

НОВЫЕ ВОЗМОЖНОСТИ МАКРОСОВ ДЛЯ РАЗРАБОТКИ ЭКОНОМИЧЕСКИХ ПРИЛОЖЕНИЙ В MS OFFICE ACCESS 2010

Для автоматизации выполнения типовых задач, создания сложной логики приложений и связи различных объектов необходимо использовать средства программирования. Простейшим языком программирования, позволяющим расширять функциональные возможности приложения, является язык макросов. Новый конструктор макросов в новой версии MS Office Access 2010 с ясным и понятным интерфейсом, поддерживающий функции IntelliSense, позволяет упростить процесс программирования экономических приложений базы данных.

Использование макросов для автоматизации управления реакцией приложения на действия пользователя позволяет создавать полноценные интерактивные приложения без написания программ на VBA. Макросы обеспечивают выполнение части команд, доступных в VBA, и для большинства пользователей создание макроса оказывается проще, чем использование VBA.

Программы на языке макросов реализуют алгоритмы решения отдельных задач приложения. В макросах можно выполнить различные действия над объектами Access и их элементами, например, открыть форму, отчет, выполнить запрос, применить фильтр, создать свое меню, выполнить любую встроенную команду Access. Имеющийся в Access 2010 набор макрокоманд реализует практически любые действия, которые необходимы для решения задач.

Макрос может быть представлен как отдельный объект – программа и может быть связан с любым событием в форме, отчете или элементе управления.

Возможно решение задач с помощью ряда взаимосвязанных макросов, первый из которых будет запускаться на выполнение из области навигации. При этом пользователь запускает первый макрос на выполнение и далее все управление выполнением задачи осуществляется из макроса. Макрос сам открывает нужные объекты, выбирает и обрабатывает данные, вызывает другие макросы, следуя алгоритму, приводящему к решению задачи. При необходимости из макроса может быть инициирован диалог с пользователем. Для перехода по различным ветвям макроса используется блок управления **Если** (If). Для организации циклов используется блок **ДляКаждойЗаписи**.

Связь макросов с событиями в форме, отчете или элементе управления позволяет пользователю управлять приложением в интерактивном режиме, так как события наступают при выполнении различных действий пользователем с объектами, прежде всего с формами. Примерами событий являются: изменение данных в поле, открытие или закрытие формы или отчета, нажатие кнопки в форме и просто передача фокуса от одного поля к другому. Механизм связывания макросов с событиями в объектах позволяет объединить разрозненные задачи приложения в единый комплекс, управляемый пользователем.

Макрос может быть внедрен в форму или отчет. Внедренный макрос всегда связывается с событием. При этом он становится компонентом формы или отчета. Сохранение внедренных макросов вместе с формами и отчетами упрощает управление объектами приложения.

При программировании в приложении получить доступ к данным базы можно только через формы и отчеты.

Наряду с кардинальной модернизацией конструктора макросов значительным новшеством в версии Access 2010 являются макросы данных. Они предназначены для добавления логики к данным в исходных таблицах. Макросы данных связываются с событиями в таблице базы и выполняются при добавлении, изменении или удалении записи. Такие макросы позволяют проверять данные или выполнять вычисления. В макросах данных, в отличие от макросов, привязанных к событиям в объектах приложения, для доступа к данным используются ссылки на поля таблиц. Обращение к данным через поля объектов приложения не допустимо.

Привязывая логику к таблицам, макросы данных обеспечивают доступ к ним из объектов приложения, а при использовании базы в сети и из различных приложений. Выполнение операций с записями в формах, построенных на базовых таблицах, инициирует выполнение макросов данных. Реализация в макросах данных циклов обеспечивает простоту решения многих задач бизнес-логики.

Макросы данных, по сути, позволяют реализовать в базе данных Access 2010 функции, которые в серверных базах данных реализуются с помощью особого вида хранимых процедур – триггеров. Триггер также связывается с определенной операцией в базовой таблице и каждый раз при выполнении этой операции вызывается. С помощью макросов данных, как и триггеров, могут выполняться действия по обеспечению реляционной целостности данных, не поддерживаемые стандартными средствами. Макрос данных также не имеет ни параметров, ни возвращаемого значения.

