



Таблицы MapInfo Professional. Структура. Типы данных.

Для начала, чтобы знать, с чем нам придется работать, разберем понятие «таблица MapInfo». Таблица, как известно, состоит из колонок и строк, или, говоря языком программистов - ПОЛЕЙ и ЗАПИСЕЙ соответственно. Каждая запись может содержать графический объект, например, прямую линию, точку или область. MapInfo также поддерживает растровые таблицы, которые содержат только изображение и не имеют записей и полей. Немного приоткроем вещи, как правило, скрытые от обычного рядового пользователя.

Физически, на диске, таблица представляет собой несколько файлов с данными MapInfo. В «спокойном состоянии», когда таблица не открыта в MapInfo, не использует специальных индексов и при создании для нее включена возможность содержать графические объекты, она представляет собой набор из четырех файлов с расширениями DAT, ID, MAP и TAB. Первые три файла содержат служебную двоичную информацию, описывать и трогать их здесь мы не будем. Остановимся на файле TAB-типа. Этот файл текстовый, содержит описатель таблицы, например:

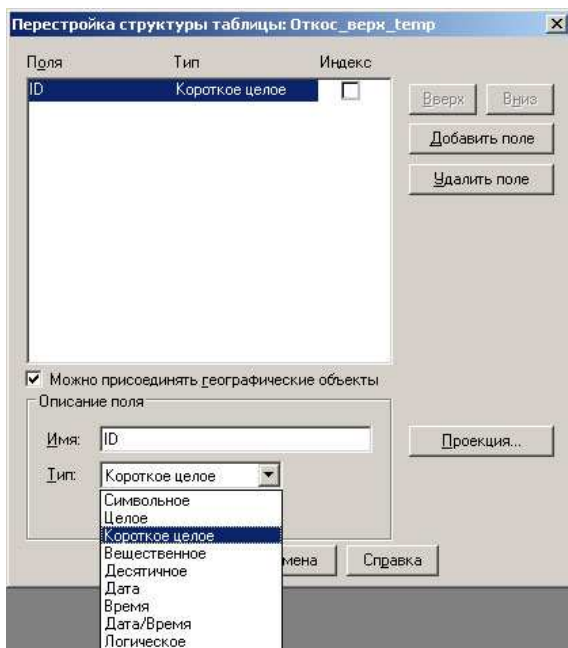
```
!table
!version 300
!charset WindowsCyrillic

Definition Table
Type NATIVE Charset "WindowsCyrillic"
Fields 5
  ID Integer;
  ObjNodeNum Integer Index 1;
  ObjectName Char (150);
  X Float;
  Y Float;
```

Из содержания следует, что это таблица, указан формат версии MapInfo при создании таблицы (100 – более ранние, 300 – начиная с 3-й версии), charset определяет кодировку символов в таблице.

Далее, определение (definition table) сообщает о типе таблицы (NATIVE – формат обмена MapInfo, DBF – формат dBASE и других совместимых баз данных и т.д.), кодировку символов, количество полей (Fields 5) – 5 полей. После этого следует описатель самих полей – имя поля, тип данных поля, при необходимости – его размерность (при строковом типе, например) и информация об индексировании поля.

Создание таблицы Вы производите с помощью пунктов меню «Новая таблица» - «Создать», изменение структуры (описания полей) имеющейся (и открытой в данный момент) таблицы с помощью пункта «Таблица» - «Изменить» - «Перестроить» главного меню MapInfo. С помощью нижеприведенного окна разберем типы данных в таблицах MapInfo. Эти знания нам обязательно потребуются в дальнейшем.



Поле служит для описания колонки в таблице, т.е. присвоению типов и размеров данных, содержащихся в данной колонке. Доступные типы данных:

Символьное (Char(n)) – символьное поле, для хранения текстовой информации. При создании необходимо дополнительно указать размерность поля (максимальную длину хранимой строки). Для таблиц MapInfo максимально доступно хранение 254 символов в одном символьном поле;

Целое (Integer) – целое число от -2147483647 до +2147483647 включительно;

Короткое целое (SmallInt) – короткое целое число от -32767 до 32767 включительно;

Вещественное (Float) – дробное число (число с плавающей запятой);

Десятичное (Decimal(n,m)) – дробное число, у которого можно жестко указать количество знаков после запятой, удобно для денежного представления, например;

Дата (Date), Время (Time), Дата/Время (DateTime) – поля для хранения различных временных величин;

Логическое (Logical) – поля для хранения логических величин, может содержать значения Т («True») – «истина», или F («False») – «ложь».

При создании таблицы должно быть определено хотя бы одно семантическое поле!

В случае, когда при создании таблицы было разрешено привязывать к ней графические объекты, таблица будет содержать еще одно поле – OBJ. Это поле часто не документировано, но именно через название этого поля мы будем обращаться к графическому объекту карты, привязанного к каждой записи таблицы. Запомним название этого важного поля – OBJ!

Знание типов данных необходимо нам для оперирования этими данными с использованием элементов языка MapBasic, т.к. различные типы данных могут требовать использования жестко привязанных к этим типам данных выражений, операторов и функций языка.

Поэтому, перед созданием таблицы желательно проводить некоторое ее предварительное проектирование, выяснить, какие семантические данные нам могут понадобиться в данной таблице в дальнейшем, определить типы и максимальные величины этих данных.

