

Практическое занятие

Программирование баз данных

Цель: получить практические навыки программирования баз данных средствами СУБД на примере базы данных *Библиотека*.

Задание:

- I. Освоить программирование с использованием встроенных функций.
- II. Ознакомиться с программированием на языке SQL.
- III. Приобрести практику программирования с использованием языка высокого уровня VBA.

Порядок выполнения

I. Программирование с использованием встроенных функций.

Запустите программу Microsoft Access и откройте базу данных Библиотека.

A. Расширение списка полей по горизонтали

Расширение списка полей - это специальная операция, позволяющая добавить к списку полей запроса любое поле, которого нет в таблицах-источниках.

Вычисление возраста по дате рождения

1. Создать новый запрос в режиме конструктора.
2. Добавить в окно запроса таблицу Читатели и создать в запросе поле Фамилия читателя.
3. Перевести курсор в заголовок второго поля и включить построитель выражений (в контекстном меню пункт Построить или нажать кнопку Построить на панели инструментов).
4. В открывшемся окне Построителя выражений набрать имя поля и через двоеточье выражение для его вычисления:

Возраст: Year(Date())-Year(Читатели![Дата рождения])

Для построения выражения воспользоваться нижними тремя полями «элементы выражений». В левом поле выбрать папку Функции- Встроенные функции. В среднем поле выбрать Дата/время и наконец в левом нужную функцию. Выбрав нажмите кнопку «Вставить».

Выражение *Читатели![Дата рождения]* можно найти в папке Таблицы и вставить в выражение нажав кнопку Вставить.

После набора всего выражения нажать кнопку ОК.

5. В окне Возраст: запрос на выборку проверить наличие вновь созданного поля и сохранить запрос с именем Возраст.
6. Выполнить созданный запрос и проверить его содержимое.

B. Вычисления по вертикали

Вычисления по вертикали – это такая возможность конструктора запросов, которая использует операции группирования.

Вычисления количества книг, находящихся на руках у курсантов

1. Создать новый запрос в режиме конструктора.
2. Добавить в окно запроса таблицы Читатели, Экземпляр и создать в запросе поле Фамилия читателя.
3. Перевести курсор во второе поле и создать новое поле на основе поля Код книги таблицы Экземпляр: *Кол Книг: Код Книги*.
4. Установить операцию группирования (в контекстном меню Групповые операции или нажать кнопку Групповые операции на панели инструментов). Убедиться, что в таблице запроса появилась строка Групповая операция.
5. В поле Кол Книг выбрать поле Группировка и щелкнуть по нему левой кнопкой мыши. В открывшемся меню выбрать агрегатную функцию Count (количество записей) и сохранить запрос с именем Кол Книг.
6. Выполнить созданный запрос и проверить его содержимое.

II. Программирование на языке SQL

A. Формирование запросов к таблице

Все запросы в SQL формируются с помощью одной команды SELECT, после которой указывается место поиска информации в базе (ключевое слово FROM).

Сформировать запрос к таблице Книги

1. Создать новый запрос в режиме конструктора.
2. Закрывать окно Добавление таблиц и щелчком правой кнопки открыть контекстное меню, в котором включить режим представления запроса SQL.
3. В окне Запрос на выборку вписать программу:

```
SELECT Автор, Название { список полей по которым выбирается информация }  
FROM Книги; { имя таблицы из которой выбирается информация }
```
4. Сохранить запрос с именем SQL Книги. Выполнить созданный запрос и проверить его содержимое.
5. Произвести изменения в написанной программе, для этого войти в режим SQL для запроса SQL Книги:

```
SELECT * { для вывода всех полей таблицы }  
FROM Книги;
```
6. Сохранить изменения в запросе SQL Книги. Выполнить созданный запрос и проверить его содержимое.

Выбор по критерию WHERE

1. Создать новый запрос в режиме конструктора и включить режим представления запроса SQL.
2. В окне Запрос на выборку вписать программу:

```
SELECT Автор, Название, Тираж  
FROM Книги  
WHERE Тираж > 60000; { Вывести данные о книгах, Тираж которых > 60000 }
```
3. Сохранить запрос с именем SQL Тираж. Выполнить созданный запрос и проверить его содержимое.

Исключение дублирующих значений

1. Создать новый запрос в режиме конструктора и включить режим представления запроса SQL.
2. В окне Запрос на выборку вписать программу:

```
SELECT Факультет  
FROM Читатели;
```
3. Сохранить запрос с именем SQL Факультет. Выполнить созданный запрос и проверить его содержимое. В запросе название факультетов повторяется многократно.
4. Для исключения дублирующих значений проводят операцию группирования с использованием оператора GROUP BY:

```
SELECT Читатели.Факультет  
FROM Читатели  
GROUP BY Факультет;
```
5. Выполнить созданный запрос и проверить его содержимое. В запросе многократное повторение факультетов исключено.

