

Scientific journal
PHYSICAL AND MATHEMATICAL EDUCATION
Has been issued since 2013.

ISSN 2413-158X (online)
ISSN 2413-1571 (print)

Науковий журнал
ФІЗИКО-МАТЕМАТИЧНА ОСВІТА
Видається з 2013.



<http://fmo-journal.fizmatsspu.sumy.ua/>

Шамшина Н.В. Вивчення макросів даних в MICROSOFT ACCESS // Фізико-математична освіта : науковий журнал. – 2017. – Випуск 2(12). – С. 165-169.

Shamshina Natalia. The Study Of Data Macros In Microsoft Access // Physical and Mathematical Education : scientific journal. – 2017. – Issue 2(12). – P. 165-169.

УДК 004.65+372.8

Н.В. Шамшина

*Сумський державний педагогічний університет імені А.С.Макаренка, Україна
shamichek@ukr.net*

ВИВЧЕННЯ МАКРОСІВ ДАНИХ В MICROSOFT ACCESS

Анотація. Стаття присвячена макросам даних в останніх версіях ACCESS, яка є однією з найпопулярніших систем управління базами даних. Автор розглядає поняття макросів даних, їх застосовування для забезпечення цілісності даних і реалізації складної бізнес-логіки при моделюванні конкретної предметної області. Макроси даних – нова функція ACCESS 2010, це нова можливість вбудовувати логіку в події, що відбуваються в таблицях за додавання, оновлення та видалення даних. Автор описує типи і підтипи макросів даних, їх призначення, правила створення і застосування, детально аналізує їх особливості. Створення макросів даних для окремих випадків проілюстровано прикладами, в яких автор пояснює призначення та правила використання окремих блоків операцій Конструктора макросів: для організації циклу, розгалуження, для пошуку запису та ін. Автор доступно і детально пояснює переваги та необхідність використання макросів даних, їх відмінності від макросів інтерфейсу та ізольованих макросів. Матеріал статті містить роз'яснення та методичні рекомендації для вивчення теми «Системи Управління Базами Даних» з дисципліни Інформаційні технології.

Ключові слова: Access, макроси даних, обмеження цілісності, бізнес-логіка.

Постановка проблеми. В наш час, незважаючи на істотні зміни навчальної програми, вивченню систем управління базами даних в курсі інформатики приділяється нітрохи не менше уваги, ніж раніше. Це пояснюється необхідністю освоєння навичок роботи в сучасних автоматизованих системах обліку і контролю даних. Однією з найбільш доступних і зручних для вивчення в шкільному курсі є Система Управління Базами Даних (СУБД) Access з пакету Microsoft Office. Розробка нової версії програми Access, її вдосконалення призвело до нових можливостей для створення макросів різних видів, у тому числі і макросів даних, що відповідають за логіку збереження даних. Макроси даних є необхідними для реалізації бізнес-логіки при моделюванні конкретної предметної області і створенні ефективних сучасних баз даних.

Аналіз актуальних досліджень. Аналіз науково-методичних джерел щодо методичних рекомендацій при вивченні макросів даних в СУБД Access, дозволяє стверджувати, що методика вивчення макросів даних на сьогоднішній день в Україні практично не розроблена. Найбільш доступним джерелом інформації є довідка з сайту розробника програмного продукту [1-3], але її недостатньо для розуміння початківця щоб опанувати створення макросів даних, до того ж довідка, як правило, є автоматичним перекладом англійського джерела. Більша кількість посилань, при пошуку по мережі, – російськомовні. Приділяють увагу вивченню макросів даних в СУБД Access викладачі економічних вищих навчальних закладів (Бекаревич Ю., Пушкіна Н., Самоучитель Access 2013, електронна книга-комерційний проект). Велика кількість форумів для зацікавлених, де дискутують про помилки при створенні макросів даних, надають поради. Все це підтверджує актуальність даного питання для сучасної освіти.