Если таблица – зарегистрированный в MapInfo растровый файл, то её TAB-файл будет выглядеть несколько иначе - добавлена ссылка на сам

растровый файл, указан растровый тип таблицы, содержится информация о координатах точек растра и координатной системе, которые использовались при регистрации данного изображения в MapInfo:

```
!table
!version 300
!charset WindowsCyrillic
```

```
Definition Table
File "10+4.tif"
Type "RASTER"
(4000,11000) (0,0) Label "Точка 1",
(5000,11000) (5904,0) Label "Точка 2",
(5000,10000) (5904,5904) Label "Точка 3",
(4000,10000) (0,5904) Label "Точка 4"
CoordSys NonEarth Units "m"
Units "m"
```

Касательно растровой таблицы скажу, что допустимо менять «вручную» название файла самого растрового изображения и координаты регистрации, делать это иногда приходилось, в частности для ускорения вывода на печать одинаково оформленных отчетов, но с разными подложенными растрами. В описателе же стандартной таблицы лучше не вносить изменения, скорее всего, будет выдана ошибка при попытке открыть измененный описатель NATIVE-формата.

Вы должны знать, что после выполнения запроса, при получении ненулевой выборки все данные копируются в специальную таблицу Selection. Это, по сути, результат Вашего запроса. Кроме того, если был включен вывод результата запроса в список, таблица Selection дублируется в автоматически создаваемых таблицах с именами Query__ с числовыми индексами в конце. В этих таблицах будет сохраняться история результатов Ваших предыдущих запросов, а таблица Selection будет содержать только результат последнего запроса. Это тоже надо запомнить.

Таблица может быть получена путем импорта из других форматов. Совет: при импорте из DXF-формата – перед импортом файл DXF должен быть сохранен с типом AutoCAD R12/LTL2 DXF. Кроме того, при обратном экспорте в DXF-формат, во избежание дальнейших ошибок я рекомендую производить предварительную упаковку экспортируемых таблиц для окончательного уничтожения ранее удаленных записей.

В следующей статье постараюсь подробно осветить вопросы создания различных выборок данных в таблицах карты стандартными средствами MapInfo, используя доступные операторы и функции MapBasic.

*С уважением и успехов Вам!
Карасов Андрей.*

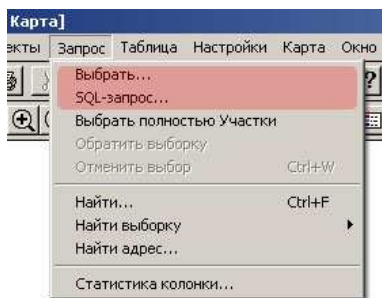
Все права на данные статьи принадлежат ООО «КАРАСОФТ Дизайн», запрещается копирование и опубликование этих материалов на других публичных ресурсах без согласования с автором.



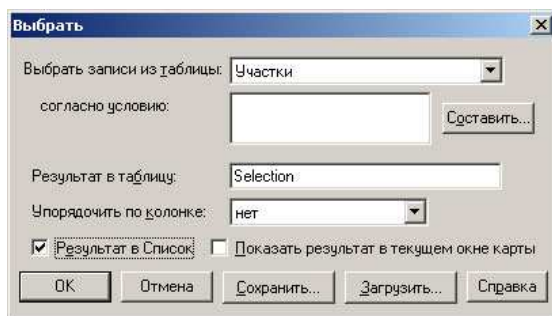
Использование языка MapBasic в запросах MapInfo

Кроме того, что MapInfo позволяет отображать данные на географических картах в виде графики, она содержит также мощный аппарат анализа, который дает Вам возможность группировать и организовывать полученные данные. Разбив данные на логические группы, осуществив необходимые выборки, Вы можете проводить более продуктивную работу, применяя групповые действия только к необходимому диапазону полей и записей.

Остановимся на выполнении запросов в MapInfo. Стандартными средствами для этих операций мы, очевидно, будем использовать два нижеуказанных пункта меню программы:



Использование первого пункта «Выбрать» служит для создания наиболее простых выборок данных в пределах одной таблицы. Выбор этого пункта приводит к открытию диалогового окна

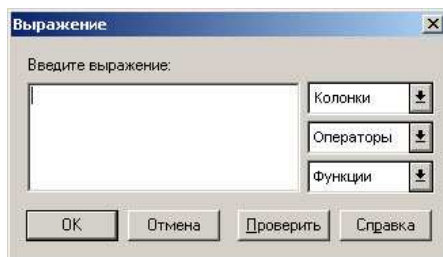


Назначение элементов ввода и управления данного диалога Вам уже должно быть известно из опыта, в противном случае, воспользуйтесь Справкой, там все подробно описано, пока не буду загружать эту статью, если кому надо – включу сюда описание.

Хотелось бы подробно остановиться на заполнении окна «согласно условию:», так как именно здесь строится Ваш запрос к таблице и именно в этом поле ввода мы должны грамотно сказать базе данных MapInfo, что мы хотим получить от той или иной ее таблицы. Это уже можно назвать программированием и здесь все имеет строгий синтаксис, и далеко не все будет зависеть от Вашей фантазии, хотя применять ее в начальном осмыслении запроса далеко не плохо! Так вот, условие можно писать как от руки, так и воспользоваться помощью кнопки «Составить...», которая предложит Вам список доступных в данном окне операторов и функций MapBasic. Изучению элементов этого списка была посвящена отдельная статья, сейчас мы займемся практическим применением некоторых.

Во-первых, скажу, что выполнение данного запроса с пустым полем «согласно условию:» (буду далее называть просто «условием») приводит к выборке всех записей таблицы. Это тоже применяется в работе, и довольно часто.

