Postings to GeoReference:


On August 18th, 2006 Adam suggested that we begin a GeoReference thread devoted to small SQL examples (thread below). This MS Word XP (2002) document is an attempt to organize various SQL examples posted to GeoReference and Manifold-L. It is hoped that the Table of Contents will allow people to navigate quickly to their area of interest.

Initially, I kept all the related postings and there was considerably more information provided then just the basic SQL. I have stripped some of that out in the following summary; however, there are many entries that are rather long and contain a lot of the back and forth commentary which I found helpful.

There are many personal annotations that are specific to my interests (marked with "LAK" or "L. Ketch", etc.). I have not made any attempt to remove them before posting to GeoReference – it is just too time consuming to keep two copies of this document 😊

L. Ketch – Oct 2, 2006

adamw at 8/18/2006 6:36 AM (#26107)
http://69.17.46.171/Site/Thread.aspx?id=25234&ti=632926356671030000

………..
I'm going to start saving all the queries I run across in this forum and save them in a .map file.

How about posting them into a separate thread on this forum, with a small annotation for each query? If we had such a thread, it would sooner or later grow into a separate section of the site dedicated to sharing code snippets, and then possibly into a snippet sharing service integrated into the product! :-)
Document Formatting

- Word Styles have been created for:
  
<table>
<thead>
<tr>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
</tr>
</tbody>
</table>

- Standard Word Heading Style names like "Heading 1" are used throughout. This allows one to build an automatic Table of Contents.

- There are many personal annotations generally indicated by "LAK" or "L. Ketch"

- I have added the URL back to the original threads. This will allow you to obtain the full text if required.

- I have tried to credit each author. Generally the author's name and posting details will appear in Word style "Author" as shown above. I apologize up front if I missed anyone.

- The Titles used are often my invention. I was trying to come up with clear explanations of what each SQL example does so that the TOC could be used to quickly find the area of interest. This may be clear to me but is sure to be confusing to others. You can easily change these titles and then recompose the TOC.

To regenerate the TOC

Right click inside of the TOC, then choose "Update Field" from the context menu.

Select "Update entire table"

Click OK

If you hover the mouse cursor over a TOC page number you will see something like:

CTRL + click on the page number will take you directly to the topic.
Table of Contents

SQL REFERENCE TEXTS AND VIDEOS.................................................................1

QUERIES USING MULTIPLE TABLES (TAKEN DIRECTLY FROM MANIFOLD® V7 HELP FILES).....1

OVERVIEW (INNER AND OUTER JOINS) .................................................................1
INNER JOIN Operator (Details) ..............................................................................4
OUTER (LEFT, RIGHT, FULL) JOIN Operators (Details) ........................................5
CROSS JOIN Operator (Details) ............................................................................7

GENERAL METHODOLOGIES ........................................................................8

WRITING COMPLEX QUERIES – CREATE IN PARTS .................................................8
ADDING A SQL SELECTION TO A DRAWING .......................................................8
TOGGING AUTOMATIC SELECTION OF OBJECTS AND RECORDS IN QUERIES ........9
CALLING A QUERY COMPONENT FROM A SECOND QUERY COMPONENT ..............10

SQL SPEED TRICKS ..................................................................................11

SQL SURFACE SELECTION USING BUFFERED LINES TAKES FOREVER – VARIOUS APPROACHES TO REALLY SPEEDING UP THIS QUERY ......................................................11
CIRCUMVENTING SLOW "=" ANY ()" AND "IN ()" CLAUSES – USE JOINS ................13

DISTANCE CALCULATIONS ........................................................................13

NEAREST CLINICS TO CLIENTS ......................................................................13
SELECT A SURFACE BASED ON THE DISTANCE FROM A POINT .........................14
FIND THE CLOSEST NEIGHBOUR TO EACH POINT IN A POINT DRAWING ..........14
SELECT SURFACE PIXELS WITHIN A CERTAIN DISTANCE OF SELECTED PIXELS ....15
SELECT OBJECTS WITHIN A GIVEN DISTANCE ...............................................15
SELECT PARCELS WITHIN 500 FEET OF THE CURRENTLY SELECTED PARCEL ....16
FIND THE DISTANCE (IN MILES) BETWEEN ALL POINTS IN TWO DRAWINGS ....17
SELECTING FROM A SURFACE, BASED ON THE DISTANCE FROM A SET OF POINTS .........18
FIND THE DISTANCE FROM A TOWER TO THE NEAREST URBAN CENTRE ..............18
FIND WHICH POLYLINE SEGMENT LIES CLOSEST TO A GIVEN POINT LOCATION ....19
FINDING THE DISTANCE BETWEEN EVERY LINE NODE IN A DRAWING ................19
DETERMINE DISTANCE FROM POINT TO NEAREST AREA OF CERTAIN TYPE ........21
FIND THE CLOSEST ROAD TO EACH TOWER .....................................................21
UPDATE SURFACE, ASSIGN EACH PIXEL WITH THE DISTANCE TO A DRAWING ......22
DETERMINE THE DISTANCE DOWN A LINE TO A POINT (THAT MAY NOT LIE DIRECTLY ON THE LINE) ...............................................................23
ALSO SEE: "FIND NEAREST & FARthest OBJECT FROM A GROUP AND CREATE LINES TO THEM" .................................................................24

SELECTING CONTAINED AND/OR BOUNDED OBJECTS ..................................24

SELECT SURFACE PIXELS THAT LIE WITHIN AREAS ....................................24
SELECT RECTANGLES THAT CONTAIN BUFFERED CIRCLES ..............................26
CREATE A SEPARATE AREA BOUNDING EACH CLOSED LINE IN A DRAWING ........26
SELECTS ALL STATES CONTAINING AT LEAST ONE POINT FROM THE DRAWING CITIES 27
CREATE A BOX AROUND A GIVEN COORDINATE ..........................................28
CREATE RECTANGLES ENCLOSING POINTS WITH THE SAME ATTRIBUTE ..........28
ANALYSE LAND USE WITHIN A BUFFERED AREA ...........................................29
FIND THE TOTAL LINE LENGTHS INSIDE A NUMBER OF INDIVIDUAL AREAS ......31

CREATING LINES .....................................................................................33

CONVERT TRACK POINTS TO LINES – SUCCESSIVE RECORD CALCULATIONS ......33
CREATE A LINE GEOM BETWEEN TWO POINTS ON EACH ROW .....................42
CREATING LINES FROM POINTS (USING WKT) ..................................................43
HOW TO USE THE “CONVERTToLINE” SQL FUNCTION ..................................43
CROSS TABULATION MATRIX TO COMPARE LANDCOVER TYPES (PIVOT TABLE) .......................81
FIND LINES THAT ARE ADJACENT TO A POLYGON, AND THEN SUM THE RESULTS .................82
SELECT A UNIONED GEOM AND AN AREA SUM FOR EACH UNIQUE ATTRIBUTE ..................82
SPLIT A LINE WITH SQL ........................................................................................................83
INTERPOLATE HEIGHTS ACROSS A SERIES OF POINTS THAT REPRESENT A STREAM SEGMENT ..................................................................................................................................................83
AVERAGE VALUE OF AN ATTRIBUTE SHARED BY A GROUP OF POLYGONS, WEIGHTED BY THEIR AREAS ..................................................................................................................................................85
OBTAINING THE LAT/LONG COORDINATES OF AN IMAGE IN RELATIONSHIP TO THE MAP DISPLAY CONTAINING THAT IMAGE .............................................................................................................85
SQL Reference Texts and Videos

SQL text recommended by liofr on GeoReference on Aug 21, 2006 (#26246)
http://69.17.46.171/Site/Thread.aspx?id=26220&ti=632926356630100000

Advanced SQL Programming
by Joe Celko
3rd edition in English
Published August 2005
808 pages
ISBN 0123693799

Art Lembo's Training Video's
http://www.gisadvisor.com/

Text recommended by Art Lembo in his free "spatial SQL in Manifold©" video.
See:
http://dspace.library.cornell.edu/handle/1813/2489

Art mentioned a text titled: "GeoGraphic and Spatial Analysis" by O'Sullivan and Unwin. I could not find that specific title; however, the following text by these authors is available.

Geographic Information Analysis Wiley: Hoboken, NJ

willh on 6/27/2006 6:22 PM (#23429)
http://69.17.46.171/Site/Thread.aspx?id=23351&ti=632870293446070000

I cannot recommend one (good SQL reference – LAK), but I can tell you that the Manifold SQL engine, though proprietary, is most like JET SQL (i.e. Access); so keep that in mind while hanging out at the library---I mean Borders.

I can tell you that if you want to get into the heavier stuff, pick up a book by C. J. Date (what is with these SQL gurus and two initials & last name?); in particular, I like Database in Depth--it's great theory book.

Queries Using Multiple Tables (taken directly from Manifold© V7 Help Files)

Overview (Inner and Outer Joins)

Queries can incorporate fields from two different tables. There are several methods for doing so depending on the structure of the tables and their contents.
This is the easiest situation is when the two tables have a key field in common. We can then combine them by using an **inner join** between the tables. This is also called an **equi-join** in some database management systems.

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] INNER JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The **inner join** operation combines two tables (in the above case, the Customers table and the Orders table) using a key field they have in common. The example shown above will create a table that takes the Company Name field from the Customers table and shows it with the Employee ID field from the Orders table. It will create a table for all records where there are both customers and orders.

We can also create a similar query that will show all customer records in the results even if there are no orders for some customers. We might also want to run the query in a way that includes all records in the orders table even if there are no customers for some orders. This should not occur (a well-run business should not be recording orders for which there are no customers), but it is a way of checking for errors. To do this, we use an *"outer join"* which is phrased as either a **left join** or a **right join**.

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] LEFT JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The **left join** operation will include all of the records from the left-hand table mentioned, the Customers table in the example above, even if there are no matching values to records in the right hand table (the Orders table in the example above).

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] RIGHT JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The right join operation will include all of the records from the right-hand table mentioned, the Orders table in the example above, even if there are no matching values to records in the left hand table (the Customers table in the example above).