Для создания макроса данных используются команды ленты панели инструментов **Работа с таблицами/Таблица** (рис. 1), доступной при открытой таблице. На ленте отображаются кнопки создания макросов данных **Перед изменением** и др.

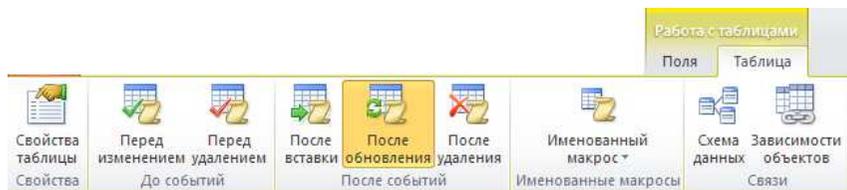


Рис. 1. Команды работы с макросами данных на вкладке **Таблица**

Для каждой таблицы может быть создано по одному макросу данных на каждое из представленных на ленте событий. Макросы данных внедряются в таблицу. На рис. 2 показано окно создания макроса данных таблицы ТОВАР для события **После удаления**.

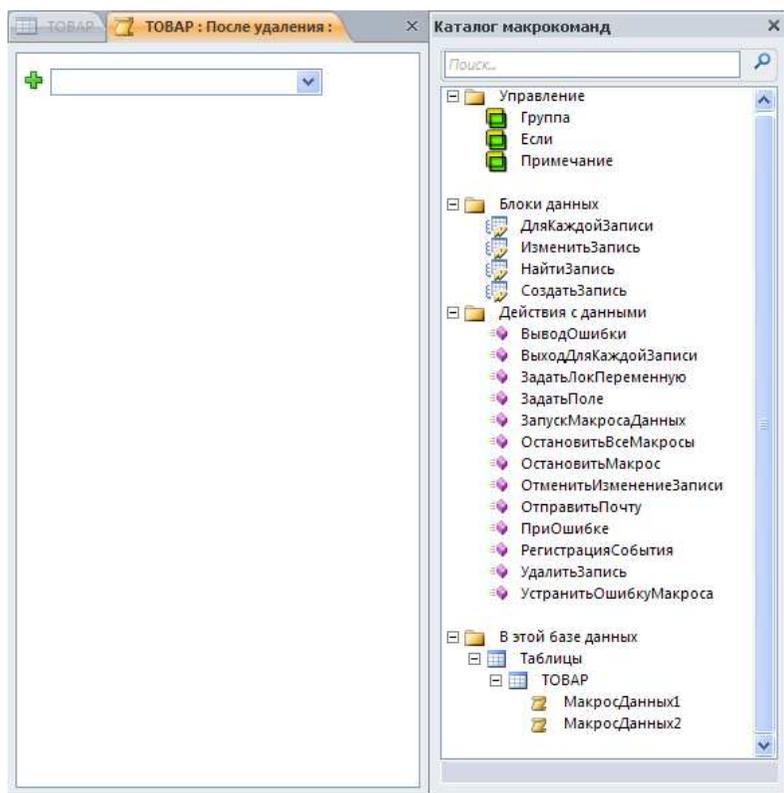


Рис. 2. Окно макроса данных с каталогом макрокоманд

В приведенном окне представлены доступные в макросе данных макрокоманды. Макрокоманда блока управления **Если** позволяет организовать выполнение различных ветвей макроса. Макрокоманда блоков данных **ДляКаждойЗаписи** позволяет организовать цикл для набора данных. Макрокоманды блока **Действия с данными** предназначены для выполнения вычислений, операций с записями, действий с макросами. В разделе каталога макрокоманд **В этой базе данных** отображаются макросы данных, созданные для таблиц.

Кроме макросов данных, связанных с отображенными на вкладке событиями, могут быть созданы именованные макросы данных. В именованном макросе могут определяться параметры. Значения параметрам присваиваются в вызывающем макросе.