Жмем кнопку «Составить...», получаем еще одно диалоговое окно:



Здесь справа мы имеем три выпадающих списка. «Колонки» – список семантических полей запрашиваемой таблицы, плюс поле obj, которое, как мы уже знаем, содержит графический объект каждой записи и к которому тоже можно обращаться. И два списка с доступными здесь операторами и функциями MapBasic. Описание элементов этих двух списков достаточно объемно и мною было выложено в отдельную статью, кроме того, выкладываю для скачивания в конце цикла этих статей перевод полного руководства по MapBasic в формате PDF. Займемся полезными примерами, вводя различные выражения:

RowID <= 10 – получаем выборку записей таблицы с порядковым номером меньше или равным 10, грубо говоря, первые 10 записей. RowID() – это, вообще говоря, функция, причем редко документированная – ее даже нет в списках операторов и функций данного окна, возвращает порядковый номер текущей записи таблицы, эту строку приходится писать вручную;

RowID > 10 And RowID < 30 – получаем выборку с 11 по 29 запись по порядку;

Area(obj, "m") > 1000 – получаем выборку записей, площадь чьих объектов карты больше 1000 кв. м.;

Area(obj, "hectare") > 0 – получаем выборку только площадных объектов;

Area(obj, "hectare") = 0 – получаем выборку не площадных объектов (точки, линии и т.д.);

ObjectLen(obj, "m") > 0 – получаем выборку только линейных объектов таблицы;

ObjectLen(obj, "m") > 100 – имеем выборку линейных объектов таблицы, имеющих длину более 100 м;

Немного посложнее:

(RowID / 2 - Int(RowID / 2)) = 0 – выборка только четных записей по порядку следования в таблице,

(RowID / 2 - Int(RowID / 2)) <> 0 – только нечетных записей.

Здесь я намеренно не затрагиваю работу с семантическими полями, там поле деятельности еще более обширное, но, при наличии описания операторов и опыта предыдущих примеров Вам не составит труда разобраться с составлением этого типа выборки самостоятельно. Ну, допустим, необходимо выбрать все объекты таблицы строений с содержимым поля «Тип» равным «КН» Пишем условие **Тип = "КН"**. Ничего сложного? Я же говорю – без особого труда сами разберетесь.

Думается, принцип ясен. Имея список доступных операторов и функций, а также их описание, некоторую фантазию и конкретную задачу, Вы сможете составить собственные условия! Если возникнут проблемы – пишите, разберем Ваш пример.

При установленной галочке «Результат в Список» Вы получаете результат в дополнительно создаваемой таблице с именем QueryNN, где NN – порядковый номер вспомогательной таблицы. Данные этих таблиц Вы можете использовать как данные обычных таблиц, кроме того, данные в Query – по сути, аналог данных основной таблицы, изменения этих данных приведут к изменению данных этих же записей в основной таблице. Помните это!

Имея выборку данных, легко осуществлять групповые операции именно над этой выборкой, не затрагивая остальные записи основной таблицы! При наличии таблицы с 20000 записями, поменять что-либо только у четных, к примеру, записей становится гораздо проще пареной репы. Не так ли? Об операциях автоматической модификации групповых данных таблиц расскажу в отдельной статье.

Перейдем теперь к следующему пункту меню MapInfo «SQL-запрос...».

Из-за обилия форм ввода данное окно может показаться довольно сложным. На самом деле это не совсем так, здесь содержится только все необходимое для построения грамотных SQL-запросов с участием от одной до нескольких таблиц. Напомним, что SQL переводится с английского как «структурированный язык запросов», является стандартным средством для управления базами данных, получения из необходимой информации (только необходимой – так и хочется вставить), сохранения в них наших данных.

Рекомендую всем почитать пункты справки для освоения назначения всех элементов диалога и рекомендаций по их применению (кнопка «Справка»).

Вообще, принцип работы по построению запросов здесь сходен с принципом предыдущего пункта меню. В этом диалоге присутствуют дополнительные возможности от доступного здесь использования сразу нескольких таблиц. Можно строить более сложные запросы, например, необходимо выбрать объекты карты из Таблицы_1, которые территориально входят в объекты Таблицы_2. Делаем следующее: в поле «из таблиц:» заносим **Таблица_1**, **Таблица_2**. В поле «с условием:» пишем или набираем используя выпадающие списки справа **Таблица_1 Within Таблица_2**. Какое-то время любуемся результатом!

По роду деятельности мне часто приходилось сталкиваться с запросами именно географического характера, касающиеся взаимного расположения объектов двух таблиц на карте. Использовались такие операторы как Contains, Contains Entire, Within, Entirely Within и Intersects. Описание этих и других команд MapBasic Вы найдете в статьях «Описание операторов и функций языка MapBasic для производства выборок данных в таблицах MapInfo».

К сожалению, SQL-запросы в MapInfo не будут работать с данными из таблиц Selection и QueryNN. Здесь существует разрешение на использование только базовых, полноценных таблиц. Это надо учитывать при проектировании Вашего запроса. Поэтому, в качестве выхода, приходится предварительно сохранять выборку в полноценную таблицу, открывать и работать с ней в дальнейших запросах.

Имея эти сведения, Вы должны справиться с построением собственных выборок и SQL-запросов в таблицах MapInfo. При возникновении проблем, наличии вопросов, прошу воспользоваться механизмом отзывов на темы, либо пишите мне на mail@karasoft.ru Постараюсь ответить и помочь Вам в кратчайшие сроки.

С уважением и успехов Вам!
Карасов Андрей.

Все права на данные статьи принадлежат ООО «КАРАСОФТ Дизайн», запрещается копирование и опубликование этих материалов на других публичных ресурсах без согласования с автором.