Б. Создание запроса на основе данных нескольких таблиц

При использовании нескольких таблиц поле прописывается полным именем:

<имя таблицы>.<имя поля>

Вывести фамилии курсантов посещающих библиотеку на основе таблиц: Экземпляр и Читатели

1. Создать новый запрос в режиме конструктора и включить режим представления запроса SQL.
2. В окне Запрос на выборку вписать программу:

```
SELECT Экземпляр.[Номер читательского билета], Читатели.[Фамилия читателя]  
FROM Читатели, Экземпляр  
WHERE Читатели.[Номер читательского билета]=Экземпляр.[Номер читательского билета]
```
3. Сохранить запрос с именем SQL Посещение. Выполнить созданный запрос и проверить его содержимое. В запросе фамилии курсантов повторяются многократно.
4. Для исключения дублирующих значений проводят операцию группирования с использованием оператора GROUP BY:

```
SELECT Экземпляр.[Номер читательского билета], Читатели.[Фамилия читателя]  
FROM Читатели, Экземпляр
```

WHERE Читатели.[Номер читательского билета]=Экземпляр.[Номер читательского билета]

GROUP BY Экземпляр.[Номер читательского билета], Читатели.[Фамилия читателя];

5. Выполнить созданный запрос и проверить его содержимое. В запросе многократное повторение фамилий курсантов исключено.

III. Программирование с использованием языка высокого уровня VBA

Язык VBA применяется для программирования сложных выражений, повторяющихся циклов, обработок ошибок когда программирование встроенными средствами СУБД Access становится затруднительным. Основная форма представления программ на VBA – это процедуры и функции.

В Access предусмотрен ряд способов связывания разработанной процедуры-функции с объектами базы данных. Наиболее распространенными из них являются:

- использования результата вычисления функции в качестве значения поля таблицы *в форме*;
- использования результата вычисления функции в качестве критерия *в запросе*.

A. Связывание результата вычисления функции с критерием отбора записей

Для доступа к редактору VBA используется Модуль. Модуль – это контейнер для хранения программ написанных на языке VBA.

Вывести сведения о книгах с момента издания которых прошло более 50 лет

1. Создать запрос, содержащий следующие поля: Автор, Название, Издательство, Год издания, Количество экземпляров. Сохранить запрос под именем Раритет.
2. Активизируйте окно модуля и нажмите кнопку Создать. Откроется окно модуля и панель управления языком высокого уровня VBA.
3. В меню Insert вызовете команду Procedure. Откроется диалоговое окно процедуры-функции.
4. Нажмите селекторную кнопку Function, введите в поле Name имя функции *Возраст_книги* и щелкните мышью на кнопке ОК (имя функции - это **одно слово** длиной < 255 символов).
5. Access генерирует заготовку функции (начальный и конечный операторы с заказанным именем):

```
Public Function Возраст_книги()
```

```
End Function
```

6. Введите операторы производящие необходимые расчеты:

```
Public Function Возраст_книги()
```

```
Dim ТекГод As Integer 'Вводим переменную ТекГод и присваиваем ей тип Integer
```

```
ТекГод = Year(Date) 'Переменной ТекГод присваиваем значение текущего года
```

```
Возраст_книги = ТекГод - 50 'Вычисляем критерий Возраста раритета
```

```
Debug.Print (Возраст_книги) 'Выводим результат в окно Immediate
```

```
End Function
```

Примечание: окно *Immediate* открывается: *View – Immediate Window*

7. Запустите программу на выполнение (*Run*) и при необходимости устраните ошибки. Затем обязательно сохраните программу (*File – Save Библиотека*).
8. Откройте запрос Раритет в режиме конструктора и в поле Год издания щелкните мышью в строке Условие отбора.
9. Запустите программу Построитель выражений (контекстное меню – команда Построить). При этом открывается окно Построитель выражений.
10. В левом окне откройте закладку Функции – Библиотека, в среднем –Module1 и вставьте в верхнее окно редактирования функцию *Возраст_книги*, а затем условие отбора:

```
< Возраст_книги()
```

Нажмите кнопку ОК и убедитесь, что Условие отбора ссылается на написанную Вами функцию *Возраст_книги()*.

11. Не закрывая запроса Раритет (Внимание! Связывание модуля написанного на языке VBA с объектами базы данных происходит лишь в том случае если эти объекты находятся в фазе выполнения) откройте и еще раз запустите, созданную Вами программу, на выполнение. После этого функция *Возраст_книги()* буде однозначно связана с запросом Раритет.
12. Закройте все открытые объекты и выполните запрос Раритет. Убедитесь, что запрос выполняется правильно.