Запуск именованного макроса выполняется макрокомандой **ЗапускМакросаДанных**, доступной как из макроса данных, так и из любых других макросов приложения. Например, работая в форме, можно не выполнять вычисления через обращения к полям формы, а вызвать именованный макрос данных, который выполнит вычисления непосредственно в таблицах. Изменения в таблицах тут же отобразятся и в форме.

Макросы данных и именованные макросы отображаются в окне диспетчера макросов данных для каждой из таблиц (рис. 3).

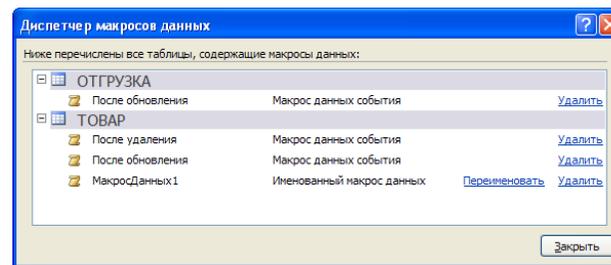


Рис. 3. Список таблиц открытой базы, содержащих макросы данных

При разработке практических приложений часто необходимо при изменениях в одних документах решать задачи, позволяющие актуализировать текущие данные в других. Проиллюстрируем возможности реализации подобных задач примерами.

Рассмотрим создание макроса данных для таблицы ТОВАР, в котором после обновления записи о товаре проверяется изменение цены товара и производится перерасчет стоимости отгруженного товара в таблице ОТГРУЗКА.

Для открытой таблицы ТОВАР на ленте **Работа с таблицами/Таблица** нужно выполнить команду **После обновления**. В открывшееся окно макроса данных вводятся макрокоманды, которые надо выбрать в каталоге (рис. 2), и заполняются аргументы, как показано на рис. 4.

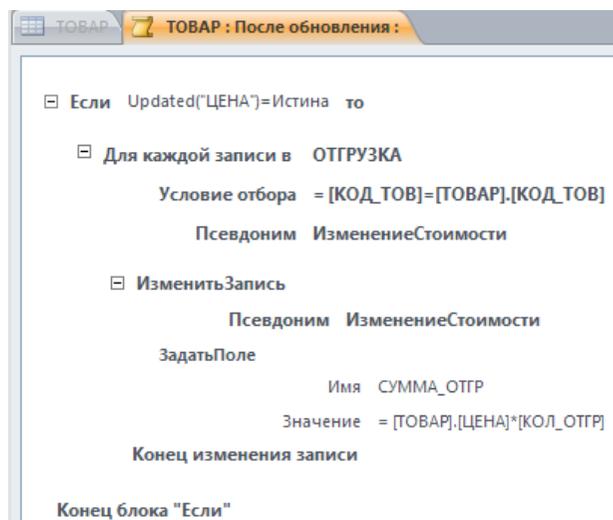


Рис. 4. Макрос данных для обновления поля в записях таблицы ОТГРУЗКА

В блоке **Если** в логическом выражении условия с помощью функции *Updated* («имя поля») проверяется, изменялось ли значение в поле ЦЕНА. Если цена изменялась, макрокомандой **Для каждой записи в** организуется цикл. В нем для каждой записи таблицы ОТГРУЗКА, содержащей значение кода товара из обновленной записи таблицы ТОВАР, макрокомандой **ИзменитьЗапись** рассчитывается новое значение в поле СУММА_ОТГР.

Заметим, макрокоманда **ИзменитьЗапись** должна быть вложена в макрокоманду **Для каждой записи в**, иначе не будет найден псевдоним текущей записи цикла. Макрокоманда **ЗадатьПоле**, в свою очередь, должна быть вложена в **ИзменитьЗапись**.

Рассмотрим создание именованного макроса, который будут сохранять сведения об обновлениях цены товара в некоторой таблице, например ИзмененныеЗаписи.