Операторы языка MapBasic для уловий выборов в MapInfo

Сразу скажу, что поскольку занялись мы программированием, то существуют определенные правила и соглашения использования в выражениях строковых констант, чисел и дат.

Символьные строки

Символьные константы в выражении заключаются в кавычки. Если строка не заключена в кавычки, то MapInfo Professional будет воспринимать её как имя поля (колонки).

Например, строки 1 и 2 - константы, а строки 3 и 4 - нет.

1. "Красное"
2. "Красноярск"
3. Красное
4. New York

Числа в выражениях

Для задания чисел в выражениях не используйте запятую, знак доллара и другие знаки, не являющиеся цифрами, точкой и знаком минус для отрицательных чисел. Для задания числовых величин в экспоненциальном виде может также использоваться символ E.

Даты в выражениях

Дата состоит из месяца дня и года (последнее необязательно). Год может задаваться двумя или четырьмя цифрами. Значения месяца, дня и года разделяются либо знаком дефиса, либо косой чертой. Сама дата заключается в кавычки. Если год в дате не задан, то понимается, что эта дата текущего года.

Для описания операторов MapBasic, используемых в диалогах выборов воспользуемся справочными данными языка и немного перегруппируем списки команд.

Математические операторы в выражениях

Математические операторы часто используются в выражениях. Следующая таблица представляет символы операторов, имена, примеры и правила преобразование типов значений.

Символ	Описание	Пример	Подробнее о примере
+	сложение	A + B	дата + число: дата 04/01/2007 + 4: 04/05/2007 вещественное + дата: дата 4 + 232: 236
-	вычитание	A - B (вычитание) -A (отрицательное)	дата - число: дата число - дата: дата 236 - 4: 232 любое число - любое число: вещественное
*	умножение	A * B	любое целое * любое целое: целое любое число * любое число: вещественное
/	деление	A / B	любое число / любое число: вещественное число
^	возведение в степень	A ^ B	любое число ^ любое число: вещественное число

Разрешаются следующие виды вычислений:

- Сложение чисел с датами с получением новой даты
- Вычитание чисел из дат с получением новой даты
- Вычитание даты из даты с получением числа

При прибавлении чисел к датам или вычитании чисел из дат, MapInfo считает число, определённым количеством дней. Так, для вычитания или прибавления недели надо использовать число 7, если необходимо прибавить или вычесть месяц, следует использовать 30 или 31. Когда MapInfo Professional вычитает дату из даты, в результате получится количество дней.

Использование строковых операторов в выражениях

+	"конкатенация" - объединение строк и строковых выражений.
&	(может использоваться если предваряется пробелом)

Помните, что строки должны быть заключены в двойные кавычки.

"Здравствуй, " + "мир." дает "Здравствуй, мир."
"4"+"5" даёт "45."

Использование операторов сравнения в выражениях

В некоторых диалогах MapInfo есть возможность вызвать диалог Выражение для составления математических и алфавитно-цифровых выражений формул. В выражениях часто используются операторы сравнения. В следующей таблице приводятся знаки для операторов сравнения и их описание.

Операторы	Описание
=	"равенство "
<>	"неравенство "
>	"больше"

<	"меньше"
>=	"больше или равно"
<=	"меньше или равно"

Сравнение чисел

Численное сравнение основано на численных значениях выражений и численных констант.

Пример: выбрать все строки, в которых доход больше \$65,000.

```
HH_INC>65000
```

Комментарий: Не добавляйте знак доллара или запятую. MapInfo Professional не знает что делать с этим и Вы получите сообщение об ошибке.

Пример: покажите все районы, где средний возраст населения составляет 42 года;

```
MED_AGE=42
```

Комментарий: Это выражение выбирает только те записи, где средний возраст ровно 42 года. Когда колонка со средним возрастом содержит десятичную точку (как в случае демографических данных поставляемых в комплекте с MapInfo Professional), то маловероятно, что найдётся много регионов, где средний возраст будет равен ровно 42.

Следующее выражение даёт лучший результат:

```
Round(MED_AGE, 1)=42
```

Комментарий: Функция "round(любое число, любое число)" округляет первое число, с точностью до второго числа. В этом примере первое число это средний возраст (MED_AGE), а второе это 1, указывающее что средний возраст должен быть округлён до ближайшего целого числа.

Пример: Все строки, где значение не равно \$23,000.

```
AMOUNT<>23000
```

Комментарий: Вы можете использовать функцию Round, с параметром округления 10, если не преследуете цель получить значение точно равное 23000.

Сравнение строк

Сравнение строк основано на точном соответствии символов содержащихся в строке. В случае ">" это обозначает "дальше в алфавите" и "<" обозначает "раньше в алфавите." Когда вводите строку в выражении заключайте её в кавычки, для того чтобы MapInfo Professional рассматривало её как строку, а не как название колонки.

Пример: Все строки, где продавец Acme.

```
VENDOR="Acme"
```

Комментарий: обратите внимание, что строка Acme заключена в кавычки, для того чтобы MapInfo Professional рассматривало её как символьную строку, а не искало колонку с названием Acme.