Б. Связывание результата вычисления функции с значением поля

Вычисляемое с помощью процедуры-функции значение поля можно связать лишь с двумя объектами базы данных: Формами и Отчетами.

Вычисление стоимости утерянной книги

Стоимость утерянной книги вычисляется по следующей формуле:

$$C_{PK} = C_K \cdot K_P \cdot K_S,$$

Где C_K - цена книги;

K_P - коэффициент пересчета, устанавливается администрацией Библиотеки (Пусть $K_P = 3$);

K_S - коэффициент старения (для новой книги равен 1, в последующем каждый год уменьшается на 1%).

1. Составить запрос к базе Библиотека, содержащий следующие поля: Код Книги, Фамилия читателя, Отметка о потере, Цена, Компенсация за потерю.
2. В качестве Условия отбора выбрать «Да» в поле Отметка о потере. И сохранить под именем Потеря книги.
3. Используя Мастер форм, создать форму к запросу Потеря книги. Сохранить форму с именем Стоимость потери.
4. Войти в режим создания и редактирования Модулей, записать и сохранить следующую программу с именем Module2:

Public Function Потеря()

'Ссылка на процедуру вычисляющую компенсацию за потерю книги

Вычисление

End Function

Public Sub Вычисление()

'При расчете компенсации за потерю книги указывается путь к вычисляемому полю через!

*Forms![Стоимость потери]![Компенсация за потерю] = Forms![Стоимость потери]!Цена
* КоэфПотери() * КоэфСтарения()*

End Sub

Public Function КоэфПотери()

'Функция в которой определяется значение коэффициента потери

КоэфПотери = 3

End Function

Public Function КоэфСтарения()

'Для вычисления Коэффициента старения используется функция СрокЭксплуатации()

*КоэфСтарения = 1 - СрокЭксплуатации() * 0.01*

End Function

Public Function СрокЭксплуатации()

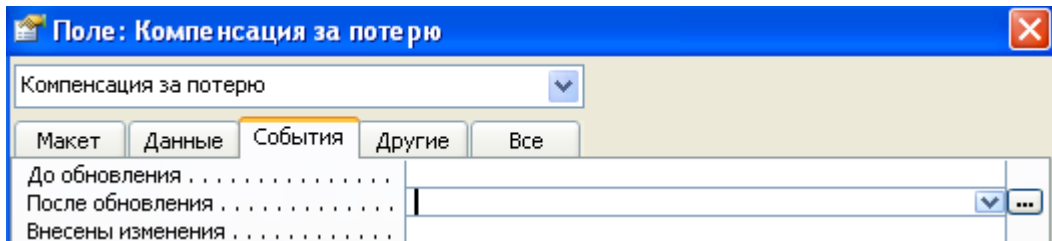
'Срок эксплуатации книги определяется как разность между текущим годом и годом издания

СрокЭксплуатации = Year(Date) - Forms![Стоимость потери]![Год издания]

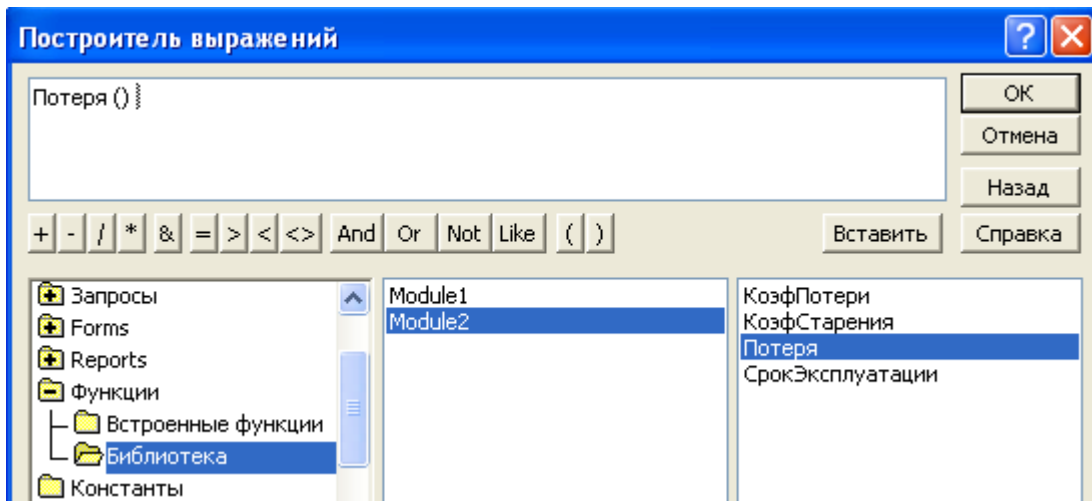
End Function

5. Открыть форму Стоимость потери в режиме Конструктора, выделить поле Компенсация за потерю и в контекстном меню к данному полю открыть меню Свойства.