Мета статті – описати різні типи макросів даних в СУБД Access, їх призначення і правила створення, надати роз'яснення та методичні рекомендації для вивчення теми «Системи Управління Базами Даних» з дисципліни Інформаційні технології.

Виклад основного матеріалу. Одним з істотних недоліків попередніх версій СУБД Access була відсутність процедурної підтримки обмежень цілісності і неможливість створення і використання тригерів. Тригер – це SQL-оператор, який активізується при виконанні операцій з об'єктами бази даних. Тригери реалізують бізнес-логіку зміни даних і призначені для підтримки цілісності бази даних. Цілісність бази даних без реалізації логіки є лише формальною властивістю, урахуванням стандартного універсального набору обмежень цілісності, тоді як бізнес-логіка враховує всі залежності поведінки об'єктів предметної області, яку система моделює [5].

Макроси даних, які вперше з'явилися в Microsoft Access 2010, дозволяють вбудовувати бізнес-логіку в події, що відбуваються в таблицях за додавання, оновлення та видалення даних. Можна сказати, що макроси даних подібні до тригерів і виконують ті ж функції, що є необхідними для логічної роботи з даними.

Будь який макрос виконує дії, тобто записану послідовність команд призначених для автоматизації роботи з базою даних. Макроси даних автоматизують операції, які необхідні при зміні змісту бази даних, ці операції виконуються при здійсненні транзакції.

Макроси даних дозволяють реалізовувати наступні сценарії:

- перевірка значення поля, перед тим як дозволити додавання іншого запису;
- перевірка точності даних у таблиці;
- збереження історії змін запису;
- створення повідомлення електронної пошти при зміні значення поля.

В Access існують також інші типи макросів: ізольовані макроси, макроси користувальницького інтерфейсу. Ізольовані макроси і макроси користувальницького інтерфейсу пов'язані з різними об'єктами інтерфейсу, такими як кнопки, текстові поля, області даних форми і звіту. Цим вони відрізняються від макросів даних, які зв'язуються з таблицями [4].

Головна особливість макросів даних у тому, що вони дозволяють прив'язувати логіку до записів в конкретній таблиці, це призводить до наступних переваг їх використання:

- всі форми і запити для додавання, оновлення та видалення даних в таблиці не потребують окремо прописаних макросів, кількість об'єктів бази даних скорочується;
- макроси даних спрощують програмування забезпечення цілісності даних, так як немає необхідності передбачати всі можливі варіанти здійснення операцій над даними, достатньо передбачити їх безпосередньо в базовій таблиці;
- за допомогою макросів даних можна надійно забезпечити цілісність даних.

Розглянемо, наприклад, ситуацію, коли ізольований макрос відповідає за операції при додаванні запису і спрацьовує у формі, що прив'язана до таблиці без макросу даних. Якщо користувач має права доступу до таблиць або може виконувати запити, він зможе обійти форму, порушивши таким чином логіку. При вбудовуванні логіки безпосередньо в таблицю операції виконуються, навіть якщо користувач вносить зміни без використання форми [2].

Макроси даних не відображаються в розділі *Макроси* в області переходів робочого вікна бази даних. Керування макросами даних здійснюється за допомогою вкладки *Робота с таблицями /Таблиця* під час перегляду таблиці у режимі таблиці або за допомогою вкладки *Робота с таблицями* при модифікації таблиці в режимі Конструктора.

Існує два основні типи макросів даних:

- 1) керовані подіями макроси даних, що викликаються подіями в таблицях, прив'язують логіку до записів;
- 2) іменовані макроси даних, що запускаються шляхом виклику по імені, прив'язують логіку до таблиці.

Розглянемо перший тип макросів даних. Подія в таблиці – це кожна операція додавання, редагування або видалення запису. Можна запрограмувати запуск макросу даних після будь-якої з цих трьох подій або безпосередньо перед подією видалення або зміни.