Пример: Все строки, где продавец не Acme.

```
VENDOR<>"Acme"
```

Сравнение дат

Пример: Все сообщения принятые 9 октября, 1991.

```
RECEIVED="09.10.1991"
```

Примечание: Имейте в виду:

- Дата заключается в кавычки
- Дата записывается в форме: месяц, день, год
- Числа в дате разделяются дефисом или косой чертой (/)
- Для года было использовано два символа. Но можно использовать и четыре символа (1991)

Пример: Все сообщения принятые после 9 октября, 1991.

```
RECEIVED>"09.10.1991"
```

Комментарий: Это выражение не выберет записи с сообщениями принятыми 9 октября, 1991. Если Вы хотите чтобы были выбраны и они:

```
RECEIVED>="09.10.1991"
```

Пример: Записи с сообщениями принятыми до Августа.

```
Month(RECEIVED)<8
```

Комментарий: Это выражение использует функцию Month чтобы извлечь месяц из даты. Конкретный год не указывается. Если база данных содержит записи за несколько лет, это выражение не будет принимать во внимание год.

Логическое сравнение

Пример: Всё что продано.

```
Shipped
```

Комментарий: Колонка "Shipped" - логическая. Она содержит "T" если значение истинное, в противном случае "F" - ложное. Когда товар продан, он помечается "T". В противном случае он не продан. Для проданных товаров выражение из примера ищет значение истина. Для непроданных - значение ложь.

Пример: Всё что не продано.

```
Str$(Shipped)="F"  
Not Shipped
```

Логические операторы в выражениях

"And" (И), "or" (ИЛИ) и "not" (НЕ) - это логические операторы. MapInfo Professional рассматривает каждое такое выражение как проверку и применяет к каждой записи в таблице. Результатом каждой проверки является ответ "да" или "нет" (т.е. "истина" или "ложь"). Комбинируя результаты проверки каждого условия с помощью логических операторов, MapInfo Professional выдает общий ответ: удовлетворяет ли данная запись условию выбора.

Оператор	Описание
AND	принимает значение "истина" только в том случае, если оба ее аргумента (логические выражения) истинны. То есть запись должны удовлетворять обоим условиям, чтобы попасть в выборку.
OR	принимает значение "истина" если один или оба ее аргумента (логические выражения) истинны. То есть запись должны удовлетворять хотя бы одному из условий, чтобы попасть в выборку. Записи также выбираются, если удовлетворяют обоим условиям.
NOT	принимает значение "истина", если аргумент имеет значение "ложь". То есть записи не должны удовлетворять условию, чтобы попасть в выборку

Предположим, Вы хотите выбрать дома стоимостью от \$250,000 или больше, находящиеся в графстве Колумбия. Каждая запись должна отвечать двум критериям, каждый из которых может быть определен как простое выражение:

```
VALUE >= 250000
COUNTY = "Columbia"
```

Сначала Вы можете сделать выборку всех домов, которые стоят от 250,000 и более. Затем сделать новую выборку из этой выборки, отобрав дома которые находятся в графстве Колумбия. Однако, проще объединить два эти оператора в одно выражения, используя логический оператор "and":

```
1. COUNTY = "Columbia" and VALUE >= 250000
```

MapInfo Professional проверяет запись на соответствие условию заданному в выражении, для этого делается две проверки: Действительно ли в поле COUNTY значение Columbia? Является ли значение в поле VALUE больше или равным 250000? Если ответ на оба эти вопроса истина (т.е. ДА), значит, запись попадает в текущую выборку. Если ответ на один или оба вопроса - НЕТ (т.е. ложь), запись не добавляется к выборке.

Теперь, допустим, Вы ищете недвижимость стоимостью от \$250,000 и находящуюся где угодно, только не в графстве Columbia? Вы можете использовать оператор "not" чтобы обратить значение первой части прошлого выражения, превратив его в выражение:

```
not (COUNTY="Columbia") and VALUE>=250000
```

Только те записи, в которых графство не Columbia будут удовлетворять требованиям первой части выражения.

```
2. not (COUNTY="Columbia" and VALUE>=250000)
```

Выражение 2 это отрицание для выражения 1. Любая запись, удовлетворяющая выражению 1 не удовлетворяет выражение 2. Любая запись, не удовлетворяющая выражению 1 удовлетворяет выражение 2.

Вы можете использовать "or" когда хотите указать альтернативные условия, например:

```
3. COUNTY="Columbia" or COUNTY="Greene"
```

Любая запись относящаяся к графству Columbia или к графству Greene будет удовлетворять условию. Условие, конечно, может содержать и численные значения. Например:

```
TOTAL_AREA>40 or VALUE>250000
```

Это условие отбирает записи, в которых площадь больше 40 или стоимость больше 250000. Если запись отвечает хоть одному из этих условий, она помещается в выборку. Будьте внимательны при построении выражений с использованием логических операторов.

Следующее выражение выглядит нормально, однако работать корректно оно не будет:

```
COUNTY="Columbia" or "Greene"
```

Оценивая это выражение как английскую фразу-ГРАФСТВО равно Колумбия или Грин - мы можем предположить, что это выражение будет работать так же как выражение 3 и выбрать все записи, где встречается графство Columbia или графство Greene. Но правила вычислительной логики несколько отличаются от правил английского языка.

Численные значения в выражениях

Пример: Все домовладения, доход которых выше \$65,000 и средний возраст 42.

```
NH_INC>=65000 and Round(MED_AGE, 1)=42
```

Комментарий: это выражение состоит из двух частей. Части соединены оператором "and", это означает, что обе части выражения должны быть истинными, для того чтобы запись была выбрана.

Пример: выбрать все записи, у которых цена между \$50,000 и \$100,000.

```
4. PRICE>=50000 and PRICE<=100000
```

Комментарий: теперь мы устанавливаем два условия, которым должны отвечать записи, для того чтобы быть выбранными. Используя, операторы "больше или равно" (>=) и "меньше или равно" (<=) мы включаем в выборку записи, у которых значение точно равно указанным \$50,000 и \$100,000. Если мы укажем просто "больше чем" (>) и "меньше чем" (<), выражение выберет записи от \$50,001 до \$99,999 но не включит \$50,000 или \$100,000.