Пусть в таблице ИзмененныеЗаписи содержатся поля – код товара, старая цена, новая цена и ключ с типом данных **Счетчик**. Использование такого ключа позволит включать в таблицу неоднократные изменения записи с одним и тем же кодом товара.

Для открытой таблицы ТОВАР надо создать именованный макрос, выбрав соответствующую команду на вкладке **Работа с таблицами/Таблица**. В макросе для получения из вызывающего макроса значений кода, старой и новой цены товара надо создать соответствующие параметры (рис. 5). Далее выбрать макрокоманду **Создать запись в** с аргументом ИзмененныеЗаписи и три макрокоманды **ЗадатьПоле** для присвоения значений полям таблицы ИзмененныеЗаписи.

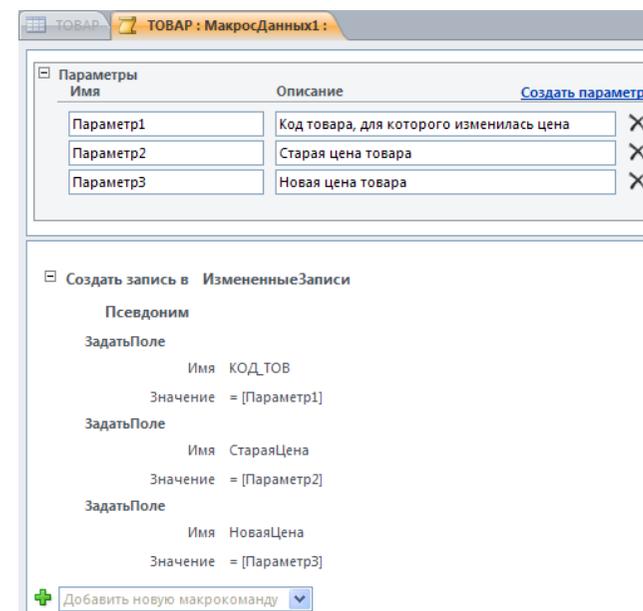


Рис. 5. Именованный макрос данных с параметрами

По умолчанию созданный макрос получит имя ТОВАР.МакросДанных1.

Для вызова именованного макроса при изменении цены в таблице ТОВАР надо дополнить ранее созданный для нее макрос данных макрокомандой **ЗапускМакросаДанных** (рис. 6).

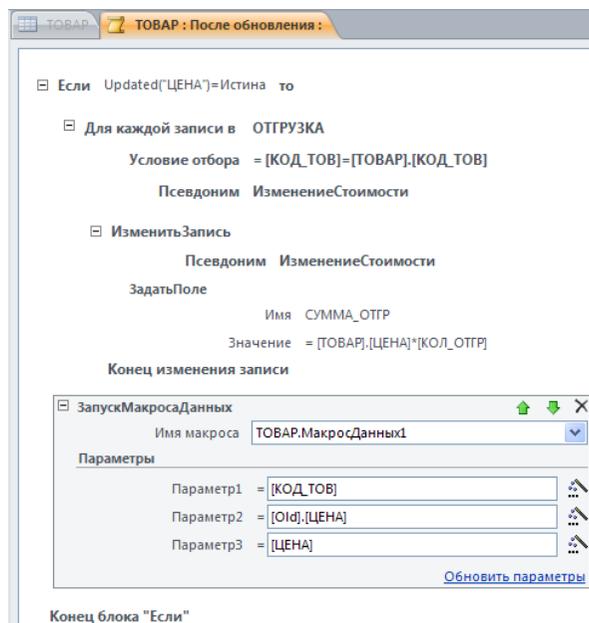


Рис. 6. Вызов именованного макроса данных и передача ему параметров

Для вывода в макросе строки параметров нужно использовать **Обновить параметры** (рис. 6). Параметрам надо присвоить значения из обновленной записи таблицы ТОВАР, а для получения значения старой цены использовать функцию *[Old] [ЦЕНА]*.

Теперь при изменении цены одного из товаров в таблице ТОВАР в таблице ОТГРУЗКА во всех записях с соответствующим кодом товара сумма отгрузки (СУММА_ОТПР) пересчитается, а в таблицу ИзмененныеЗаписи будет внесена новая запись. Обновления в таблицы вносятся вне зависимости от того, открыты они или закрыты.