**Simplified Syntax for Inner Joins**

Inner joins can be created by default in Manifold using a simplified syntax such as the following:

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers],[Orders]
WHERE [Customers].[Customer ID] = [Orders].[Customer ID]
```
This simplified syntax replaces the inner join and on words with a comma in the list of tables involved and a simple where clause. Many SQL systems can use the above syntax. Access uses the inner join syntax because the "inner join" construct is used as a hint to the Jet database engine within Access to build temporary indices that will be used to optimize the query. Manifold SQL optimizes both inner join and where constructs so the above simplified syntax works just as fast as using inner join.

SQL can also be used to combine fields from more than two tables, and even from multiple tables that do not have a key field in common. However, such SQL topics are beyond the introductory level of this help documentation. For advanced SQL discussion we suggest readings in one of the many good books oriented to SQL in a Microsoft Windows environment.

**The Identity Command**

Right clicking on a column head in a table created by a query pops open a context menu from which we can choose the Identity command. The Identity command allows us to select the ID column to be used in tables that have more than one identity column.

Suppose we have two drawings, one of cities and one of states, called Cities and States. The following query…

```sql
SELECT * FROM Cities, States
WHERE Contains(States.ID, Cities.ID);
```

…will generate a table with two identity columns: States.ID and Cities.ID.

Suppose the query generates a record containing a New York city and a state of New York. Let's also suppose that New York city is selected and the state of New York is not. The selection state of the record that is linked to both of these objects is determined by the active identity column. The record will appear selected if the active identity column is set to Cities.ID and unselected if the active identity column is set to States.ID.

The active identity column also specifies how selection in the table selects objects in the associated drawings. If the Cities.ID column is made the active identity column then selecting a record in the table will select the corresponding object in the Cities drawing. If the States.ID column is made the active identity column, then selecting a record in the table will select the corresponding object in the States drawing.

**Note**

When removing components contributing to a content of a query Manifold tries to preserve the query output if a query table is open. Suppose we select data from two tables in a query and the resulting query table shows some columns from the first table and some columns from the second table. If we delete the first table while the query is open, the cells of the resulting table that belong to the first table will become empty but the cells that belong to the second table will
INNER JOIN Operator (Details)

Combines records from two tables.

Syntax:

    table [INNER] JOIN table ON condition

The INNER JOIN operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
<tr>
<td>condition</td>
<td>A Boolean expression restricting the records in the resulting table.</td>
</tr>
</tbody>
</table>

Remarks:

We can use an INNER JOIN operator in any FROM clause. For example, we can use INNER JOIN with the "Categories" and "Products" tables to select all products in each category. In contrast, to select all categories (even if some categories have no products) or all products (even if some do not belong to any category), we need to use a LEFT JOIN or RIGHT JOIN operator to create an outer join.

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column:

    SELECT [Category Name], [Product Name] FROM [Categories] INNER JOIN [Products] ON [Categories].[Category ID] = [Products].[Category ID];

We can use more than one column in a join condition.

We can nest JOIN statements using the following syntax:

    table join (table join (table ...) ON condition) ON condition

Examples:

This example uses the INNER JOIN operator to select all orders assigned to Andrew Fuller:
SELECT [Order ID], [Order Date]
FROM [Orders] INNER JOIN [Employees]
ON [Employees].[Employee ID] = [Orders].[Employee ID]
    AND
[Employees].[First Name] = "Andrew"
    AND
[Employees].[Last Name] = "Fuller";

The word "INNER" can be omitted, so the above query can be rewritten as:

SELECT [Order ID], [Order Date]
FROM [Orders] JOIN [Employees]
ON [Employees].[Employee ID] = [Orders].[Employee ID]
    AND
[Employees].[First Name] = "Andrew"
    AND
[Employees].[Last Name] = "Fuller";

This example uses nested INNER JOIN operators and GROUP BY clause to produce a list of employees and their total sales using "Employees," "Orders," and "Order Details" tables:

SELECT Sum([Unit Price] * [Quantity]) AS [Sales],
    ([First Name] & " " & [Last Name]) AS [Name]
FROM [Employees] INNER JOIN ([Orders] INNER JOIN [Order Details]
    ON [Order Details].[Order ID] = [Orders].[Order ID])
    ON [Orders].[Employee ID] = [Employees].[Employee ID]
GROUP BY ([First Name] & " " & [Last Name]);

**OUTER (LEFT, RIGHT, FULL) JOIN Operators (Details)**

Combine records from two tables allowing NULL values on left, right, or both sides.

**Syntax:**

```
    table {LEFT | RIGHT | FULL} [OUTER] JOIN table ON condition
```

The LEFT JOIN, RIGHT JOIN, and FULL JOIN operators have these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
<tr>
<td>condition</td>
<td>A Boolean expression restricting the records in the resulting table.</td>
</tr>
</tbody>
</table>
Remarks
Use a LEFT JOIN operator to create a left outer join. Left outer joins include all of the records from the first (left) of two tables, even if there are no matching values for records in the second (right) table.

Use a RIGHT JOIN operator to create a right outer join. Right outer joins include all of the records from the second (right) of two tables, even if there are no matching values for records in the first (left) table.

Use a FULL JOIN operator to create a full outer join. Full outer joins include all of the records from both tables, even if there are no matching values for records in the other table.

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column to produce a list of all categories, including those that contain no products. The GROUP BY clause and the First aggregate function are used to collapse multiple records for products sharing the same category):

```
SELECT [Category Name], First([Product Name])
FROM [Categories] LEFT JOIN [Products]
ON [Categories].[Category ID] = [Products].[Category ID]
GROUP BY [Category Name]
```

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column to produce a list of all products, including those that contain no categories:

```
SELECT [Category Name], [Product Name]
FROM [Categories] RIGHT JOIN [Products]
ON [Categories].[Category ID] = [Products].[Category ID]
```

To combine records disallowing NULL values on any side of a join, use an INNER JOIN operator.

We can use more than one column in a join condition.

When nesting any of the LEFT JOIN, RIGHT JOIN, and FULL JOIN operators with one or more of the INNER JOIN operators, for performance purposes it is recommended that we put LEFT JOIN, RIGHT JOIN, and FULL JOIN operators inside the INNER JOIN operators.

Examples:
This example uses the LEFT JOIN operator and the GROUP BY clause to select employees and the number of orders they processed in June, 1991:
SELECT [First Name] & " " & [Last Name] AS [Name],
    Count([Order Date]) AS [Order Count]
FROM [Employees] LEFT JOIN [Orders]
ON [Orders].[Employee ID] = [Employees].[Employee ID]
    AND
    [Order Date] >= #6/1/1991#
    AND [Order Date] < #7/1/1991#
GROUP BY [First Name] & " " & [Last Name];

This example uses the LEFT JOIN operator, several INNER JOIN operators, and the GROUP BY clause to show whether a given product has or has not been processed at least once by Andrew Fuller:

SELECT [Product Name], "Yes" AS [Served]
FROM [Products] LEFT JOIN ([Order Details] INNER JOIN ([Orders] INNER JOIN [Employees]
ON [Orders].[Employee ID] = [Employees].[Employee ID]
    AND [First Name] = "Andrew"
    AND [Last Name] = "Fuller")
ON [Order Details].[Order ID] = [Orders].[Order ID])
ON [Order Details].[Product ID] = [Products].[Product ID]
GROUP BY [Product Name];

**CROSS JOIN Operator (Details)**

Combines records from two tables without any conditions.

**Syntax:**

table CROSS JOIN table

The CROSS JOIN operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
</tbody>
</table>

**Remarks:**

A CROSS JOIN operator is analogous to an INNER JOIN operator without a condition.

A CROSS JOIN operator is frequently used with a WHERE clause that filters the resulting records using criteria that would be not be used within an INNER JOIN.
A CROSS JOIN operator is also frequently used with the same table. In this case, it is often necessary to provide an alternate name for one of the copies of the table, so that the names of columns in the two copies of the table are no longer ambiguous.

**Example**

This example uses the CROSS JOIN operator and a WHERE clause to select the orders that were shipped later than those orders made after them (the DISTINCT operator collapses records for the same order into one record):

```
SELECT DISTINCT [Orders].[Order ID]
WHERE [Orders].[Order Date] < [Copy].[Order Date]
   AND
   [Orders].[Shipped Date] > [Copy].[Shipped Date];
```

**General Methodologies**

*Writing Complex Queries – Create in Parts*

http://69.17.46.171/Site/Thread.aspx?id=13388&ti=632588328000000000

artlembo on 8/5/2005 10:00 AM (#13393)

Oftentimes, that is how I write more sophisticated SQL - I create two queries, and then I wrap one of the queries in parenthesis and insert it after the FROM clause as you did.

"Don't eat the elephant in one bite!"

*Adding a SQL Selection to a Drawing*

http://69.17.46.171/Site/Thread.aspx?id=17377&ti=632719561943570000

adamw at 12/28/2005 9:51 PM (#17455)

The MapInfo's way of selecting drawing objects into a separate drawing may indeed be slightly shorter than Manifold's. Having said that, the Manifold's way has its advantages. After you create a SELECT query (not SELECT INTO), you can import it as a drawing using File - Import - Drawing, This Project (select the query in the "Table / query" combo, make sure the "Use" combo is set to "geometry column" and the "Geometry" column is set to "Geom (I)", and press OK). The best part of this is that if you replace File - Import - Drawing with File - Link - Drawing, the produced drawing will maintain the link to the query, and you will be able to refresh it using View - Refresh Data after making changes to the original data.
I tried this using the HURDAT file "NHCBestTracks1851To2004 Table" with a single grid cell in a separate table "OneCell". The Select structures below seem to work. They can be linked back into a map using Adam's instructions above.

One can use either the "Geom (I)" or "ID" columns in the "Where Touches" clause. Note however, that "[NHCBestTracks1851To2004 Table].[Geom (I)]" column MUST be included in the Selected fields. It is "[Geom (I)]" or [ID] that carries the track nodes and the Selected tracks cannot be drawn without these instructions.

```sql
SELECT [OneCell].[Cell_Name], DatabaseStormNumber, Name_E, Date_E, Hour.UTC, MaxWind_kts, SS_E, SystemSpeed_kts, RecordNumber,
[NHCBestTracks1851To2004 Drawing].[Geom (I)]
From [NHCBestTracks1851To2004 Drawing], OneCell
Where Touches ([NHCBestTracks1851To2004 Drawing].[Geom (I)], [OneCell].[Geom (I)])
Order by DatabaseStormNumber, MaxWind_kts DESC, RecordNumber;

SELECT [Grid_1degree Table].[Cell_Name], DatabaseStormNumber, Name_E, Date_E, Hour.UTC, MaxWind_kts, SS_E,
    SystemSpeed_kts, RecordNumber, [NHCBestTracks1851To2004 Table].[Geom (I)]
From [NHCBestTracks1851To2004 Table], [Grid_1degree Table]
where Touches ([NHCBestTracks1851To2004 Table].[ID], [Grid_1degree Table].[ID])
    AND [Grid_1degree Table].[Cell_Name] = "41.5N 065.5W"
Order by DatabaseStormNumber, MaxWind_kts DESC, RecordNumber;
```

---

**Toggling Automatic Selection of Objects and Records in Queries**

http://69.17.46.171/Site/Thread.aspx?id=17367&ti=632711954316730000

**adamw at 12/23/2005 10:14 PM (#17373)**

Selecting objects and records returned by a query is a purely UI thing, done for the user's convenience. You can turn it off in Tools - Options, Miscellaneous (uncheck the "Automatically select query records" option). To select objects returned by a query from within a script, convert a SELECT query into an equivalent UPDATE query which sets the [Selection (I)] column, eg:

```sql
--SQL
UPDATE (SELECT [Stations].[Selection (I)] FROM ... WHERE ...) SET [Selection (I)] = True;
```

Prior to running the above query, unselect all objects using:
--SQL
UPDATE [Stations] SET [Selection (I)] = False;

Adam also provided some elaboration on Manifold©-L on Apr 26, 2006
The automatic selection of objects returned by queries is a purely UI thing that is done after the query has run by the table window. To achieve the same script effect, either do Query.Table.Open, or rewrite the SELECT query into an equivalent UPDATE query, which would set the Selection (I) column for the objects you want to select to True.

Yet a further refinement that can be used in some situations suggested by Adam on Manifold©-L on Apr 27, 2006

Rather Than:

```
UPDATE [Weed Drawing]
    SET [Weed Drawing].[Selection (I)] = True
    WHERE [Switch 1] = 1
```

To Deselect Objects with Switch 1 different from 1, it might be better to use:

```
UPDATE [Weed Drawing] SET [Selection (I)] = ([Switch 1] = 1)
```

(This is really neat. The ([Switch 1] = 1 will return TRUE or FALSE depending upon whether ([Switch 1] is 1 (TRUE) or something else. All records are reset. LAK)

**Calling a Query Component from a Second Query Component**

http://69.17.46.171/Site/Thread.aspx?id=23452&ti=632870712471870000

As described by WillH below, Manifold© SQL can use the name of a Query Component in the same manner as a regular table name – you can link queries.

Example:
SELECT * INTO [List] FROM [Initial]
Where "Initial" is the name of a Query Component which might contain Select * from [Seq]

Willh on 6/28/2006 6:00 AM (#23462)
Write the transform query as one component, we'll call it [my transform], then run another query component with SELECT * INTO [new one] FROM [my transform]
SQL Speed Tricks

SQL Surface Selection Using Buffered Lines Takes Forever – Various Approaches to Really Speeding up this Query

http://69.17.46.171/Site/Thread.aspx?id=20793&ti=632829585442530000

(there is some good reading here – LAK)

paquet at 5/2/2006 12:39 PM (#20794)

The following query takes quite a long time to run. Is there something we could modify to make it quicker? The same query, using areas instead of buffered lines, runs a lot faster...

OPTIONS CoordSys("DrawingA" AS COMPONENT);
UPDATE [SurfaceA] SET [Selection (I)] = True
WHERE
Intersects (SELECT buffer(AllBranches([id]),20) as [daBuffer] from [DrawingA]
WHERE [Selection (I)] = True AND [type (I)]=2,
NewPointLatLon([SurfaceA].[Longitude (I)], [SurfaceA].[Latitude (I)]))

This one works great! It uses areas to select the surface:

OPTIONS CoordSys("DrawingA" AS COMPONENT);
UPDATE [SurfaceA] SET [Selection (I)] = True
WHERE
Intersects ((SELECT unionall([ID]) from [DrawingA] WHERE [Selection (I)] = True AND [Type (I)]=3),
NewPointLatLon([SurfaceA].[Longitude (I)], [SurfaceA].[Latitude (I)]))

adamw at 5/3/2006 6:37 AM (#20813)

The OPTIONS clause is unnecessary, since you do not link the query as a component.

The reason the query is slow is that it recomputes the buffer returned by the inner SELECT for each pixel. The query engine is smart enough to cache the selected lines the SELECT operates upon, but it is not smart enough to cache the result of the aggregate on these lines. The solution is to get rid of computations in the column list of the SELECT, or, better yet, get rid of the SELECT in favor of a JOIN.

Here are some numbers I obtained using hand-made data.

Query 1 - a slightly modified replica of the original query, the surface is named "S" and the drawing "D" - about 2500 seconds:
--SQL
UPDATE [S] SET [Selection (I)] = True
WHERE Intersects(SELECT Buffer(AllBranches([ID]), 20) FROM [D] WHERE [Type (I)] = 2,
    NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)])))

Query 2 - computations in the column list are reduced to a minimum - 29 seconds:

--SQL
UPDATE [S] SET [Selection (I)] = True
WHERE (SELECT Min(Distance([ID], NewPointLatLon([S].[Longitude (I)],
    [S].[Latitude (I)])))
FROM [D] WHERE [Type (I)] = 2) <= 20

Query 3 - SELECT is replaced with an INNER JOIN - 3 seconds:

--SQL
UPDATE (SELECT [S].[Selection (I)] FROM [S] INNER JOIN [D] ON
    Distance([D].[ID], NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)]))
<= 20)
SET [Selection (I)] = True

paquet on 5/3/2006 9:46 AM (#20822)
Based on Query 3, the following creates buffers for lines only:

UPDATE (SELECT [S].[Selection (I)] FROM [S] INNER JOIN [D] ON
Distance([D].[ID], NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)]))
<= 20
where [Type (I)]=2)
SET [Selection (I)] = True

It takes a shorter time to run, but the results of Query 2 are of better quality. In Query 3, part of
the buffers around lines ends are missing. It's like something took a bite off some of the buffers.
This looks like a bug. Maybe not. I wish I could post a .map file...

adamw at 5/4/2006 6:33 AM (#20841)
I can not see the difference on my data. Could you upload your MAP file to one of the free file
sharing services, such as turboupload.com and post the link to the file?

ejburn at 5/4/2006 7:08 PM (#20854)
A side note regarding speed (little off topic - I apologise). I would suggest that one always look
at and set a component’s precision to the desired accuracy. I have found that almost all operations
benefit from this in terms of speed.)
I could not agree more. According to tech support, using an overly small value of the location precision parameter is the #1 reason for the performance not being quite what it could be that they see in the lab, surpassing even the failure to use the same projection for all data layers involved in the analysis, when that is feasible (!).

Circumventing Slow "= ANY ()" and "IN ()" Clauses – Use Joins

Processing speed is greatly increased by using Joins rather than the "= ANY ()" and "IN ()" operators as illustrated by Robert Sanson below.

Sent: Mon 03/27/2006 7:11 PM
From: Robert Sanson [SansonR@agriquality.com]
To: lketch@hfx.eastlink.ca; Manifold-L@lists.directionsmag.com
Subject: Re: [Manifold-l] SQL in Script vs. Query Component


Hi Lorne

Responding to your final question about speed, if you do a left outer join and then select values that are not NULL, the query will be faster:

```
SELECT [HURDAT Drawing].*
From [HURDAT Drawing] LEFT JOIN [InitialSelection]
ON [HURDAT Drawing].[RecordNumber] = [InitialSelection].[RecordNumber] 
AND [InitialSelection].[RecordNumber] IS NOT NULL;
```

Cheers,

Robert Sanson

Distance Calculations

Nearest Clinics to Clients

http://69.17.46.171/Site/Thread.aspx?id=27305&ti=632933335734000000

Hi!
...We need to find the nearest clinic to them (our clients) (or up to 5 of the nearest), straight line distance, including mileage...

-Beth
Why not use a query like this:

PARAMETERS Address text, City text, State text, Zip text,
[Search Radius in Miles] integer, [Number of Locations] integer;

SELECT TOP [Number of Locations] *
FROM (Select DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') as [Distance to Location], [Health Clinics Table].*
FROM [Health Clinics Table]
WHERE DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') < [Search Radius in Miles])
ORDER BY [Distance to Location];

Brian

---

**Select a Surface Based on the Distance from a Point**

http://69.17.46.171/Site/Thread.aspx?id=17386&ti=632711894743770000

adamw at 12/26/2005 10:07 AM (#17388)

---SQL

UPDATE (SELECT S.[Selection (I)] FROM [Elevation] S, [Cities] D
WHERE Distance(AssignCoordSys(
    NewPoint(S.[Center X (I)], S.[Center Y (I)]),
    CoordSys("Elevation" AS COMPONENT)),D.ID, "m")
    BETWEEN 0 AND 80)
SET [Selection (I)] = TRUE;

---

**Find the Closest Neighbour to each Point in a Point Drawing**

http://69.17.46.171/Site/Thread.aspx?id=13388&ti=632588328000000000

Johnny on 8/5/2005 9:07 AM (#13389)

I have a point layer and want for each point its closest neighbouring point.

Johnny on 8/5/2005 9:27 AM (#13392)

OK - embarrassed again I tell you that I found a way in the end:

my final query looks like:
SELECT pt1.ID, pt2.ID, Dist
FROM (SELECT pt1.ID AS ID_Pt1,
MIN(Distance(pt1.[ID], pt2.[ID])) AS [Dist]
FROM points as pt1, points as pt2
WHERE pt1.id <> pt2.id
GROUP BY pt1.ID) as interm_q, points as pt1, points as pt2
WHERE Distance(pt1.ID, pt2.ID)= [interm_q].[Dist] and [interm_q].ID_Pt1 = pt1.ID

Select Surface Pixels within a Certain Distance of Selected Pixels

http://69.17.46.171/Site/Thread.aspx?id=25234&ti=632926356671030000

Mike Pelletier at 8/15/2006 9:17 AM (#25952)
I couldn’t figure out how to select surface pixels within a certain distance of selected pixels. If surfaces had a Geom it would make it simple like it is for drawings.

adamw at 8/17/2006 6:42 AM (#26035)
You could do this with a query as well:

--SQL
UPDATE ( SELECT S.[Selection (I)] X
FROM [Surface] S, [Surface] T
WHERE T.[Selection (I)] AND NOT S.[Selection (I)] AND Distance( 
    NewPoint(S.[Center X (I)], S.[Center Y (I)]), 
    NewPoint(T.[Center X (I)], T.[Center Y (I)]) < 10
) ) SET X = True

Select Objects within a Given Distance

http://69.17.46.171/Site/Thread.aspx?id=18244&ti=632737959890130000

artlembo at 1/25/2006 10:42 AM (#18247)
use SQL:

select * from drawing where 
    distance(drawing.id,(newpoint(-52,9.5))) < 300

bnhilton at 1/25/2006 2:26 PM (#18259)

Or to get really fancy:
PARAMETERS Address text, City text, State text, Zip text, [Search Radius in Miles] integer;

SELECT DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') as [Distance to Location], [Your Table].*
FROM [Your Table]
WHERE DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') < [Search Radius in Miles]
ORDER BY [Distance to Location];

Select Parcels within 500 feet of the Currently Selected Parcel

http://69.17.46.171/Site/Thread.aspx?id=26568&ti=632925212321370000

Mike Pelletier at 8/28/2006 2:03 PM (#26569)
I'm trying to write a query that allows the user to select parcels within 500 feet of a currently selected parcel.............
Mike

seatrails at 8/28/2006 2:17 PM (#26570)
........Try:

Select [parcels].* from [parcels], [parcels 2] WHERE
Distance([parcels].[id], [parcels 2].[id]) <=500
AND [parcels 2].[selection (I)]
AND [parcels].[id] <> [parcels 2].[id]

artlembo at 8/29/2006 3:53 PM (#26663)
Mike,

to make it work, probably do something like:

Select * from [Parcels] WHERE
Distance([Parcels].[ID],
(Select ID from Parcels Where [Parcels].[Selection (I)] = True)) <= 500

seatrails at 8/30/2006 10:17 AM (#26711)
This always returns the original parcel too.

Mike Pelletier at 8/30/2006 11:20 AM (#26716)
Your right. Here's code that doesn't select the original parcel.
select * from [parcels] WHERE
distance([parcels].[id], (select id from parcels where
[parcels].[selection (I)] = true)) <= 500
EXCEPT select * from parcels where
[parcels].[selection (I)] = true

Also, here's code that allows the user to type in a parcel number instead of selecting it.

parameters [Parcel Number] text;
select * from parcels WHERE
distance (parcels.id, (select id from parcels where parc_num =
[Parcel number])) <= 500
EXCEPT select * from parcels where parc_num =
[Parcel number]

This SQL stuff is kind a fun when you get the hang of it!

artlembo at 8/30/2006 11:32 AM (#26717)
great! another convert!

Mike, you may want to try this and see if it speeds things up:

parameters [Parcel Number] text;
select * from parcels WHERE
distance (parcels.id, (select id from parcels where parc_num =
[Parcel number])) <= 500
AND parc_num <> [Parcel number]

that way you may be avoiding another selection into the whole table.

Find the Distance (in miles) Between All Points in Two Drawings

http://69.17.46.171/Site/Thread.aspx?id=26643&ti=632924946083230000

artlembo at 8/29/2006 1:38 PM (#26659)

Select Distance(d1.id, d2.id, "mi") from d1, d2

adamw at 8/29/2006 8:57 PM (6 hours ago) (#26670)
It is possible once you project the drawing you want to use from lat/lon to, say, Mercator.
Selecting from a Surface, Based on the Distance from a Set of Points

http://69.17.46.171/Site/Thread.aspx?id=19179&ti=632767831363070000

johnrobot at 3/1/2006 4:12 AM (#19275)
Mike, if you are trying to select from a surface, based on the distance from a set of points, the following code could help you. I tried to do it a while ago and I had some serious help from this forum, thanks again for that. Anyway, here it is:

UPDATE (SELECT S.[Selection (I)] FROM [Elevation] S, [Cities] D
WHERE Distance(AssignCoordSys(
NewPoint(S.[Center X (I)], S.[Center Y (I)]),
CoordSys("Elevation" AS COMPONENT)), D.ID, "km") BETWEEN 0 AND 3)
SET [Selection (I)] = TRUE;

Magnus

Find the Distance from a Tower to the Nearest Urban Centre

http://69.17.46.171/Site/Thread.aspx?id=6455&ti=632362718400000000

KevinMunro on 11/16/2004 6:46 AM (#6456)
Hello, I have an urban table with 339 records and a towers table with 21000 records.

I want to get the distance to the nearest urban area.......... Kevin.

abram on 11/16/2004 1:21 PM (#6458)
Hi

First of all, you don't need any aggregate function to retrieve the name of the tower... therefore your query becomes as follow.

Second if your column 'name' is set as text type the min aggregation function will return the smaller one..

SELECT [Towers 2].[Name] AS Tower,
MIN(Distance([Urban 2].[ID], [Towers 2].[ID], "m")) AS [Dist from Urban]
FROM [Towers 2], [Urban 2]
GROUP BY [Towers 2].[Name]

abram
Kevin Munro on 11/16/2004 1:46 PM (#6459)
Thanks Abram, I didn't think to try that!

I don't know why I assumed I had to put a MIN around the first column. I think that's what I would have to do in SQL Server and I'm just used to it. When you get into a mindset it's sometimes hard to get out of it.

If I wanted to take the MIN (distance) and also the name of the urban area bringing back the MIN (distance) do you know how I'd do that?

Is this approach valid for getting the nearest point to a reference point or is there something like a NEAREST spatial function?

Kevin.

Abram on 11/17/2004 3:42 AM (#6485)
To get over the city which is the nearest and not only the distance you'll have to create a new query calling the result of the first one. It becomes the query as follow.

FirstQuery:

```sql
SELECT [Towers 2].[Name] AS Tower,
MIN(Distance([[Urban 2].[ID]], [Towers 2].[ID], "m")) AS [Dist from Urban]
FROM [Towers 2], [Urban 2]
GROUP BY [Towers 2].[Name]
```

SecondQuery:

```sql
SELECT Tower, [Dist from Urban], [Urban 2].[ID]
FROM [FirstQuery], [Urban 2], [Towers 2]
WHERE Distance([[Urban 2].[ID]], [Towers 2].[ID], 'm') = [FirstQuery].[Dist from Urban]
```

Abram

**Find which Polyline Segment Lies Closest to a Given Point Location**

http://69.17.46.171/Site/Thread.aspx?id=21832&ti=632846720096630000

Adamw on 5/31/2006 8:17 AM (#21928)
First determine the segment closest to the desired location using a query, and then use your trigonometry skills to project the location to the segment.
To determine the closest segment, create a table named "Seq" with the only integer column named "Column" and the values 0, 1, 2, 3, ... N, where N is the maximum number of coordinates in any single line. After that, run the following query, substituting the coordinates of the desired location in place of "NewPoint(0, 0)" near the end of the query:

--SQL
SELECT [id], First([segment]) FROM (
    SELECT [id], [seq].[column], NewLine(
        Coord([geom (i)], [seq].[column]),
        Coord([geom (i)], [seq].[column] + 1)) [segment]
    FROM [drawing] INNER JOIN [seq]
    ON [seq].[column] < CoordCount(Branch([geom (i)], 0)) - 1
    ORDER BY [id], Distance(NewLine(
        Coord([geom (i)], [seq].[column]),
        Coord([geom (i)], [seq].[column] + 1)),
      AssignCoordSys(NewPoint(0, 0), CoordSys("Drawing" AS COMPONENT)))
) GROUP BY [id] ASC

Try running the query, then link it as a drawing.

The query does not handle lines with more than one branch (it ignores all branches but the first).

**Finding the Distance between every Line Node in a Drawing**

http://69.17.46.171/Site/Thread.aspx?id=18948&ti=632754365264370000

WillH at 2/13/2006 9:54 AM (#18965)
It isn't so much that you "loop" in SQL as you operate on "sets".

... the query below, compares every inflection in a line in "Drawing" with every other inflection in a line in the same drawing......

SELECT DistanceEarth(a.[g], b.[g])
FROM (SELECT [ID], [g] FROM [Drawing] SPLIT BY Coords([ID]) AS [g]) a,
     (SELECT [ID], [g] FROM [Drawing] SPLIT BY Coords([ID]) AS [g]) b
**Determine Distance from Point to Nearest Area of Certain Type**

http://69.17.46.171/Site/Thread.aspx?id=16445&ti=632677346230000000

**dsachs on 11/15/2005 4:30 PM (#16446)**

……I have a set of points representing sampling locations for forest plots… I also have a vector (drawing) map that has three classes of polygons….. What I need to do is to find the minimum straight line distance (in meters) from any forest cover plot point to the nearest edge of the nearest polygon with live trees (Class 1).

Don

**jkelly on 11/16/2005 1:08 AM (#16460)**

Don

I am not quite sure why you need to use contours or surfaces for minimum straight line distance between points and areas. The minimum distance is the horizontal distance, so there is no need for slope distances to be used.

To create a table with the minimum distances, the easiest way is to use SQL.

Import the table with the plot points and make a make a drawing from this. Make sure the coordinate system is set right. I suspect that you want to keep the distances in metres, so UTM will be suitable. Import the polygon file.

Use a SQL statement like:

```
Select [Drawing Pts].[id], MIN(Distance([Drawing].[id],[Drawing Pts].[id], "m")) as DIST
from [Drawing],[Drawing pts] where [Drawing].[Type] = 1
Group by [Drawing Pts].[id]
```

Where Drawing pts are the locations of the plots, Drawing is the regions, and type is the Class. This gives straight line distances in metres. This produces a table, you can then paste this and create a relation with the original point table. This table can be exported out.

**Find the Closest Road to Each Tower**

http://69.17.46.171/Site/Thread.aspx?id=16503&ti=632678240360000000

**KevinMunro on 11/17/2005 8:25 AM (#16504)**

Hello, I have a table of 20,000 towers and 16,000 major roads. I want to find the closest road to each tower.
Kevin Munro on 11/17/2005 11:33 AM (#16517)
I changed my approach and found that this query worked.

It is much faster and did the whole lot in minutes:

```sql
SELECT [Towers].[Name],
    (SELECT TOP 1 [Major Roads Table].[NatlCode1], Distance([Major Roads Table].[ID], [Towers].[ID])
    FROM [Major Roads Table]
    ORDER BY Distance([Major Roads Table].[ID], [Towers].[ID])) FROM [Towers];
```

Update Surface, Assign Each Pixel with the Distance to a Drawing

http://69.17.46.171/Site/Thread.aspx?id=25137&ti=632897519654100000

artlembo on 7/27/2006 2:20 PM (#25138)
Well, I'm most of the way there, but not entirely. I have two queries that I'm using to update a Surface to assign each pixel with the distance to a drawing.

The two queries are:

[Query 2]
```sql
SELECT AssignCoordSys(NewPoint([easting (i)], [northing (i)]),
    CoordSys("Universal Transverse Mercator - Zone 18 (N)")) as j FROM [S] as C
```

[Query]
```sql
'UPDATE S SET [height (i)] =
(SELECT DISTANCE(d.id,[query 2].j) FROM d,[query 2])
```

in the attached map, if you run [Query], you will get the distance from each pixel to the line.

But, if you remove the comment in the update portion, nothing happens to the Surface. The [Height (i)] field doesn't get updated.

Also, if there was another vector object, it would double the total output. So, there has to be some kind of TOP 1 parameter put in too.

Any ideas?

adamw on 7/28/2006 6:45 AM (#25162)
Try:
--SQL
UPDATE [S] SET [Height (I)] = (SELECT TOP 1
    Distance([D].[ID], NewPointLatLon([S].[Center Longitude (I)], [S].[Center Latitude (I)])) F
FROM [D] ORDER BY F)

adamw on 7/29/2006 6:39 AM (#25224)
Here is a faster version of the query, suggested by a colleague:

--SQL
UPDATE [S] SET [Height (I)] = (SELECT
    Min(Distance([D].[ID], NewPointLatLon([S].[Center Longitude (I)], [S].[Center Latitude (I)])))
FROM [D])

On my test data, the new query is 8x faster than the old one.

*Determine the Distance Down a Line to a Point (that may not lie directly on the line)*

http://69.17.46.171/Site/Thread.aspx?id=17440&ti=632720112243270000

njengler at 12/28/2005 1:40 PM (#17441)
…. How can I determine the distance along a line for a point that does not fall directly on the line? …. 

adamw at 12/28/2005 10:05 PM (#17456)
Assuming a drawing "P" with an arbitrary number of points and a drawing "L" with a single line, here is a query that would "project" all points onto the line:

--SQL