Розрізняють п'ять підтипів керованих подіями макросів даних: після вставки, після оновлення, після видалення, до видалення, до зміни. Можливі дії, які можна здійснити з допомогою макросу даних для різних подій в таблиці описані нижче (табл. 1).

Як правило, краще викликати подію *До измененія* для зміни даних і виконання додаткового коду, ніж дозволити оновити запис, а вже потім внести додаткові зміни в ту ж запис, яка тільки що була збережена в таблицю. Макрос даних *После обновления* краще використовувати для підведення підсумків і збереження підсумкових значень в інші таблиці [3].

Для створення макросу даних необхідно відкрити таблицю бази даних в режимі перегляду або Конструктора. На вкладках відображаються кнопки створення макросів даних, які будуть виконуватися при наступних подіях: *После вставки*, *После обновления*, *После удаления*, *До удаления*, *До измененія*, а також кнопка *Создать именованный макрос*.

Щоб створити, наприклад, макрос даних, що виконується до видалення запису з таблиці, виберіть подію *До удаления*. При виборі відповідної кнопки для створення макросу даних звертайте увагу на текст знизу – підказку про призначення даного виду макросів. Після натискання кнопки відкриється Конструктор

макросів. Якщо макрос для цієї події вже створений, Конструктор макросів відкриється для редагування макросу. Кнопка зі створеним макросом даних на вкладці відрізняється кольором.

Таблиця 1

Макроси даних, що керовані подіями

Події			Можливі дії
BeforeChange	<i>До изменения</i>	До зміни	Видача повідомлення про помилку і блокування змін Порівняння старого значення з новим і відкриття іншої форми
BeforeDelete	<i>До удаления</i>	До видалення	Видача повідомлення про помилку і видалення Виклик форми для зміни порядку слідування елементів
AfterUpdate AfterInsert AfterDelete	<i>После обновления После вставки После удаления</i>	Після оновлення Після вставки Після видалення	Створення повідомлення електронної пошти Обхід у циклі набору записів і оновлення їх стану

Макроси даних впроваджуються в таблицю. Для кожної таблиці може бути створено по одному макросу даних на кожну з представлених на вкладці подій.

До складу Access 2010-2016 входить новий Конструктор для створення макросів. Кожен макрос складається з однієї або кількох команд, в залежності від ситуації деякі команди можуть бути недоступні. Список макрокоманд для макросу даних відрізняється від пропонованого для інших макросів бази даних. В розділі каталогу макрокоманд *В этой базе данных* відображаються іменовані макроси даних, створені для таблиць.

Розглянемо другий тип макросів даних, який відрізняється більшою універсальністю. Іменовані або «автономні» макроси даних пов'язуються з певною таблицею, але не з подією. Їх може бути декілька, з різними іменами для однієї таблиці. Застосовують іменовані макроси для спільного використання макрокоманд в макросах даних і коді подій.

Перевага іменованого макросу у використанні параметрів. За допомогою параметрів в іменованій макрос передаються значення, які потім можна задіяти в умовних операторах або в розрахунках. Крім того, за допомогою параметрів із звичайних макросів в іменованій макрос даних передаються посилання на об'єкти. Параметри також забезпечують додаткову гнучкість, оскільки дозволяють повторно використовувати один і той же макрос без змін [3].

Для створення параметра в іменованому макросі даних в режимі Конструктора макросів використовують кнопку *Создать параметр*. У полі *Имя* вказують унікальне ім'я параметра. Це ім'я потрібно для вказівки параметра у виразах. Рекомендується також ввести текст опису параметра в полі *Описание*, при використанні параметра текст буде відображатися у вигляді підказки.

При збереженні створеного іменованого макросу вказують його ім'я, за замовчуванням: *МакросДанных1*. У виразах з використанням іменованого макросу даних вказується ім'я таблиці, до якої він прив'язаний: *Таблица1.МакросДанных1*.