<>bПример: выбрать записи с ценами от \$50,000 до \$100,000 или от \$150,000 до \$200,000.

```
5. (PRICE>=50000 and PRICE<=100000) or (PRICE>=150000 and PRICE<=200000)
```

Комментарий: общий вид выражения: (expression1) or (expression2). Каждое из этих выражений имеет тот же вид что и выражение 4. Когда строка отвечает условию, указанному в любом из этих выражений expression1 или expression2, MapInfo Professional поместит эту строку в выборку.

Строки в выражениях

Пример: выбрать всех клиентов от N до Q.

```
LAST_NAME >= "N" and LAST_NAME <"R"
```

Комментарий: первая часть выражения ищет фамилии, начинающиеся на букву "n" или расположены в алфавите после (дальше) "n". Вторая часть выражения проверяет фамилии, которые начинаются с буквы, которая в алфавите расположена раньше (до) буквы "r". Любая фамилия, начинающаяся от "n" до "q" удовлетворяет условию.

Пример: выбрать всех клиентов от N до Z.

Примечание: Операторы сравнения строк в MapInfo Professional не чувствительны к регистру.

```
LAST_NAME >= "N"
```

Комментарий: так как "z" последняя буква в алфавите, то дополнительной проверки не требуется. Не может быть клиентов, фамилия у которых начинается с некой буквы следующей за буквой "z".

Пример: все клиенты, фамилия которых начинается на C.

```
LAST_NAME>="C" and LAST_NAME<"D"
```

Пример: выбрать всех клиентов, фамилии которых между "Ce . . ." и "Cn . . ."

```
LAST_NAME=>"CE" and LAST_NAME<"CO"
```

Пример: выбрать всех кто не из Техаса.

```
not STATE="TX"
```

Комментарий: в этом выражении MapInfo Professional будет проверять содержимое колонки STATE. Если содержимое в этой колонке отличается от "TX", запись будет выбрана.

Даты в выражениях

Пример: выбрать все записи с датой Август 1990

```
Month(RECEIVED)=8 and Year(RECEIVED)=1990
```

Комментарий: в этом выражении мы указываем год явно, используя функцию "year()" чтобы извлечь год из даты.

```
RECEIVED>="01.08.90" and RECEIVED<="31.08.90"
```

Комментарий: в этом выражении мы явно не запрашиваем месяц и год. Вместо этого мы создали фразу, которая на английском звучит так: все записи с датой 1 августа 1990 и позднее и с датой до 31 августа 1990 включительно. Оба выражения дают одинаковый результат, но сформулированы запросы по-разному.

Пример: выбрать записи в которых дата июль или сентябрь

```
Month(RECEIVED)=7 or Month(RECEIVED)=9
```

Комментарий: как и в предыдущем примере, год не учитывается. В зависимости от данных в таблице и Вашей цели это может быть или не быть проблемой.

Если вы хотите указать год:

Пример: выбрать записи, в которых дата июль или сентябрь 1989 года

```
month(RECEIVED)=any(7, 9) and year(RECEIVED)=89
```

Комментарий: первая часть этого выражения использует ключевое слово "any". Первая часть выражения будет отвечать условиям, если будет совпадение со значениями указанными в скобках, в данном случае это месяцы июль и сентябрь.

Ключевые слова в выражениях

В MapInfo Professional можно использовать специальные зарезервированные слова "any", "all", "in" и "between". При задании выражений эти ключевые, зарезервированные слова надо набирать с клавиатуры.

Слово "any" обозначает выбор любого из элементов множества.

```
6. ABBR = any("AL", "MN", "TX")
```

Выражение будет истинным для любой записи, у которой штат Alabama, Minnesota, или Texas.

Чтобы понять значение "all", рассмотрим пример:

```
ABBR <> all("AL", "MN", "TX")
```

Это выражение означает: выбрать все записи, для которых штат не равен Алабама, Миннесота или Техас. Будут выбраны все записи, кроме записей для Алабама, Миннесота или Техас. Посмотрим что получится если мы используем выражение:

```
ABBR <> any("AL", "MN", "TX")
```

Заказы из Алабамы будут выбраны, так как они не из Миннесоты или Техаса, и записи из Миннесоты и Техаса будут выбраны по той же причине.

Этот пример показывает использование ключевого слова "in":

```
ABBR in("AL", "MN", "TX")
```

Результат будет такой же как и в выражении 6.

В данном случае зарезервированное слово "in" эквивалентно "=any", а "not in" - "<>all".

И наконец, рассмотрим выражения 7 и 8 в которых иллюстрируется применение "between":

```
7. ЦЕНА between 50000 and 100000
```

```
8. (ЦЕНА between 50000 and 100000) or (ЦЕНА between 150000 and 200000)
```

Это эквиваленты выражений 4 и 5 соответственно. Вы также можете использовать between с символьными строками.

Географические операторы в выражениях

В MapInfo Professional имеется несколько географических операторов. Они используются для выбора объектов на основании их взаимного расположения в пространстве. С географическими операторами в MapInfo Professional используется специальное зарезервированное слово: "obj" или "object". Оно определяет, что MapInfo Professional должно вычислить значение на основании графических объектов, а не соответствующих им табличных данных.