```
SELECT P.ID, Centroid ( ClipIntersect ( L.ID, Buffer ( P.ID,
    Distance (P.ID, L.ID) * 1.01 ) ) ) AS R FROM P, L
```

Link this query as a drawing.

dmbrubac at 12/29/2005 12:39 PM (#17467)
You can also draw lines between the points simply with
There is a small problem with this solution though;

Imagine a point on or near the split angle of an inflection point and on the obtuse side. There is no point on the line which is perpendicular to the point off the line, therefore the point on the line is either the vertex or very close to the vertex. How close is controlled by the 1.01 multiplication factor. I was experimenting with smaller values when I ran into a bug and lost the map (don't worry though it extremely unlikely you will run into this bug, and no, I can't tell you what it is).

There is another problem with points on the acute side of the inflection that is related to this, but is less severe.

Depending on what you are doing these are either big problems or no problem at all. In surveying this is unacceptable, unfortunately.

Oops - one more problem
When you do a buffer, you are actually generating a 'stroked' circle, therefore the distance from where the buffer crosses the line might not be exactly 1.01D, therefore the midpoint (Centroid) may not be exactly perpendicular to the point off the line.

Don't get me wrong - this is an elegant solution and lovely SQL - I just need more precision.

Also See: "Find Nearest & Farthest Object from a Group and Create Lines to Them"

Selecting Contained and/or Bounded Objects

Select Surface Pixels that Lie Within Areas

Mainfold-L
Author: Greg Reinecke (NA)
Date: 04-06-2005 12:44

Hello List,

My original question was:
> I have a surface (proda) and a drawing (poly). I am having trouble selecting a subset of the
> grid cells from (proda) that are within or intersect (poly).

Manifold-L
Author: Adam Wachowski (NA)
Date: 01-21-2005 22:17

> I have a surface (proda) and a drawing (poly). I am having trouble selecting a subset of the
> grid cells from (proda) that are within or intersect (poly).
>
> 1. Does anyone have the query syntax for doing this?

OPTIONS CoordSys("proda" AS COMPONENT);
SELECT * FROM proda WHERE EXISTS(
SELECT * FROM poly WHERE Contains(ID,
NewPointLatLon(proda.[Longitude (I)], proda.[Latitude (I)]))))

> 2. Once the cells are selected how can they get pasted into a new surface?

Import (or link) the above query as a surface: invoke File - Import (Link) - Surface, This Project,
set the "Table / query" combo to point to the query and click OK.

> 3. How will the grid cells that are intersected by the boundary of the (poly) feature be
> treated (ie. will only grid cells whose centroid is touching or within the polygon be
> selected or any part of the grid cell touching will be selected, etc)?

The query should select all pixels whose centers are in the drawing objects, however, right now it
will select all pixels whose left-bottom corners are in the drawing objects. This is a bug which
will be fixed in the nearest update.

If you want to select pixels whose centers are in the drawing objects, use:

OPTIONS CoordSys("proda" AS COMPONENT);
SELECT * FROM proda WHERE EXISTS(
SELECT * FROM poly WHERE Contains(ID,
AssignCoordSys(NewPoint(proda.[X (I)] + 0.5,
proda.[Y (I)] + 0.5), CoordSys("proda" AS COMPONENT))))

--
Adam Wachowski
Manifold Development Team
Select Rectangles that Contain Buffered Circles

http://69.17.46.171/Site/Thread.aspx?id=27320&ti=632934545241970000

KlausDE on 9/10/2006 3:08 AM (1 hour ago) (#27347)

--SQL
Select [Rectangle].[ID] FROM [Rectangle], [Circle]
WHERE Contains([Rectangle].[ID], ASSIGNCOORDSYS( InnerBuffer([Circle].[ID], 0.05), COORDSYS( "Circle" AS COMPONENT )))

Create a Separate Area Bounding Each Closed Line in a Drawing

http://69.17.46.171/Site/Thread.aspx?id=27560&ti=632938282644630000

go on 9/14/2006 10:33 AM (29 minutes ago) (#27573)

Bounded Areas transform creates areas that are completely bounded by one, or more than one, lines. What I want is to create, for each closed line in my drawing, the area bounded by this line. May be I have to write a script that launches two queries, one that selects all the closed lines in my drawing and the other that, for each element in the set returned, creates a bounded area…

And yes, I would like to use SQL because of speed.

willh on 9/14/2006 10:57 AM (5 minutes ago) (#27576)

Ok, so you have a situation where two or more lines form a "bounded area". Got it. Here is a SELECT that will create the areas, I'll leave it to you to sort out changing it to an INSERT and adding whatever other fields you need.

SELECT ConvertToArea([Geom (I)]) AS g FROM [drawing] WHERE IsRing([Geom (I)])

(LAK – The key here is the IsRing fuction which returns True if a given line is closed and contains one, and only one, branch with no self-intersections and False otherwise.)

go on 9/14/2006 11:27 AM (1 hour ago) (#27579)

Thanks Will.

Only some minutes before your answer I managed to create the following query:

SELECT ConvertToArea(GEOM([ID])) AS [Geom], [Column] FROM [D] WHERE IsClosed(GEOM([ID]))
I have a feeling that your query is better because of the IsRing function. However I don't fully understand that function.

**willh on 9/14/2006 12:49 PM (19 minutes ago) (#27584)**

From the manual:

IsRing: Returns True if given line is closed and contains one (and only one) branch with no self-intersections and False otherwise.

In other words, IsClosed will return true on a line shaped like an "8" but IsRing will not. Also, unless you're using a much older version of Manifold, Geom([ID]) is unnecessary; replace it with [Geom (I)]

**Selects All States Containing at Least One Point from the Drawing Cities**

http://69.17.46.171/Site/Thread.aspx?id=16897&ti=632695268990000000

**johnrobot on 12/5/2005 10:04 AM (#16898)**

The code

```sql
SELECT * FROM States, Cities
WHERE Contains (States.ID, Cities.ID);
```

selects all the states containing at least one point from the layer Cities. How would the code look if I was looking for states containing > 3 or perhaps between 5 and 150 points

Magnus

**jkelly on 12/5/2005 5:06 PM (#16916)**

Magnus

A fairly simple query just returning the state name and the number of cities (as long as the number is between 5 and 150) is as follows

```sql
Select * from
(SELECT count(Cities.ID) as CityCount,[States].[Name] FROM States, Cities
WHERE Contains (States.ID, Cities.ID)
GROUP BY [States].[Name])
where CityCount > 5 AND CityCount < 150
```
Create a Box around a Given Coordinate

http://69.17.46.171/Site/Thread.aspx?id=17220&ti=632727079991970000

adamw at 12/19/2005 10:43 PM (#17251)
The following query takes a component name and a pair of XY coordinates and returns a four-corner shape centered at the supplied location in the coordinate system of the supplied component. You can link this query as a drawing.

```sql
--SQL
PARAMETERS [Name] TEXT, [X] DOUBLE, [Y] DOUBLE;
VALUES (AssignCoordSys(CGeom(CGeomWKB("POLYGON (" &
   ([X]-10) & " " & [Y] & "," &
   [X] & " " & ([Y]-10) & "," &
   ([X]+10) & " " & [Y] & "," &
   [X] & " " & ([Y]+10) & "," &
   ([X]-10) & " " & [Y] & ")),
   CoordSys([Name] AS COMPONENT)));
```

nmack at 1/6/2006 1:32 PM (#17621)
I'm curious. Instead of using parameters in the example above, does the VALUES statement allow for copying of values from another table to substitute the [X] and [Y]?

nick

BTW - Adam thanks for your help. My IMS custom buffers work well.

adamw at 1/9/2006 11:23 PM (#17711)
No. To use the values from an existing table, use the SELECT statement.

Create Rectangles Enclosing Points with the Same Attribute

http://69.17.46.171/Site/Thread.aspx?id=16690&ti=63268834824000000

mdsumner on 11/24/2005 10:26 PM (#16712)
Oh, sorry I missed the bit about rectangles.

You could create the enclosing rectangles with a query (create a linked drawing to this query):

```sql
SELECT EnclosingRectangle(AllCoords(Geom([ID]))) FROM [D] GROUP BY [factor];
```

And then you could wrap that up with a SELECT that obtained the MinX/Y MaxX/Y and figure out which was the longest diagonal, even by using distanceEarth . . . That's a bit beyond me to actually do, I'm sure one of the SQL gurus can finish this off. ;)

28
**Analyse Land Use within a Buffered Area**

http://69.17.46.171/Site/Thread.aspx?id=8167&ti=632435137800000000

demeter on 2/1/2005 8:26 AM (#8168)

….. I have a drawing with different land use forms, like grassland, arable, marshes etc. (a Corine map). I would like to now what is the land use composition of a certain area defined by me, e. g. a circle, or in a buffer zone surrounding a point. Specifically, I want to analyse the land use composition surrounding white stork nests, in a 1 km circle.

Laszlo

KlausDE on 2/2/2005 8:58 AM (#8214)

Assuming you have created buffers around the stork nests in a Drawing [NestBuffers] and [NS1] is the type of land use in a Drawing [Corine] then you can use this nested query

```
SELECT [NS1], Sum([sqMeter]) AS SumA
FROM ( SELECT Area(ClipIntersect([Corine].[ID],[NestBuffers].[ID]),"sq m") AS sqMeter,
[Corine].[NS1] AS NS1
FROM [Corine], [NestBuffers] )
GROUP BY [NS1]
```

to sum up the area in reach of all stork nests grouped by the land use.

Klaus

demeter on 2/4/2005 6:04 AM (#8266)

Hi Klaus,
Thanks very much for the help, i hoped there is something more simple. In Arcview this analysis is almost automatic.

I made a first try and it didn't work. I got a two column table, the first is the land use types, the second one is empty (SumA). I try to figure out what's the problem.

Best,

Laszlo

KlausDE on 2/4/2005 11:47 AM (#8279)

May I suspect that you used the points of stork nests without buffering. If there is no area then there is nothing to ClipIntersect.

So use Transform Toolbar "All Objects" - "Buffer" - 1000 with [nest points] to create them and repeat the Query with [nest points] instead of [NestBuffers].
Don't be scared of the SQL code. It's not as difficult as it looks like. This one is a nested query and may not be a good starting point. But you can easy break it in a series of two queries and understand them just by reading:

Query1:

```sql
SELECT Area( ClipIntersect([Corine].[ID],[NestBuffers].[ID]),"sq m") AS sqMeter,
        [Corine].[NS1], [Corine].[NS2], [Corine].[NS3]
FROM [Corine], [NestBuffers];
```

Query2:

```sql
SELECT [NS1], [NS2], [NS3], Sum([sqMeter]) AS SumA
FROM [Query1]
GROUP BY [NS1], [NS2], [NS3];
```

In ArcView you really have to do the ClipIntersect and produce a new geometric data source. Additional in AV3 you then have to use a script hidden in samples or a 3rd-party extension to calculate the area. Than you call AV's query builder and produce the result of my first example.

Your used to it and so YOU do it almost automatic.

A) But what happens if there is another stork nest next year?
   1. Correcting the data in Manifold or in ArcView.
   2. Then simply run the query again in Manifold or repeat the hole process in ArcView.
B) And what happens if you want the analysis to show the detailed land use coded in [NS1], [NS2] and [NS3]? My second example solves this with only minor changes of the query. Now how do you solve this an AV? You can of course but far from automatically.
C) And what happens if you want the analysis to show the land use for the different pairs of storks or to change the unit of Area to foot or kilometer ...

SQL and especially Manifolds Spatial Extension is very powerful and worth learning TMHO.

good luck
Klaus

demeter on 2/5/2005 5:37 AM (#8291)
I had to reverse the order of clipintersect to [nestbuffers].[ID], [corine].[ID], instead of [corine].[id], nestbuffers].[id] to get results. Now it works fine. Maybe you understand why.

Thanks again,

Laszlo
Oops, there is a comparable swap of parameters in help concerning HeightSum(). In topic Spatial Extension you can read.

ClipIntersect(geom, clipWith)
Returns parts of given geometric object lying within clipping area.

ClipSubtract(geom, clipWith)
Returns parts of given geometric object lying outside clipping area.

I'll tested this again and report an error in help to tech@manifold.net. Glad you found a solution.

Klaus

To bring this to an end:

I thoroughly tested the effect of swapping parameters and found as expected: There is none.

I don't now what went wrong in Laszlo's query. For me it works perfect.

Klaus

Find the Total Line Lengths inside a Number of Individual Areas

http://69.17.46.171/Site/Thread.aspx?id=23541&ti=632900031658330000

I need to find the total length of lines in a number of individual areas at the same time. I know I've asked this before, and it's true, much the same operation can be done with spatial overlays and the "easier" transform operators, however, I think I can safely say that using SQL would be quicker, if only I could get the query template.

So, I have a Line drawing L and an Area drawing A, and I want to find out the sum total of the length of L within each individual area object. The lines are not currently clipped to the boundaries of the areas, so I also want to clip them within the query.

I have looked at the Spatial SQL video by artlembo, and have tried to use it as a template, but I am not getting anywhere. The code I have is as follows:

```
select sum (ClipIntersect([L].[Length (I)], (Select [A].[ID]
from [A]
where contains ([A].[ID], [L].[ID]))))
From [L]
```
I understand there are no books on Spatial SQL, but is there anything on normal SQL which might help me to get to grips with the language? And yes I do intend to purchase the Gisadvisor videos, but haven't got round to asking my boss yet!

Try:

```
--SQL
SELECT [A].[ID] AID, [L].[ID] LID, Length(ClipIntersect([L].[ID], [A].[ID]), "m")
FROM [L] INNER JOIN [A] ON Touches([A].[ID], [L].[ID])
```

That works fine, thanks. I have been trying to get it to group it by Area using:

```
SELECT [A].[ID] AID, [L].[ID] LID,
Sum (Length(ClipIntersect([L].[ID], [A].[ID]), "m"))
FROM [L] INNER JOIN [A] ON Touches([A].[ID], [L].[ID])
Group by [A].[ID]
```

but it's not working. Any ideas?

Many thanks for all your help

Since you'll be losing the individual line information by summing, try:

```
SELECT [A].[ID] AID,
Sum (Length(ClipIntersect([L].[ID], [A].[ID]), "m"))
FROM [L] INNER JOIN [A] ON Touches([A].[ID], [L].[ID])
Group by [A].[ID]
```

The GROUP BY clause requires that all the terms after SELECT are aggregate type terms (pardon the terminology, my SQL is also quite basic still). The field you’re grouping by doesn't need to be aggregated, but all the others do. You could've had:

```
SELECT [A].[ID] AID, First([L].[ID]) LID,
Sum (Length(ClipIntersect([L].[ID], [A].[ID]), "m"))
FROM [L] INNER JOIN [A] ON Touches([A].[ID], [L].[ID])
Group by [A].[ID]
```

but that LID wouldn't really mean anything.
Hope this makes sense. Also, I haven't tested the SQL, just assuming that that was the issue.

Henry

sknox on 8/1/2006 4:26 AM (#25321)
Yes that's great thanks - worked a treat. I understand why it wasn't working now.

**Creating Lines**

Convert Track Points to Lines – Successive Record Calculations

http://69.17.46.171/Site/Thread.aspx?id=13635&ti=632607432600000000
This Very Long thread is extremely interesting for all who have an interest in track lines (animal tracks in this thread but it could be storm tracks etc.). It is worth a read from top to bottom.
Note: a number of sections deleted from the original. – LAK

mdsumner on 8/11/2005 7:35 PM (#13636)
Hello - excuse my ignorance, but here are two questions that might have new answers - I am looking for pure SQL with the great new powerful Geom (I).

1. Can I join a set sequential points as a set of lines using SQL? (Think of the ScriptSequentialLines script, but with no scripting). I'd also like to have either one big long line object or many lines for each segment, which brings me to my next question, which is more database SQL than spatial:

2. Can I in SQL calculate values between successive records? I.e. can I find the difference (for e.g.) between record(i+1) and (record(i)

which would give me n-1 values (where n = # of records). I'd like to calculate things like the implied speed based on the distance and time between successive points for a track.

KlausDE on 8/12/2005 3:36 AM (#13641)
Still experimenting with your 1st question as it's really interesting having a mere SQL-solution for shared drawings.
Question 2. is easier:

```sql
SELECT [TrackPoint].[ID],
Distance([TrackPoint].[ID],[NextPoint].[ID]) AS Dist
FROM [TrackPoint]
Inner Join [TrackPoint] AS [NextPoint]
ON [TrackPoint].[ID] + 1 = [NextPoint].[ID];
```

You don't need `Geom()` and 6.5 to calculate differentiating values. The trick is aliasing a second instance of the table. Perhaps counting the intrinsic `ID`+1 is not generally applicable. You need some running index.

Klaus

(Notes: I have tested this. If you delete intermediate records, the ID field is not automatically renumbered so Klaus' caution above is valid. You can easily have IDs that increment by values more than 1 as noted by WillH below – LAK )

WillH on 8/13/2005 3:16 PM (#13696)
> KlausDE - 2005-08-12 3:36 AM

> SELECT [TrackPoint].[ID],
> Distance([TrackPoint].[ID],[NextPoint].[ID]) AS Dist
> FROM [TrackPoint]
> Inner Join [TrackPoint] AS [NextPoint]
> ON [TrackPoint].[ID] + 1 = [NextPoint].[ID];

I hate to be one of those guys that "clarifies" other people's solutions but... In Klaus' comment on the intrinsic ID not always being the best choice, he is getting into an area where the query above will break if TrackPoint.ID does not have a continuous set of numbers (e.g. 1,2,4,9)

WillH on 8/13/2005 3:50 PM (#13697)
  mdsummer - 2005-08-11 7:35 PM

Hello - excuse my ignorance, but here are two questions that might have new answers - I am looking for pure SQL with the great new powerful Geom (I).

You're right: the geom rocks! I'm always finding new & cool ways to use it.

1. Can I join a set sequential points as a set of lines using SQL? (Think of the ScriptSequentialLines script, but with no scripting). I'd also like to have either one big long line object or many lines for each segment, which brings me to my next question, which is more database SQL than spatial:
Try this to insert the line object directly into the source drawing (much like, and thereby replacing in some applications, the GUI based "import table as drawing with LineID")

Insert Into Drawing (LineID, [Geom (i)])
(Select LineID, ConvertToLine(AllCoords(ID)) From Drawing Group by LineID )

(This is going to give you a polyline containing the original point nodes containing the same LineID – i.e. One Line per group LAK)

You could very easily change this to a SELECT for a dynamic / static drawing separate from the points drawing.

2. Can I in SQL calculate values between successive records? I.e. can I find the difference (for e.g.) between

I was recently looking at something similar but using an active column and I think I may have found a bug either in the Manifold or the manual. Here is a bit from the Active Column topic:

Syntax and Conventions for Active Columns
When used within an Active Column script, 'Record' refers to the context record that has the following properties:

ID The ID of the drawing object. Only for drawing tables.
SequentialNumber Sequential number of the record within the table.

SequentialNumber, while potential very useful, doesn't seem to work. Has anyone else gotten it to work?

KlausDE on 8/13/2005 8:16 PM (#13698)
I mentioned "running index" to point to this.

Let’s discuss a solution that always will find the next record independent of type or continuity of data. I think I have a solution. Took me some time to work it out:

We have a [TrackPoint].[OrderField] of datatype DATE in the example and want the next [NextPoint].[OrderField] where [NextPoint].[OrderField] > [TrackPoint].[OrderField]. The Points have not been created in the correct order so we can’t use [ID]. There are gaps in the row of dates. See the labelled image at the bottom.

This aggregate query returns the correct series of FromDate - ToDate
SELECT [TrackPoint].[OrderField] AS [FromDate],
MIN([NextPoint].[OrderField]) AS [ToDate]
FROM [TrackPoint] AS [NextPoint] CROSS JOIN [TrackPoint]
WHERE [NextPoint].[OrderField] > [TrackPoint].[OrderField]
GROUP BY [TrackPoint].[OrderField]

The Result of this query is:

<table>
<thead>
<tr>
<th>FromDate</th>
<th>ToDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.08.2005</td>
<td>14.08.2005</td>
</tr>
<tr>
<td>14.08.2005</td>
<td>15.08.2005</td>
</tr>
<tr>
<td>15.08.2005</td>
<td>17.08.2005</td>
</tr>
<tr>
<td>17.08.2005</td>
<td>18.08.2005</td>
</tr>
<tr>
<td>18.08.2005</td>
<td>20.08.2005</td>
</tr>
<tr>
<td>20.08.2005</td>
<td>24.08.2005</td>
</tr>
<tr>
<td>24.08.2005</td>
<td>31.08.2005</td>
</tr>
</tbody>
</table>

We could have used every field type but binary to group the data.

Now we can join instances of the original table to the FromDate and to the ToDate and collect everything we need:

SELECT [FromDate], [ToDate], [FromPoint], Geom([Track2].[ID]) AS ToPoint,
        Distance( [FromPoint], Geom([Track2].[ID]) ) AS [Dist]
FROM [TrackPoint] AS [Track2]
INNER JOIN ( 
    SELECT [FromDate], [ToDate], Geom([Track1].[ID]) AS FromPoint
    FROM [TrackPoint] AS [Track1]
    INNER JOIN ( 
        SELECT [TrackPoint].[OrderField] AS [FromDate],
        MIN([NextPoint].[OrderField]) AS [ToDate]
        FROM [TrackPoint] AS [NextPoint] CROSS JOIN [TrackPoint]
        WHERE [NextPoint].[OrderField] > [TrackPoint].[OrderField]
        GROUP BY [TrackPoint].[OrderField] )
    ON [Track1].[OrderField] = [FromDate] )
ON [Track2].[OrderField] = [ToDate];

This is the resulting table with distances matching what we know to be true distances by the tracker tool.
Klaus

Oops, sorry for the German format of dates and floats

WillH on 8/13/2005 9:04 PM (#13699)
Interesting, Klaus. What about something like this:

```
Select d1.*,
    Distance(ID, (Select Top 1 d2.[Geom (I)] From Drawing d2 Where d2.ID < d1.ID
    Order by d2.ID Desc ) )
From Drawing d1
```

This give the distance to the previous point but can easily be switched (I actually wrote it as the
distance to the next point at first).

mdsumner on 8/13/2005 10:25 PM (#13704)
I've been stewing on some of this stuff for a while in different ways, and this is going to be really
important. Thanks so much for the kick-start!

It should be really easy to provide a bunch of simple tools for users of animal track data to
display, filter for quality, and summarize time spent and other behaviours - things that some Arc-
Etc extensions do. Much of this thread shows how to take a lot of the hard work out.

More soon. :-)

WillH on 8/13/2005 10:55 PM (#13708)

```
WillH - 2005-08-13 3:50 PM
```

```
insert into drawing (lineid,[geom (i)])
    (select lineid,converttoline(allcoords(id)) from drawing group by lineid)
```

This one, along with the possible bug above, is for Adam the Great; of course anyone that wants
to add to the concepts :-) What are your thoughts on the following: extending the Manifold DDL
to include something like

```
CREATE DRAWING [My Drawing] ([Name] TEXT, [Sample Date] DATETIME);
```
Fairly self explanatory, it would create a drawing component whose table takes was the defined columns. This would more useful when we could actually include multiple ";" separated in one query. Better yet, we could have SELECT INTO for drawings. That would be very cool and time saving!! Where you might have a SELECT query in which one of the columns is a geom column and you really want to make a static drawing from the result, you currently have to go through the GUI in one way or another. This could allow a one step query to create new drawings from queries.

KlausDE on 8/14/2005 6:43 AM (#13711)
Will,

Many reasons to prefer your solution. :-) 

I was stuck to the problem how to get results of a subquery accessible so that other results needed could be calculated in the same step.

But that was unnecessary. Your type of query simply is extendable by another line to calculate DateDiff() for instance.

Next reason to prefer your solution is that an aggregated join query as the heart of "my" solution will probably perform less efficient on large datasets. So my last thoughts deal with creating a record index on the fly counting trackpoints separate for every animal for example in SQL to overcome this. I'm sure I have done this in MS-Access before but I just can't remember how - next task for the SQL-veterans. You have my vote for including multiple ";" separated (queries) in one query and SELECT INTO for drawings as these have been part of my previous considerations.

Klaus

mdsumner on 8/14/2005 6:28 PM (#13716)
I'll just point out that the running index is the date time values - they should be ordered in time since that's what makes sense for a trajectory - plus with some handling of duplicates (both spatial and temporal), but this will need to be within an animal's ID - so that multiple "tracks" can be stored in one drawing. I'll post more on how I handle this elsewhere.

WillH on 8/14/2005 10:42 PM (#13719)

I'll just point out that the running index is the date time values - they should be ordered in time since that's what makes sense for a trajectory

Ok, how about this:
Select
Distance(ID, (Select Top 1 d2.Geom (I))
From Drawing d2
Where d2.[time] < d1.[time]
And d2.animal_ID = d1.animal_ID
Order by d2.[time] Desc)
),
d1.*
From Drawing d1
Order By [animal_ID],[time]

The path we headed down for non-continuous IDs is quite suited to switching to date-time values. I added a bit for handling multiple animal tracks in one table; how does that look?

- plus with some handling of duplicates (both spatial and temporal), but this will need to be within an animal's ID

Ok, I'm fairly certain I understand you in that you're saying that with multiple animals, there would possibly be records at the same time or place and so you have to be careful about the join? If so, the above should work as it is just grabbing the next record in time for the same animal.
WillH’s SQL construct above is an incredibly powerful tool. It can be used to create line segments from points. This would be a common requirement when building storm tracks from say, 6-hour point locations. We would generally want to preserve the tabular data for the last point in the storm history. This could be done by building a zero-length line as described further down. Using line segments, rather than a polyline track, allows you to thematically code the lines based upon, say the wind speed or the pressure at a given time.

We can do this with code via the object model BUT SQL is generally faster – far faster!