Рассмотрим возможность обновления текущих остатков товара на складе при отгрузке товара по накладной. В базе данных остаток товара размещается в поле таблицы ТОВАР, поскольку товар хранится только на одном складе.

Для решения этой задачи можно создать макрос данных **ПослеВставки** для таблицы ОТГРУЗКА, который будет выполняться при добавлении новых записей в нее. В макросе (рис. 7) будем использовать макрокоманду

Найти запись в с аргументом ТОВАР и для найденной записи макрокоманду **ИзменитьЗапись**. В рамках последней с помощью макрокоманды **ЗадатьПоле** надо присвоить полю Остаток новое значение, уменьшенное на количество отгруженного товара.

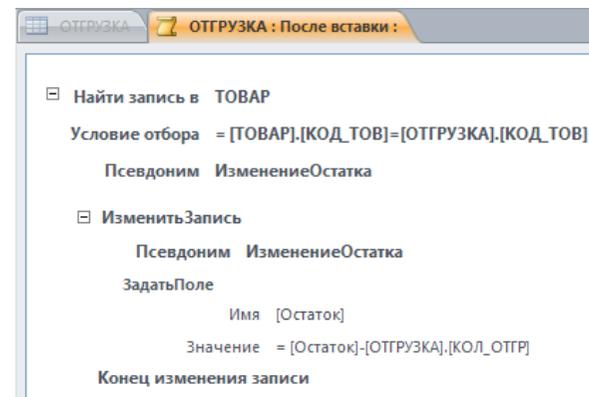


Рис. 7. Макрос данных после вставки записи в таблицу ОТГРУЗКА

Рассмотрим возможность расчета суммарного количества товара, отгруженного по договору, при добавлении новой записи об отгрузке товара. Такой расчет позволит поддерживать в базе данные о выполнении плана отгрузки по договорам в актуальном состоянии. Ввод данных по отгрузке может осуществляться непосредственно в таблицу ОТГРУЗКА или через форму.

Количество товара, которое должно быть отгружено покупателю в соответствии с договором в конкретном месяце, хранится в записях таблицы ПОСТАВКА_ПЛАН. Для сохранения общего количества товара, отгруженного по договору в указанный срок, можно предусмотреть в этой таблице дополнительное поле ВыполнениеПлана.

Макрос для решения этой задачи, так же как и при расчете остатка товара на складе, должен выполняться при наступлении действия **ПослеВставки**. Поиск в таблице ПОСТАВКА_ПЛАН, требующей обновления записи, может быть вынесен в именованный макрос. Вызов именованного макроса и подготовка необходимых для него параметров выполняются в существующем для таблицы ОТГРУЗКА макросе **ПослеВставки**.

На рисунке 8 представлен именованный макрос таблицы ОТГРУЗКА МакросДанных1, в котором определены параметры, необходимые для поиска записи и обновления значения в поле ВыполнениеПлана таблицы ПОСТАВКА_ПЛАН. Поля номер договора, месяц отгрузки и код товара составляют ключ этой таблицы. Макрокоманда **Найти запись в** таблице ПОСТАВКА_ПЛАН возвращает запись, соответствующую заданному условию отбора:

```
( [ПОСТАВКА_ПЛАН].[НОМ_ДОГ]=Параметр4 And
[ПОСТАВКА_ПЛАН].[СРОК_ПОСТ]=[Параметр3] And
[ПОСТАВКА_ПЛАН].[КОД_ТОВ]=[Параметр1] )
```

Макрокоманда **ИзменитьЗапись**, входящая в блок макрокоманды **Найти запись в**, ссылаясь на псевдоним найденной записи, выполняет макрокоманду **ЗадатьПоле**, которая прибавляет указанное в записи об отгрузке количество к имеющемуся в поле ВыполнениеПлана. Заметим, чтобы это вычисление могло быть выполнено при первой отгрузке товара, полю ВыполнениеПлана по умолчанию должно присваиваться нулевое значение.

