является вопрос сохранения видео- и

фотосъёмки из космоса, потому что такие данные требуют вместительных хранилищ данных, а также высокую скорость доступа.

Центральным объектом в исследовании методов доступа к данным MS SQL Server является объектная модель ADO.NET (ActiveX Data Objects for .NET) [5]. ADO.NET вобрала в себя самое лучшее от каждого из методов доступа к данным, которые использовались за последние 20 лет. ADO.NET отличается от предыдущих технологий доступа к данным тем, что она позволяет взаимодействовать с базой данных автономно, при помощи отсоединённого от базы кэша данных [6].

При наличии большого выбора доступных источников данных ADO.NET должна иметь возможность поддерживать множество таких источников. Каждый такой источник данных может иметь свои особенности или набор возможностей. Поэтому ADO.NET поддерживает модель *поставщиков* (provider) [1]. Сервер БД Microsoft SQL Server поддерживает следующие провайдеры данных:

- SQL Client .NET Data Provider;
- OLE DB .NET Data Provider;
- ODBC .NET Data Provider;
- SQL XML .NET Data Provider.

Каждый из вышеперечисленных поставщиков данных имеет свои особенности, преимущества и недостатки. Поэтому для эффективного использования провайдеров .NET, их необходимо глубоко изучить и выявить особенности работы для дальнейшего использования.

Объектная модель ADO.NET была создана с целью предоставления разработчикам и пользователям баз данных возможности легко управлять и взаимодействовать со своим файловым или серверным хранилищем данных [2]. ADO.NET наследует предыдущую технологию доступа к данным, разработанную Microsoft, которая называется классической ADO (ActiveX Data Objects) или просто ADO. Технология ADO.NET призвана помогать в разработке эффективных многоуровневых приложений для работы с БД в локальных сетях и Интернет, для чего она и предоставляет все необходимые средства [5].

В данной статье рассматривается спектр вопросов, связанных с исследованиями провайдеров данных .NET и предлагаются алгоритмы для эффективного доступа к данным в зависимости от объёма таблиц, а также приведены основные результаты исследований и последующие перспективы работы в рассмотренном направлении.

### **Актуальность**

Поскольку MS SQL Server поддерживает несколько провайдеров данных, то возникает закономерный вопрос: какой из них эффективнее или предпочтительнее? Ведь обеспечение высокой скорости доступа к данным

является одной из важнейших характеристик конкретного поставщика данных. Поэтому в данной работе проведены различные эксперименты над поставщиками данных SQL, OLE DB и ODBC с целью выявления их преимуществ и недостатков. Выполненные эксперименты позволяют определить зависимости для производительности и скорости доступа к данным таблиц БД SQL Server.

### ***Постановка задачи***

Целью исследования методов доступа к данным БД Microsoft SQL Server является: определение и обоснование оптимального по скорости доступа к данным БД MS SQL Server провайдера.

Для достижения поставленной цели решаются такие задачи:

1. Изучение характеристик и свойств существующих поставщиков данных БД Microsoft SQL Server.
2. Анализ технологии кэширования данных.
3. Постановка экспериментов для исследования поставщиков данных.
4. Анализ полученных результатов экспериментов и определение зависимостей.
5. Выбор оптимального провайдера данных по заданным критериям.

### ***Обзор состояния исследований и разработок***

Так как база данных Microsoft SQL Server является разработкой компании Microsoft и её исходные коды, алгоритмы и методы функционирования не доступны, то заинтересованность в исследовании этой БД учёными или научными деятелями в данный момент находится на низком уровне. Анализ информации в сети Интернет показал, что исследованием MS SQL Server занимаются лишь разработчики SQL Server и фирмы, которые тесно сотрудничают с компанией Microsoft и широко используют её продукты.

Ведущими исследователями и непосредственными разработчиками методов доступа к данным базы данных Microsoft SQL Server являются:

- Дэвид Сеппа (David Sceppa) [6];
- Сахил Малик (Sahil Malik) [5];
- Шон Вилдермут (Shawn Wildermuth) [3];
- Билл Гамильтон (Bill Hamilton) [1];
- группа разработчиков технологий Microsoft.

## **Методика использования провайдеров для доступа к данным БД**

Важным аспектом является технология кэширования данных, которую поддерживает MS SQL Server [3]. Для увеличения производительности доступа к данным предлагается использовать пул соединений [6]. Ещё одним важным компонентом при работе с таблицами большого объёма выступает файл подкачки ОС Microsoft Windows. Рекомендуется устанавливать размер файла подкачки не меньшим размера таблиц (в байтах), к которым выполняются запросы на считывание данных.

Исходя из вышерассмотренных технологий, предлагается рекомендация: время считывания данных можно уменьшить за счёт увеличения оперативной памяти и характеристик жёсткого диска, а именно: время считывания/записи и время позиционирования головки.