```sql
Select NewLine(ID,  
(Select Top 1 D2.[Geom (I)] From [Cyclone_364] D2  
Where  
  D2.[ID] > D1.[ID]  
Order by D2.[ID] )  
), [Name_E], [Dt], [Hour_UTC], [Lat], [Long], [YearlyStormNumber],  
[DatabaseStormNumber],[MaxWind_kts], [SS_E],  
[SystemSpeed_kts],[Pressure_mb], [Comment], [FiveYearPeriods],  
[TenYearPeriods], [FirstPosition], [LastPosition]  
From [Cyclone_364] D1  
Where D1.[ID] < (Select Max([ID]) from [Cyclone_364])
```

The query above will create 2-node line segments connecting the points. There will be one fewer line segments than points because of the

```sql
Where D1.[ID] < (Select Max([ID]) from [Cyclone_364])
```

If we do not use the WHERE clause, we end up with a final node with a NULL GEOM. If we want to preserve the tabular data for the last point, then copy that last point and past it back into the drawing (i.e. two last points). This will result in the last line being zero-length and all original tabular data is preserved.

The example above assumes that ID numbers always increment and that the native record order of the table is the logical, real world progressive track. Not sure if ID always increments. If we had a field that gave the Points "Order By" value then we could use it in a Where clause rather than the ID. This would allow us to reorder points from their native table positions as shown below.

```sql
Select NewLine(ID,  
(Select Top 1 D2.[Geom (I)] From [Points_OrderedByIndex] D2  
Where  
  D2.[ID] > D1.[ID]  
Order by D2.[ID] )  
), [Name], [ValidTime] [Dy], [Time], [Lat], [Long],  
[MaxWind_kts], [MaxWind_km], [SS], [Comment], [Stage],[Key]  
From [Points_OrderedByIndex] D1  
Where D1.[Key] < (Select Max([Key]) from [Points_OrderedByIndex] )  
Order by D1.[Key]
```
Finally, we could use an Insert Into clause to create new line segment objects in the same table, then delete the old, thereby converting the points drawing to a line drawing. (As of V7, Manifold® does not allow direct change of a Geom from one object type to another.)

The Query below will result in a drawing containing both points and line segments.

```
Insert Into [Test_Points2Lines]
( [Geom (I)], [Name], [ValidTime], [Dy], [Time], [Lat], [Long],
  [MaxWind_kts], [MaxWind_kmh], [SS], [Comment], [Stage], [Label], [Key] )
Select NewLine(ID,
  (Select Top 1 D2.[Geom (I)] From [Test_Points2Lines] D2
   Where D2.[Key] > D1.[Key]
   Order by D2.[Key] )
), [Name], [ValidTime], [Dy], [Time], [Lat], [Long],
  [MaxWind_kts], [MaxWind_kmh], [SS], [Comment], [Stage], [Label], [Key]
From [Test_Points2Lines] D1
Where D1.[Key] < (Select Max([Key]) from [Test_Points2Lines] )
Order by D1.[Key]
```

So far we are dealing with only one track. What about the situation where multiple tracks are involved that could be sequentially ordered by some Track Identifier column and a Track order or sorting column? This can be done but things get clumsy and slow for big tables.

```
INSERT INTO [HURDAT_Points]
( [Geom (I)], [Name_E], [Name_F], [Dt], [Dt_E], [Dt_F], [Yr], [Mon], [Dy], [Julian_Dy],
  [Hour_UTC], [Long], [Lat], [YearlyStormNumber], [DatabaseStormNumber],
  [MaxWind_kts], [SS_E], [SS_F], [SystemSpeed_kts], [Pressure_mb], [Comment],
  [FiveYearPeriods], [TenYearPeriods], [LineStyle], [FirstPosition],
  [LastPosition], [RecordNumber] )
SELECT NewLine
( ID,
  ( SELECT TOP 1 D2.[Geom (I)] From [HURDAT_Points] AS D2
    WHERE D2.[DatabaseStormNumber] = D1.[DatabaseStormNumber]
    AND D2.[RecordNumber] > D1.[RecordNumber]
    ORDER BY D2.[RecordNumber] )
), [Name_E], [Name_F], [Dt], [Dt_E], [Dt_F], [Yr], [Mon], [Dy], [Julian_Dy],
  [Hour_UTC], [Long], [Lat], [YearlyStormNumber], [DatabaseStormNumber],
  [MaxWind_kts], [SS_E], [SS_F], [SystemSpeed_kts], [Pressure_mb], [Comment],
  [FiveYearPeriods], [TenYearPeriods], [LineStyle], [FirstPosition],
  [LastPosition], [RecordNumber]
FROM [HURDAT_Points] AS D1
Where D1.[RecordNumber] < (Select Max(D3.[RecordNumber]) From [HURDAT_Points] as D3 Where D1.[DatabaseStormNumber] = D3.[DatabaseStormNumber])
ORDER BY D1.[DatabaseStormNumber], D1.[RecordNumber]
```
PROBLEMS TO WORK ON:

1. There is a problem with the last query that seems to be tied to that last "Where" clause:
   
   ```sql
   Where D1.[RecordNumber] <
   (Select Max(D3.[RecordNumber]) From [HURDAT_Points] as D3 Where
   D1.[DatabaseStormNumber] = D3.[DatabaseStormNumber])
   
   This additional restriction brings the query to a standstill on large, multi-track databases.
   
   2. All the Queries above require that last Where clause to prevent a blank Geom being created
   at the end of a single track or at the end of each track group in a multi-track file. What is really
   needed is a solution that eliminates that requirement but still gives the right result.
   ```

   SQL above is implemented in project "Testing_LineSegments.map" in Folder
   C:\Documents and Settings\Lorne_Limited\My Documents\Manifold\VBscript

---

Create a Line Geom between Two Points on Each Row

http://69.17.46.171/Site/Thread.aspx?id=17241&ti=63270778445000000

bjs at 12/21/2005 3:45 PM (#17309)

```sql
SELECT
AssignCoordSys(CGeom(CGeomWKB(  "L"INESTRING  (" &
  CStr(Table.X1) &  " " &
  CStr(Table.Y1) &  ", " &
  CStr(Table.X2) &  " " &
  CStr(Table.Y2) &  ")" )
 )), CoordSys("Drawing" AS COMPONENT)) AS aline FROM Table
```

The problem was that the AssignCoordSys was trying to get the coordinate system from a table
with no coord system, so change it to another drawing in your project (its the "Drawing" in the
above statement) which has the coord sys that the points in your table are on.

The extra columns p and Q in my first statement were other columns in your table that you might
have wanted to include with the newly created lines, for formatting as you said earlier.

Working from the inside out, you create a text string in WKT format, convert it to a WKB
format, convert that to a manifold GEOM format, then assign a coordinate system to it, so we
know where the lines lay on the planet.

Then yes, link the query as a drawing and presto.

Hope this helps.

Bernie
**Creating Lines from Points (using WKT)**

http://69.17.46.171/Site/Thread.aspx?id=17639&ti=632724443090770000

Johnny at 1/7/2006 11:02 AM (#17640)
Is there a way using SQL to create lines? I have a table with points and managed to write a query that comes up with two columns both containing point Geoms.

adamw at 1/9/2006 10:58 PM (#17706)
Try going through WKT:

```sql
--SQL
SELECT CGeom(CGeomWKB("LINESTRING (" &
   CentroidX([Geom1]) & " " & CentroidY([Geom1]) & "," &
   CentroidX([Geom2]) & " " & CentroidY([Geom2]) & ")") FROM [Tab];
```

(It appears that the two Geoms are located in the same record. LAK)

**How to Use the "ConvertToLine" SQL Function**

Manifold®-L
Author: AdamW

To add to others answering your question, here is how you would use ConvertToLine (the point drawing is named "P", the order of coordinates is given by the column named "X", link this query as a drawing):

```sql
SELECT ConvertToLine(AllCoords(G)) FROM
   (SELECT [Geom (I)] G FROM [P] ORDER BY [X])
```

Adam Wachowski
If multiple lines are involved, then use a Group By clause such as:

```
INSERT INTO [Drawing] ([Geom (I)])
Select ConvertToLine(AllCoords([Geom (I)]))
FROM [Drawing] GROUP BY [LineID];
```

**NOTE:** This is the method that Manifold uses when you choose the line option when importing a points table – see figure to the right.

If the point sequence is not in a logical order representing a real world track, then use Order By to pre-process the point order first as shown below.

```
INSERT INTO [Drawing] ([Geom (I)])
Select ConvertToLine(AllCoords([Geom (I)])) as pLines
FROM (Select * FROM [Drawing] Order by [LineID],[Point_Sequence])
GROUP BY [LineID];
```

**NOTE:** This method only creates a multi-node polyline. That approach is not too useful in track displays. Often one would like to thematically format the track sub-segments based upon some physical parameter. Example: a storm maximum wind. In that case, you really need a series of 2-node lines connecting the dots, which is much harder to create.

---

**Find Nearest & Farthest Object from a Group and Create Lines to Them**

Bcowper on 9/13/2006 4:38 AM (#27495)

I would like to see a Nearest and Furthest to Object function added to Manifold as a Transform tool and a SQL statement. This would work by finding the nearest/furthest distance to an object from a group of objects and create lines representing the nearest/furthest distance. An example would be a single line (2 nodes) as the target object and a bunch of points placed outside the line, when you run the Nearest command perpendicular lines are drawn from the points to the target line.

This is a feature that was added to MapInfo 8 (not enticing enough for me to upgrade!) that would be very handy in Manifold; perhaps there is a way of achieving this already that I've missed; if not, I will submit a feature request to Manifold.

artlembo on 9/13/2006 6:32 AM (#27499)

Here you go:
Assume you have 2 drawings: D1, D2. D1 has a whole bunch of points. D2 as one point. What this query does is create a line object between the D2 point and the closest D1 point. Of course you could change the sort order and compute the furthest.