Имя географического оператора указывается между географическими объектами. Выбрать его можно в списке "Операторы" в диалоге "SQL-запрос". Здесь перечислены географические операторы:

Contains	Объект А содержит объект В, если центроид объекта В лежит в границах объекта А.
Contains Entire	Объект А полностью содержит объект В, если границы объекта В полностью лежат внутри границ объекта А.
Within	Объект А находится внутри объекта В, если его центроид находится в границах объекта В.
Entirely Within	Объект А находится полностью внутри объекта В, если его граница полностью находится внутри границ объекта В.
Intersects	Объект А пересекается с объектом В, если они имеют хотя бы одну общую точку.

"Contains" и "Within" имеют дело с центроидами.

"Contains Entire" и "Entirely Within" имеют дело с границами и не затрагивают центроиды.

Также, полилинии не могут содержать (contain) точку, но они могут пересекать (intersect) точку; точка не может быть внутри (within) полилинии, но может пересекать (intersect) полилинию.

Примечание: При этом следует помнить, что MapInfo Professional выполняет простые операции "Содержит" (Contains) и "Внутри" (Within) гораздо быстрее, чем "Содержит полностью" (Contains Entire) и "Полностью внутри" (Entirely Within). Поэтому, если Вам не обязательно точно знать, полностью ли один объект содержит другой, используйте Contains и Within вместо Contains Entire и Entirely Within.

С уважением и успехов Вам!
Карасов Андрей.

Данная статья создана на основе пунктов файла справочной системы MapInfo Professional 9.5



Функции языка MapBasic для условий выборки в MapInfo

Функция Abs()

Возвращает абсолютное значение числа (модуль).

Abs (num_expr)

num_expr - числовое выражение

Возвращаемая величина: вещественное число

Функция Abs() возвращает абсолютное значение числа, полученного в результате вычисления выражения num_expr. Если результат вычисления num_expr больше или равен нулю, Abs() возвращает результат num_expr без изменений. Если результат вычисления num_expr меньше нуля, Abs() возвращает результат num_expr, умноженный на минус единицу (-1).

Abs(-2.5) вернет 2.5

Функция Area()

Возвращает географическую площадь объекта.

Area(obj_expr, unit_name)

obj_expr - выражение, определяющее объект.

Возвращаемая величина: вещественное число, величина типа Float.

Функция Area() возвращает площадь географического объекта, определенного параметром obj_expr. Функция возвращает площадь области в единицах измерения, указанных в параметре unit_name; к примеру, чтобы получить площадь в акрах, укажите "акры" в качестве значения параметра unit_name. Список доступных единиц измерения:

Название единицы измерения	Используемая единица измерения
"acre"	акр
"hectare"	гектар
"perch"	перч
"rood"	руд
"sq ch"	квадратные чейны
"sq cm"	квадратный сантиметр
"sq ft"	квадратный фут
"sq in"	квадратный дюйм
"sq km"	квадратный километр
"sq li"	квадратный линк
"sq m"	квадратный метр
"sq mi"	квадратная миля
"sq mm"	квадратный миллиметр
"sq rd"	квадратный род
"sq survey ft"	квадратный топографический фут в США
"sq yd"	квадратный ярд

Только полигоны, эллипсы, прямоугольники и прямоугольники со скругленными углами имеют площадь. Результат применения функции к точечным, текстовым объектам, а также к прямой линии, дуге и полилинии будет равен нулю. Для скругленного прямоугольника функция Area() возвращает приблизительное значение. MapBasic вычисляет площадь прямоугольников со скругленными углами как если бы они были обычными прямоугольниками.

В большинстве случаев MapInfo Professional проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система - план, то выполняются декартовы вычисления.

В этом примере демонстрируется, как используется функция Area() для вычисления площади географического объекта. Обратите внимание, что выражение states.obj представляет географический объект из текущей строки указанной таблицы.

```
f_sq_miles = Area(states.obj, "sq mi")
```

Функция CentroidX()

Возвращает координату центральной точки (центроида) по оси X.

CentroidX(obj_expr)

obj_expr - выражение, результат которого есть величина типа Object.

Возвращаемая величина: вещественное число, величина типа Float.

Функция CentroidX() возвращает X-координату (или долготу) центроида точки объекта. Информация о координатах возвращается в текущей системе координат MapBasic; по умолчанию MapBasic использует координатную систему Долгота/Широта.

В следующем примере функция CentroidX() вычисляет долготу центроида одного географического объекта.

```
x = CentroidX(world.obj)
```

Функция CentroidY()

Возвращает координату центральной точки (центроида) по оси Y. Синтаксис аналогичен функции CentroidX().

Функция Distance()

Возвращает расстояние между двумя точками, заданными координатами.

```
Distance ( x1, y1, x2, y2, unit_name )
```

x1 и x2 - координаты по оси X (например, долгота).

Возвращаемая величина: вещественное число, величина типа Float.

Функция Distance() вычисляет расстояние между двумя определенными точками. Она возвращает значение в указанных единицах unit_name; например, чтобы получить расстояние в милях, задайте "mi" в качестве unit_name. Список доступных единиц измерения:

Значение unit_name	Используемая единица измерения
"ch"	чейны
"cm"	сантиметры
"ft"	футы (иногда называется "международным футом"; один "международный фут" равен точно 30,48 cm)
"in"	дюймы
"km"	километры
"li"	линки
"m"	метры
"mi"	мили
"mm"	миллиметры
"nmi"	морские мили (1 морская миля равна 1852 метрам)
"rd"	роды
"survey ft"	топографический фут в США (использовался при обмере территории США в 1927; один геодезический фут равен точно 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам)
"yd"	ярды

Координаты по осям X и Y должны быть заданы в текущей системе координат.
Если текущей системой координат является система координат Земли, то функция Distance() возвращает расстояние между двумя точками по дуге большого земного сечения. Расстояние вычисляется по большой окружности на сфере (большая окружность получается в результате сечения земного шара плоскостью, заданной этими двумя точками и центром Земного шара).
В большинстве случаев MapInfo Professional проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система - план, то выполняются декартовы вычисления.

```
dist = Distance(start_x,start_y,end_x,end_y,"mi")
```

Функция InStr()

Возвращает позицию первого символа подстроки в строке.

```
InStr( position, string, substring )
```

position - стартовая позиция для поиска, положительное целое число;
string - строковое выражение;
substring - строковое выражение, искомая подстрока.