### **Результаты работы**

Для исследования провайдеров данных были разработаны три группы тестов:

1. Выбор первых записей из таблицы БД.
2. Выбор записей из середины таблицы.
3. Выбор последних записей таблицы.

Каждая группа состоит из семи тестов: выбор одной записи, выбор 10 записей, выбор 100 записей, выбор 1`000 записей, выбор 10`000 записей, выбор 100`000 записей и выбор 1`000`000 записей.

Тестовая таблица, из которой запрашивались данные, состоит из 10 столбцов и содержит 2`000`000 записей.

Результаты выполнения первой группы тестов показаны в таблице 1.

Из результатов видно, что лучшую производительность показал провайдер данных SQL Client .NET Data Provider, а ODBC .NET Data Provider оказался наименее производительным. Данные результаты свидетельствуют о том, что поставщики данных OLE DB и ODBC затрачивают дополнительное время на обращение к драйверам, т.е. существует промежуточный уровень между базой данных и поставщиком, который выполняет специальные действия для запроса данных из конкретной БД (в данном случае SQL Server). Для провайдера SQL Client .NET этот промежуточный уровень отсутствует, т.к. сервер БД SQL Server полностью поддерживает архитектуру .NET и промежуточных преобразований нет, т.к. всё одной общей среде выполнения CLR (Common Language Runtime).

Таблица 1 – Результаты выбора первых записей из таблицы

Тип теста	SQL Client .NET Data Provider, с	OLE DB .NET Data Provider, с	ODBC .NET Data Provider, с
Выбор 1 записи.	0.4027814	0.4688808	0.5809055
Выбор 10 записей.	0.410658	0.4841793	0.6147556
Выбор 100 записей.	0.7965911	0.917433	0.9746986
Выбор 1`000 записей.	0.960248	1.039614	1.097547
Выбор 10`000 записей.	1.257502	1.3365823	1.4806395
Выбор 100`000 записей.	6.6791592	7.6805506	8.658796
Выбор 1`000`000 записей.	66.4122658	68.7826164	71.4822362

Следующая группа тестов состоит в выборе записей из середины таблицы. Результаты этих тестов представлены в таблице 2.

Таблица 2 – Результаты выбора записей из середины таблицы БД

Тип теста	SQL Client .NET Data Provider, с	OLE DB .NET Data Provider, с	ODBC .NET Data Provider, с
Выбор 1 записи.	10.9998519	11.4846866	11.156399
Выбор 10 записей.	10.9965524	10.969799	11.0450555
Выбор 100 записей.	11.2381832	11.0682824	10.925667
Выбор 1`000 записей.	10.9855707	11.1454616	10.9573699
Выбор 10`000 записей.	11.4565701	11.2906586	11.6329146
Выбор 100`000 записей.	16.8759512	18.5506053	18.7758867
Выбор 1`000`000 записей.	74.7399406	75.4274115	78.5842456

Некоторые результаты тестов данной группы отличаются от предыдущих в десятки раз. Это связано со спецификой выбора данных из таблицы. Для того чтобы считать из таблицы, например, 100 записей,

начиная с позиции 1`000`000, нужно последовательно с начала таблицы посчитать количество считанных строк. Для выбора данных не с начала таблицы нужно использовать вложенный SQL-запрос, поэтому время на считывание таких данных возрастает.

Из таблицы 2 заметно, что при выборе записей в количестве до 100`000 заметной разницы между тремя провайдерами нет. Та небольшая разница в результатах объясняется не скоростью работы поставщика данных, а скоростью работы жёсткого диска, на котором расположена таблица БД. При считывании данные начинают кэшироваться, и скорость работы провайдера отодвигается на второй план. Поэтому результаты работы второй группы тестов отличаются в среднем на 10-11 секунд – это то время, которое необходимо для выполнения внешнего SQL-запроса для подсчёта 1`000`000 строк.

Третья группа тестов отличается от второй только временем работы тестов, так как в данных тестах нужно подсчитать 2`000`000 строк.

Единственное отличие наблюдается при считывании большого количества записей (свыше 10`000). В процентном соотношении результаты работы третьей группы тестов в данном случае лучше, т.е. время выбора данных меньше. Это связано опять же с технологией кэширования, а именно: на тот момент, когда начинает выполняться непосредственно SQL-запрос на выбор данных из таблицы, большинство необходимых записей уже находится в кэше, поэтому считывание выполняется быстрее. Результаты тестов изображены в таблице 3.