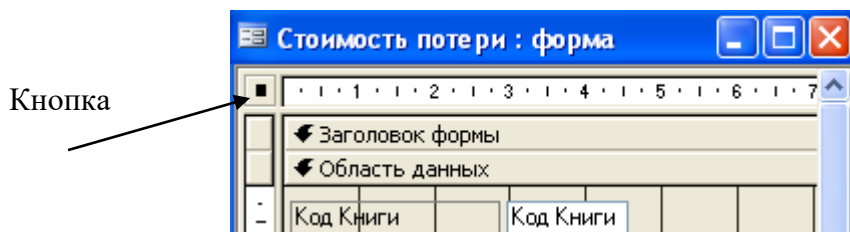
6. В окне «Поле: Компенсация за потерю» выбрать закладку События и щелкнуть на графе После обновления. Справа от графы появится пиктограмма с тремя точками



7. Щелкните на данной пиктограмме, Откроется окно Построителя. Построитель предлагает три варианта подключения программы, написанной на языке VBA к форме. Выберем режим подключения через выражения. При этом откроется окно Построитель выражений.
8. Последовательно выбрать: Функции - Библиотека- Module2 – Потеря и нажать кнопку Вставить. В окне построителя появится ссылка на функцию Потеря().

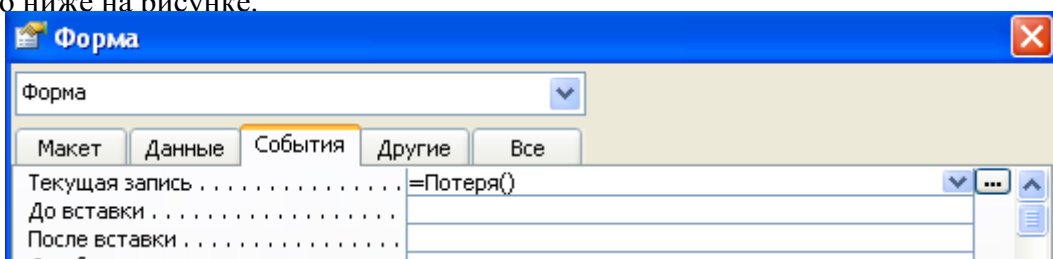


9. Нажмите кнопку ОК и убедитесь, что событие *После обновления* ссылается на функцию Потеря().
10. Закройте форму Стоимость потери, сохранив при этом изменения.
11. Откройте форму Стоимость потери и не закрывая ее откройте Module2 и запустите программу на выполнение (Run). Если трансляция прошла без ошибок, то считается, что созданная процедура-функция Потеря() успешно подключена к объекту базы данных.
12. Закройте окно редактора VBA, закройте и вновь вызовете форму Стоимость потери. Убедитесь, что расчеты проведены лишь в первой строке. Это говорит о том, что расчеты выполняются лишь для той записи, которая на данный момент является активной.
13. Для привязки программы к любой из записей необходимо их активизировать. С этой целью необходимо сделать привязку разработанной программы к текущей записи. Для этого:
 - откройте форму Стоимость потери в режиме конструктора и вызовете контекстное меню



формы.

- щелкните на папке меню Свойства и в открывшемся окне Форма выберите закладку События. Повторив пункты 6-11 привяжите текущую запись к разработанной программе, как показано ниже на рисунке.



14. Для сохранения всех внесенных в базу изменений закройте базу Библиотека и вновь откройте ее.
15. Откройте форму Стоимость потери и убедитесь в правильности работы Вашего запроса.

Дополнительное задание

Задание 1. В связи с индексацией цен стоимость всех книг выросла на 15%. Спроектировать запрос к таблице Книги выводящий информацию по следующим полям: Название, Цена, Новая Цена.

Задание 2. Спроектировать запрос к таблице Экземпляр, автоматически вычисляющий дату возврата по дате выдачи (через 10 дней) под именем Расчет возврата. По данному запросу создать форму Бланк выдачи и заполнить строку:

K 14 _ 003 _ 2 _ 25. 12. 2007_ Да _ На руках

убедится, что дата возврата заполнится автоматически.

Задание 3. Вывести сведения о максимальном количестве экземпляров книг по каждой из тематик. Запрос должен содержать два поля: Тематика, Мах Кол Экз.

Задание 4. Вывести список курсантов не посещающих библиотеку.