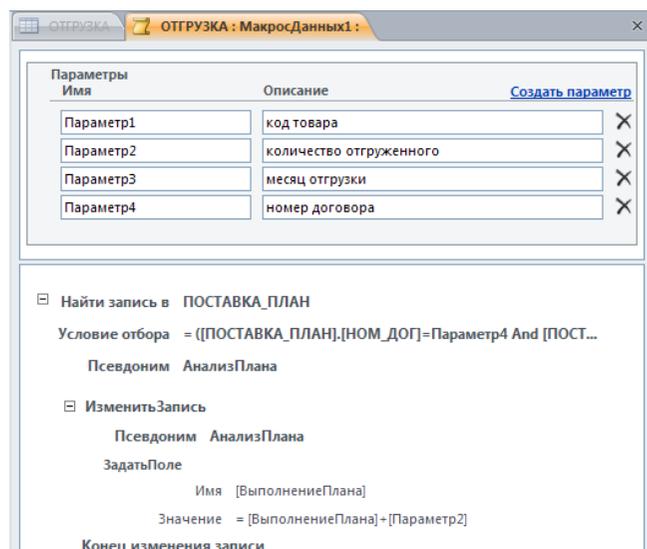


Рис. 8. Именованный макрос поиска и обновления записи в таблице ПОСТАВКА_ПЛАН

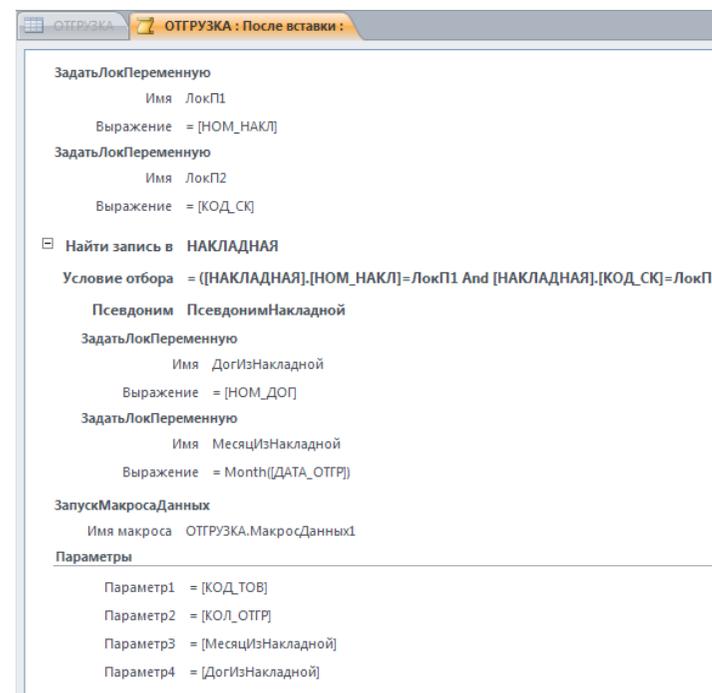


Рис. 9. Вызов именованного макроса с параметрами

Для вызова именованного макроса и подготовки значений, необходимых в именованном макросе параметров, надо дополнить ранее созданный макрос **ОТГРУЗКА.ПослеВставки** (см. рис. 7). Значения параметров Код товара и количество отгруженного выбираются из вставленной новой записи таблицы ОТГРУЗКА, а два недостающих – номер договора и месяц отгрузки – надо выбрать из таблицы НАКЛАДНАЯ, являющейся главной по отношению к таблице ОТГРУЗКА и содержащей общие сведения об отгрузке по документу. Чтобы найти запись в накладной, нужно задать значения его ключевых полей, номер накладной и код склада. Значения этих полей имеются в добавленной в таблицу ОТГРУЗКА записи, и их надо использовать в условии отбора записи из таблицы Накладная.

Рассмотренные примеры наглядно иллюстрируют новые возможности использования макросов при программировании экономических приложений в MS Office Access 2010.