Возвращаемая величина: целое число, величина типа Integer.

Функция InStr() проверяет, входит ли в состав строки string подстрока substring. MapBasic просматривает исходную строку начиная с символа с номером position; если position равно единице, то MapBasic начинает поиск с начала строки string.
Если string не содержит подстроку substring, функция InStr() возвращает ноль.
Если строка string содержит подстроку substring, то функция InStr() возвращает позицию начала подстроки в строке. Например, если substring находится в самом начале строки string, функция InStr() возвращает единицу.
Если строка substring - пустая строка, то результатом функции InStr() будет 0 (ноль).
Функция InStr() различает строчные и прописные буквы. Другими словами, функция InStr() не найдет подстроку "ИЯ" в строке "Россия" и вернет ноль.

Функция Int()

Возвращает целое число, полученное из целой части действительного числа.

```
Int( num_expr )
```

num_expr - числовое выражение

Возвращаемая величина: целое число, величина типа Integer.

Функция Int() отсекает дробную часть от действительного числа, полученного в результате вычисления выражения num_expr, и возвращает целую часть. Функция Int() возвращает ближайшее целое, меньше или равное оригиналу. Так, функция Int(-2.3) возвратит -3.

Функция Len()

Возвращает количество символов в строке или число байтов в переменной.

```
Len( expr )
```

expr - выражение; expr не может иметь тип Pen, Brush, Symbol, Font или Alias.

Возвращаемая величина: короткое целое число, величина типа SmallInt.

Информация, возвращаемая функцией Len(), зависит от типа результата выполнения expr. Если выражение expr является строкой, то функция Len() вернет количество символов в строке.

Функция ObjectLen()

Возвращает географическую протяженность объекта линии или полилинии.

```
ObjectLen( expr, unit_name )
```

expr - объектное выражение (выражение, результат которого есть величина типа Object);
unit_name - это строка, соответствующая имени единиц измерения расстояния (например, "km").

Возвращаемая величина: вещественное число, величина типа Float.

Функция ObjectLen() возвращает длину объекта. Помните, что ненулевые длины имеют только объекты линий и полилиний, для измерения периметра прямоугольника, эллипса или полигона, используйте функцию Perimeter().

Возвращаемое функцией ObjectLen() значение длины измеряется в единицах длины, определенных параметром unit_name; например, для получения длины в милях, укажите "mi" в качестве unit_name. Список допустимых единиц измерения приведен выше, в описании функции Distance(). В большинстве случаев MapInfo Professional проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система - план, то выполняются декартовы вычисления.

Функция Round()

Округляет число с заданной точностью.

```
Round( num_expr, round_to )
```

num_expr - числовое выражение;
round_to - число, определяющее точность округления переменной num_expr.

Возвращаемая величина: вещественное число, величина типа Float.

Функция Round() возвращает округленное значение от числа, полученного в результате вычисления выражения num_expr.

Число round_to задает точность округления. Функция Round() округляет переменную num_expr до ближайшего, кратного round_to. Например, если параметр round_to равен 0.01, то число num_expr будет округлено до сотых. Если round_to равен 5, то MapInfo Professional вернет значение, ближайшее к num_expr и кратное пяти.

Функция Val()

Возвращает численную величину, извлеченную из строки.

```
Val( string_expr ), где
```

string_expr - выражение, результат которого есть строка.

Возвращаемая величина: вещественное число, величина типа Float.

Функция Val() возвращает число, выделяя его из строки, определенной выражением string_expr. Считывание числа начинается с начала строки и заканчивается первым нечисленным символом. При этом функция Val() игнорирует пробелы, символы табуляции и новой строки в начале строки string_expr. Если первый символ строки не является числом, точкой, знаком минус или плюс, амперсандом (&), то функция Val() вернет 0. Амперсанд используется для шестнадцатеричных чисел.

Примечание: Если строка включает разделитель целой части числа и десятичной, то этот знак должен быть точкой, независимо от того, какой стандарт форматирования чисел используется в компьютере пользователя. Строка также не должна содержать разделители тысяч.

```
Val("12 thousand") равна 12
```

```
Val("12,345") равна 12
```

```
Val(" 52 - 62 Brunswick Ave") равна 52
```

```
Val("Eighteen") равна 0
```

```
Val("&H1A") равна 26 (шестнадцатеричное 1A)
```

Здесь представлены описания функций, на мой взгляд, наиболее часто используемых в построении выборок. Главное для Вас - уяснить принципы подачи средств языка программирования в системе справки данного языка. Описания остальных, в подобном виде, Вы без труда сможете посмотреть в файлах справки, руководстве MapBasic, также в руководстве MapInfo Professional 9.5, предоставленных для скачивания в конце данного цикла статей.

С уважением и успехов Вам!
Карасов Андрей.

Данная статья создана на основе пунктов файла справочной системы MapInfo Professional 9.5