Таблица 3 – Результаты выбора последних записей из таблицы БД

Тип теста	SQL Client .NET Data Provider, с	OLE DB .NET Data Provider, с	ODBC .NET Data Provider, с
Выбор 1 записи.	40.5702849	41.9546864	40.1792735
Выбор 10 записей.	39.627378	40.7491177	39.9562032
Выбор 100 записей.	39.7177483	40.3006241	39.8904827
Выбор 1`000 записей.	39.9782116	39.8816503	40.7431343
Выбор 10`000 записей.	40.3657874	41.4336906	40.8587089
Выбор 100`000 записей.	44.0015313	44.5750814	45.7876544
Выбор 1`000`000 записей.	83.2206237	94.806147	89.2265618

## **Выводы и перспективные направления работы**

Проанализировав результаты работы разработанных тестов можно сделать следующие выводы:

1. Время считывания записей из таблиц баз данных сильно зависит от объёма оперативной памяти, установленной на компьютере.
2. При большом количестве считываемых из таблиц БД данных важную роль играет файл подкачки Microsoft Windows (если на обработку запроса не хватает оперативной памяти).
3. При работе с небольшими таблицами (до 1`000 записей) особенной разницы в скорости доступа к данным Microsoft SQL Server нет. Поэтому в данном случае лучше использовать провайдеры данных OLE DB .NET Data Provider или ODBC .NET Data Provider, потому что эти поставщики данных могут работать не только с БД SQL Server, а также с любыми другими OLE DB- и ODBC-совместимыми источниками данных, что повышает масштабируемость приложений.
4. Если таблицы баз данных содержат свыше 10`000 записей, то провайдер SQL Client .NET Data Provider показывает лучшие результаты по скорости доступа к данным MS SQL Server, в отличие от поставщиков OLE DB и ODBC. Это свидетельствует о направленности сервера БД Microsoft на объёмные корпоративные базы данных с большим количеством таблиц и данных. Поэтому с такими таблицами эффективно работать, используя “родной” для SQL Server провайдер – SQL Client .NET.
5. Ещё один эффективный способ доступа к данным MS SQL Server заключается в предварительных запросах к БД. Даже один предварительный запрос одной записи из таблицы уменьшает время считывания данных следующего запроса к таблице БД до десятков раз. При первом запросе MS SQL Server устанавливает физическое подключение к БД и сохраняет его в пуле соединений. Поэтому последующие запросы к БД выполняются уже с существующими подключениями. Но это правило не распространяется на разные таблицы. К разным таблицам, а тем более, к разным БД создаются свои собственные подключения. Итак, рекомендуется предварительно выполнять какие-нибудь простые запросы, например `SELECT TOP 1 <ColumnName> FROM <TableName>`, к БД, чтобы настроить SQL Server на оптимальное быстроедействие.
6. При выполнении одинаковых запросов к таблице базы данных рекомендуется выполнять 4-5 предварительных запросов с целью увеличения быстрогодействия Microsoft SQL Server. После 4-5 итераций считывание данных выполняется быстрее всего, но это

верно для небольших таблиц (до 1`000 записей). Для больших таблиц нужно выполнять только один предварительный запрос для создания подключения.

Последующее исследование методов доступа к данным MS SQL Server будет состоять в анализе отсоединённых объектов, а именно объекта DataSet. Этот объект позволяет работать с базой данных в отключённом режиме, то есть все необходимые данные загружаются в объект, имеющий структуру, аналогичную с таблицей БД, и работа с записями выполняется автономно, а значит, без удержания соединения с БД, что позволяет другим пользователям работать с таблицами.

### ***Литература***

1. Hamilton, Bill. ADO.NET 3.5 Cookbook, 2nd Edition, Updated for .NET 3.5, LINQ, and SQL Server 2008. O'Reilly, March 15, 2008. Pages: 980. ISBN-13: 978-0-596-10140-4.
2. Браст Эндрю Дж., Форте Стивен. Разработка приложений на основе Microsoft® SQL Server™ 2005. Мастер-класс. / Пер. с англ. – М.: Издательство «Русская Редакция», 2007. – 880 стр.: ил.
3. Вилдермьюс, Шон. Практическое использование ADO.NET. Доступ к данным в Internet. : Пер. с англ. – М. : Издательский дом “Вильямс”, 2003. – 288 с. : ил. – Парал. тит. англ.
4. Коннолли, Томас, Бегг, Каролин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. – М. : Издательский дом “Вильямс”, 2003. – 1440 с. : ил. – Парал. тит. англ.
5. Малик, Сахил. Microsoft® ADO.NET 2.0 для профессионалов.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2006. – 560 с.: ил. – Парал. тит. англ.
6. Сеппа Д. Программирование на Microsoft® ADO.NET 2.0 Мастер-класс. / Пер. с англ. – М.: Издательство «Русская Редакция»; СПб.: Питер, 2007. – 784 стр.: ил.



### **Стародубцев Максим Владимирович.**

В 2007 году получил диплом бакалавра по специальности “Компьютерная инженерия” в Донецком Национальном техническом университете. В 2007 году поступил в магистратуру кафедры ЭВМ, ДонНТУ по специальности “Системное программирование”.  
Научные интересы: разработка Web-приложений.

---

Дата надходження до редакції 30.10.2008 р.