По імені іменованій макрос можна викликати з будь-якого іншого макросу даних або звичайного макросу. Для цього необхідно використати макрокоманду *ЗапускМакросаДанных*. Для кожного зі створених параметрів в макрокоманді передбачено поле, в якому можна вказати необхідне значення.

Створені макроси даних можна редагувати, видалити, для іменованого макросу можна змінити ім'я. Керовані подіями макроси даних видалити при роботі з таблицею в режимі Конструктора.

У макросах даних можна використовувати різні структурні елементи програмування, наприклад, розгалуження та цикли. Для цього в режимі Конструктора макросів вибирають наступні блоки команд:

- для вибору одного з декількох можливих варіантів дій в залежності від зазначених умов використовують блок управління *Если...То...Иначе*;
- для організації циклів при перевірці записів у таблиці використовують блок даних *ДляКаждойЗаписи*;
- якщо необхідно виконати операції для одного запису використовують блок даних *НайтиЗапись*;
- для зміни значень поля використовують блок даних *ИзменитьЗапись*, а в ньому *ЗадатьПоле*;
- для створення запису в резервній таблиці використовують блок даних *СоздатьЗапись*, а в ньому *ЗадатьПоле*.

При роботі з блоками даних використовують *Псевдоним* для позначення поточної записи в таблиці, псевдонім задають як внутрішній параметр макрокоманди.

Конструювання макросів даних з використанням блоків команд схоже на структурне програмування і дає можливість отримати та закріпити корисні навички, усвідомити і засвоїти сутність елементів структурного програмування.

Розглянемо приклад створення макросу з використанням розгалуження. Макрос враховує кількість товарів на складі: якщо кількість товару опускається нижче певного рівня, необхідно повідомити персонал про те, що товар закінчується, якщо ще нижче – про необхідність замовлення нової партії товару. У таблиці обліку товарів на складі «склад» створимо макрос даних, що спрацює для події *До измененія*. Для того щоб здійснити перевірку значень використовуємо оператор *Если...То...Иначе*, який буде виконуватися при кожному внесенні змін у таблицю. При значеннях в поле кількості товару менше 100 одиниць, $[кількість] < 100$, у полі [стан] задається значення «ЗАМОВИТИ!». При значеннях від 100 до 500, $[кількість] \geq 100$ And $[кількість] \leq 500$, задається значення "Закінчується!". У всіх інших випадках задається «пропуск». Макрос даних в режимі Конструктора представлено на рис. 1.

Слід зазначити, що подібну логіку також можна додати до таблиці, якщо значення поля «стан» задати як обчислюване з використанням функції *IIf* ():

$IIf([кількість] < 100; "ЗАМОВИТИ!"; IIf([кількість] \geq 100 \text{ And } [кількість] \leq 500; "Закінчується!"; " "))$,

однак, в Конструкторі макросів базова структура розгалуження представлена більш наочно і легше для розуміння. Але, якщо потрібно повідомлення виводити в іншу таблицю для автоматичного замовлення товару, макрос даних є незамінним. Розглянемо на простому прикладі як із першої таблиці, де відбуваються події, перейти до зміни запису в іншій таблиці.

Наступний приклад представляє макрос даних, який при додаванні запису в першу таблицю підбиває підсумки і зберігає підсумкові значення в іншу таблицю. Логіка така: при вставці запису про надходження певного товару в таблиці «Приход», в таблиці «склад» у рядку, що відповідає даному товару, кількість товару має збільшитися. Для цього необхідно створити макрос даних для події *После вставки* в таблиці «Приход» з передачею даних у таблицю «склад». В таблиці «склад» відшукується запис з відповідним товаром, доданим до таблиці «Приход». Для даного запису значення поля [кількість] в таблиці «склад» збільшується на величину значення поля [кільк] в таблиці «Приход»: $[кількість] + [Приход].[кільк]$. Макрос даних в режимі Конструктора представлений на рис. 2.

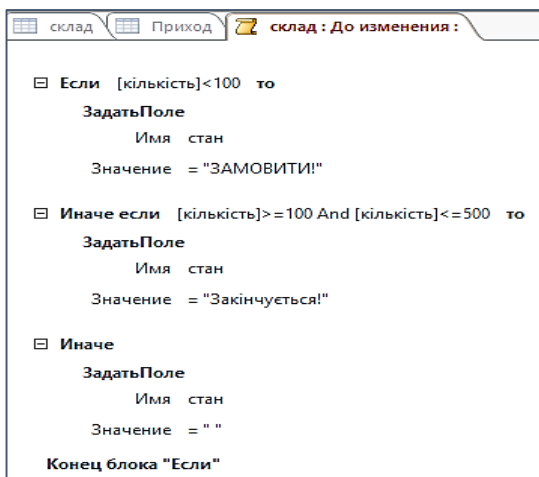


Рис. 1.

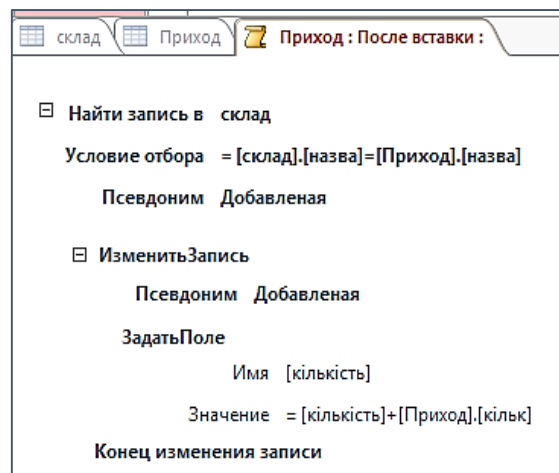


Рис. 2.

У наданому прикладі використано блоки даних *НайтиЗапись*, *ИзменитьЗапись*, *ЗадатьПоле*. Псевдонім *Добавленная* використаний для позначення поточного запису в таблиці. Зверніть увагу, що блок даних *ИзменитьЗапись*, є вкладеним у Конструкторі макросів й записаний з відступом! Це важливо, інакше псевдонім для поточного запису не буде знайдений і макрос не спрацює.

Запис в таблиці «склад» буде змінено незалежно від того відкрита чи ні таблиця «склад». Зміни в таблиці будуть видимі або при відкритті таблиці, або при оновленні. Більш того, макроси спрацюють незалежно від наявності зв'язків із забезпеченням цілісності даних між зазначеними таблицями.

Аналогічний макрос даних можна створити для таблиці «Витрати», яка відображає відпуск товарів зі складу. Тільки при цьому необхідно віднімати кількість товару із значення поля [кількість] в таблиці «склад». Таблицю «Витрати» легко створити шляхом копіювання структури таблиці «Приход», при цьому всі макроси даних, пов'язані з таблицею також копіюються, досить злегка їх відредагувати.

Висновки. Макроси даних зберігаються в таблиці, а не в об'єктах, які використовують таблицю. За допомогою макросів даних можна створити більш ефективну базу даних. Типові ситуації для застосування макросів даних.

У разі бази даних обліку товарів – якщо в таблиці товарних залишків кількість однієї із позицій падає нижче заданого числа, в іншій таблиці автоматично створюється запис для замовлення цієї товарної позиції.

У разі бази даних благодійних пожертв – надсилання повідомлення електронної пошти спонсору з подякою за внесок.

У разі бази даних продажів – автоматичний розрахунок знижки, якщо сума замовлення перевищує задану величину.

У разі бази даних контактів – повідомлення користувачу, якщо заповнені ні всі обов'язкові поля адреси.

При видаленні запису – автоматичне додавання записів в іншу таблицю для ведення журналу аудиту. Для запобігання видалення запису з таблиці при виконанні певної умови.

Наведений опис і приклади макросів даних ілюструють підвищення рівня складності матеріалу при вивченні баз даних. У навчальному плані вивчення макросів даних повинно бути передбачено після вивчення макросів інтерфейсу.

Вивчення макросів даних в темі «Системи управління базами даних» допомагає глибше засвоїти та закріпити навички використання базових структур програмування, розвиває логічне мислення, готує студентів до подальшої професійної діяльності та підвищує їх рівень компетентності.

Список використаних джерел

1. Microsoft Access 2010 Руководство по продукту [Електронный ресурс] – Режим доступа: www.kurskmed.com/itcenter/uploads/989a402.pdf – С. 21-22.
2. Работа с макросами и выражениями в Access 2010 [Електронный ресурс] материал с официального сайта Microsoft, 2010-2015. – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/office/gg435977\(v=office.14\).aspx#odc_ac14_ta_WorkingwithDataMacros_BuildingDataMacros](https://msdn.microsoft.com/ru-ru/library/office/gg435977(v=office.14).aspx#odc_ac14_ta_WorkingwithDataMacros_BuildingDataMacros)
3. Создание макроса данных. Применяется к: Access 2010 [Електронный ресурс] материал сайта Microsoft, перевод – Режим доступа: <https://support.office.com/ru-ru>
4. Шамшина Н.В., Создание интерфейса учебно-методического комплекса с использованием макросов в Access 2010 // Фізико-математична освіта: Зб. наукових праць. – Суми: Вид-во СумДПУ імені А.С.Макаренка, 2014. – № 1 (6). – С. 214-221.
5. Шамшина Н.В. Об особенностях сохранения информации в базах данных // Фізико-математична освіта: науковий журнал. – 2016. – Випуск 4(10). – С. 148-151.

References

1. Microsoft Access 2010 Керівництво по продукту [Електронний ресурс] – Режим доступа: www.kurskmed.com/itcenter/uploads/989a402.pdf – p. 21-22
2. Working with macros and expressions in Access 2010. [Electronic resource] the content from the official Microsoft site, 2010-2015.– Mode of access: [https://msdn.microsoft.com/ru-ru/library/office/gg435977\(v=office.14\).aspx#odc_ac14_ta_WorkingwithDataMacros_BuildingDataMacros](https://msdn.microsoft.com/ru-ru/library/office/gg435977(v=office.14).aspx#odc_ac14_ta_WorkingwithDataMacros_BuildingDataMacros)
3. Creating a Data Macros. Applies to: Access 2010 [Electronic resource] the Microsoft translation – Mode of access: <https://support.office.com/ru-ru>
4. Shamshina N. The creation of interface training complex with the use of macros in Access 2010 // Physico-mathematical education: Sat. scientific papers. – Sumy: Publishing house SumSPU named after A. S. Makarenko, 2014. – № 1 (6). – p. 214-221 (in Ukrainian)
5. Shamshina N.V. About the peculiarities of saving information in databases // Physical and Mathematical Education : scientific journal. – 2016. – Issue 4(10). – P. 148-151. (in Ukrainian)

THE STUDY OF DATA MACROS IN MICROSOFT ACCESS

Natalia Shamshina

Makarenko Sumy State Pedagogical University, Ukraine

Abstract. *The article is devoted to data macros in recent versions of ACCESS, which is one of the most popular database management systems. The author examines the concept of data macros, their application to ensure data integrity and implement complex business logic in modeling a particular subject area. Data macros is a new feature in ACCESS 2010 is a new opportunity to embed the logic in the events occurring in the tables for adding, updating, and deleting data. The author describes the types and subtypes of data macros, their purpose, rules of creation and application, analysing in detail their features. Create macros data for individual cases is illustrated by examples in which the author explains the purpose and rules of application of separate blocks of the operations Constructor macros: loops, branching, record search, etc. the Author is available and explains in detail the advantages and the need to use macro data and their difference from the macro interface and macro isolated. The article contains explanations and methodical recommendations to studying of a subject "database management System" on the discipline of Information technology.*

Key words: *Microsoft Access, Data Macros, integrity constraints, business logic.*