```sql
INSERT INTO D2 ([Geom (I)]) VALUES (SELECT NEWLINE(did,d2id) FROM
(SELECT TOP 1 DISTANCE(d2.id,d1.id) AS dist, d1.id as did,d2.id as d2id
FROM d2, d1
ORDER BY dist ASC))
```
of if you want to do it with multiple points, you can make some modifications.

So, this is fairly easy to do. But, what would be nice is to not have so many Query components lying around. It would be great to easily add buttons on top of the GUI and allow you to store an SQL procedure for the button. Sort of the way ArcView 3.2 did it.

So, imagine you do something like grab a toolbar, hold the CTRL key, and double click. What would then pop up is a dialog allowing you to add a new button, and in the button window is a little text box allowing you to put in some SQL code. Then, when that is done, you just can start to use the button. With this kind of ease of use, people can create hundreds of buttons for different automated tasks, and not have to deal with all those components lying around.

KlausDE on 9/13/2006 6:42 AM (#27500)
Nice idea. To generalize it the reuse of these SQL-behind-Button function with different datasources would need a new feature of component names as values for Parameters.

KlausDE on 9/13/2006 7:00 AM (2 minutes ago) (#27501)
As supplement to Art's solution here is a way to select the reference point(s) from the same Drawing as the other points.

```sql
--SQL
' Line from selected to the neares Object
SELECT AddCoord([Pt1],[OId]), Round([Dist],2) AS [Dist] FROM
(SELECT Top 1 [Objs].[ID] AS OId,
Distance([Sel].[ID], [Objs].[ID]) AS Dist,
    ConvertToLine([Sel].[ID]) AS Pt1
FROM [Drawing] AS [Sel],
WHERE [Sel].[Selection (I)]
ORDER BY Dist ASC)
```
I've added a map where you select a point and read the measured distance when you Refresh Data for the linked component [Draw Neares].
Attachments:
DrawNearest.map
Thanks Art and Klaus, I can certainly use these for points, but the main thing I need is for finding the nearest distance from a point to a line or area object, if I'm reading your solutions correctly they're for points to point queries.

I came up with a fudge solution by segmentizing the line at small intervals and creating points from the segments, then using Nearest Neighbor to create the 'perpendicular' lines, which doesn't always produce the best results, but is better than nothing.

I have no solution, if the line must draw the shortest connection. The modified query uses the Centroid of a point/line/area object to draw the line. But Distance is measured to the next point on the Border or on the line. It's only the space between objects, that counts. Not the position of the Centroid.

--SQL

' Line from selected to the nearest Object
SELECT AddCoord([Pt1],Centroid([OId])), Round([Dist],2) AS [Dist] FROM
(SELECT Top 1 [Objs].[ID] AS OId,
Distance([Sel].[ID], [Objs].[ID]) AS Dist,
    ConvertToLine(Centroid([Sel].[ID])) AS Pt1
FROM [Drawing] AS [Sel],
WHERE [Sel].[Selection (I)]
ORDER BY Dist ASC)

Create a Series of Lines of Best Fit through Points having the Same Attribute

http://69.17.46.171/Site/Thread.aspx?id=16690&ti=632688348240000000

I have a network of points with an attribute. What I would like to do is create a series of lines of best fit through the points with the same attribute. So if I had a thousand points with three different values for the attribute, I'd get three lines that weave through the network.

The closest I've come so far to the answer is by using the transform "Enclosing Rectangles," but what I would prefer is a single line through the middle of the longest dimension of these rectangles. Even better would be some sort of best fit path through the points with the same attribute.

Bernie
mdsumner on 11/24/2005 4:52 PM (#16701)
What do you mean by "best fit"? I assume lines aren't intended to be monotonic (i.e. it's not x/y regression). Do you mean "shortest path" perhaps? I also assume you want non-branching lines?

bjs on 11/24/2005 9:40 PM (#16710)
Thanks mdsumner,

Now we're getting technical. ;-)

Linear regression would be fine, but I suppose the optimal result I'm after is a bit like shortest path, but it need not pass through the points in question.

Yes non branching lines, I tried triangulation and dissolving but I'd like a single line.

mdsumner on 11/24/2005 10:26 PM (#16712)
Oh, sorry I missed the bit about rectangles.

You could create the enclosing rectangles with a query (create a linked drawing to this query):

SELECT EnclosingRectangle(AllCoords(Geom([ID]))) FROM [D] GROUP BY [factor];

And then you could wrap that up with a SELECT that obtained the MinX/Y MaxX/Y and figure out which was the longest diagonal, even by using distanceEarth... That's a bit beyond me to actually do, I'm sure one of the SQL gurus can finish this off. ;)

bjs on 11/28/2005 6:29 AM (#16745)
I Cheated and just used the transform toolbar to create the rectangles. I'm almost there I think, I've figured out how to get the two endpoints of the longitudinal axis of the enclosing rectangle:

SELECT
AssignCoordSys(NewPoint(
    (Centroidx(Coord([Geom (I)],1))+Centroidx(Coord([Geom (I)],4)))/2,
    (Centroidy(Coord([Geom (I)],1))+Centroidy(Coord([Geom (I)],4)))/2
), CoordSys("bdyRecLine" AS COMPONENT)) AS Pnt1,
AssignCoordSys(NewPoint(
    (Centroidx(Coord([Geom (I)],2))+Centroidx(Coord([Geom (I)],3)))/2,
    (Centroidy(Coord([Geom (I)],2))+Centroidy(Coord([Geom (I)],3)))/2
), CoordSys("bdyRecLine" AS COMPONENT)) As Pnt2,
RoadName
FROM bdyRecLine

But now I'm stuck on how to make a line between these two endpoints. Oh, and I've just realised I don't know whether I've got the long or short axis. Oh well, I'll post this anyway so I can follow my steps when this is finally cracked.
I had some success so I thought I'd let you know what I found. Perhaps there's other beginners like me how need some examples. The answer was there in the documentation and on georeference.org all along, but it took a bit of finding.

This SQL creates a line through the longest axis of a set of rectangles created by the "enclosing Rectangle" transform.

It starts with an IIF statement to find the longest axis of the rectangle using the DISTANCE function. Then it creates a text string in WKT format which is interpreted by the CGEOMWKB function and converted to a Manifold Geom by the CGEOM function, and assigned the coordinate system/projection of the drawing it's derived from using the ASSIGNCOORDSYS function. The COORD function is used throughout to obtain a specific vertex from the rectangle. The CENTROIDX and CENTROIDY functions are used to extract the X and Y from the vertices.

If I hope I have described the process correctly.
SELECT
  IIF (distance(Coord([Geom (I)],1),Coord([Geom (I)],4))<distance(Coord([Geom (I)],1),Coord([Geom (I)],2)),
    --true
    (AssignCoordSys(CGeom(CGeomWKB( "LINestring (" &
      CStr((Centroidx(Coord([Geom (I)],1))+Centroidx(Coord([Geom (I)],4)))/2) & "
      &
      CStr((Centroidy(Coord([Geom (I)],1))+Centroidy(Coord([Geom (I)],4)))/2) & "
      ", " &
      CStr((Centroidx(Coord([Geom (I)],2))+Centroidx(Coord([Geom (I)],3)))/2) & "
      &
      CStr((Centroidy(Coord([Geom (I)],2))+Centroidy(Coord([Geom (I)],3)))/2) & "
    )
  , CoordSys("bdyRecLine" AS COMPONENT)) ),
  --false
  (AssignCoordSys(CGeom(CGeomWKB( "LINestring (" &
    CStr((Centroidx(Coord([Geom (I)],1))+Centroidx(Coord([Geom (I)],2)))/2) & "
    &
    CStr((Centroidy(Coord([Geom (I)],1))+Centroidy(Coord([Geom (I)],2)))/2) & "
    ", " &
    CStr((Centroidx(Coord([Geom (I)],3))+Centroidx(Coord([Geom (I)],4)))/2) & "
    &
    CStr((Centroidy(Coord([Geom (I)],3))+Centroidy(Coord([Geom (I)],4)))/2) & "
  )
    , CoordSys("bdyRecLine" AS COMPONENT)) )
)--End if
AS theLine
, RoadName
FROM bdyRecLine
Creating Grids

**Build Grid using KRIGING and [X (I)], [Y (I)]**

http://69.17.46.171/Site/Thread.aspx?id=19866&ti=632792261502970000

mdsumner at 3/27/2006 2:34 PM (#19911)

……………… There are Kriging commands in the SQL Raster Extensions, but you would have to build the grid first (perhaps with a script), then use those regular points in the query i.e.

SELECT [X (I)] AS [X], [Y (I)] AS [Y], Kriging([Drawing], [Column], -1, [Geom (I)]) AS [Z] FROM [Grid]

**Building a Grid using any Min X, Max X, Min Y, Max Y and Grid Size**

Non-SQL

Dimitri on 9/26/2006 7:07 AM (4 days ago) (#28148)

The easiest way to build a grid is to use the Grid command:

1. Create a drawing that covers the desired geographic extents.
2. Choose View - Grid
3. Choose "Points" for the Style and set the From and To and Spacing values.
4. Press Create.
5. Uncheck the "Show grid" box and press OK.

You now have a grid of point objects.

seatrails on 9/26/2006 8:54 AM (4 days ago) (#28156)

http://69.17.46.171/Site/Thread.aspx?id=28146&ti=632948576637400000

You can create points with SQL in many ways. You can insert into a drawing or you can link a drawing to a query. The challenge is enumerating the middle points.

Create a table (called mids) with filled with an arithmetic series 0...1000 or so

- Parameters: minX, minY, maxX, maxY, numXgrid, numYgrid
- Create a subquery of the form:
  (Select minX + index * (maxX-minX) / (numXGrid-1) xVal from mids where index < numXGrid) as xVals
- same for the y's
- select form a cross join of those two queries with a newpoint() and appropriate assigncoordsys statements
As a variant, you could forego the helper table with a values clause with an arithmetic series in it.

L. Ketch

Following is SeaTrail's flowchart above as final SQL code. It seems to work well and is very fast on large grids.

```
OPTIONS CoordSys("Latitude / Longitude");
Select X.XVal as X, Y.YVal as Y, AssignCoordSys(NewPoint(X.XVal, Y.YVal), Coordsys("Latitude / Longitude")) as Point_Geom From

(Select 0 + [Index] * (-180-(0)) / (181-1) as XVal
 From [Mids] WHERE [Index] < 181) As X
Cross Join
(Select 0 + [Index] * (90-(0)) / (91-1) as YVal
 From [Mids] WHERE [Index] < 91) as Y

[Index] is the column in [Mids] that contains the sequential numbers.
```

See Folder:
C:\Documents and Settings\Lorne_Limited\My documents\Manifold\VBscript

Project:
Create_Grid_viaSQL

Joining Two Tables

*Find Min, Max, Avg and Point Location*

http://69.17.46.171/Site/Thread.aspx?id=17169&ti=632706305101400000
dprellwitz on 12/17/2005 3:07 PM (#17170)

I’ve got a statistics file coming in every night that contains:
Router-Name (a primary location), Port+Remote-Name (the other end of this port), date/time of sample, Rx%, Tx%, Error%, TxReject%, bandwidth.

I have a table with geocodings for each of the router-names and Remote Names with Lat/Lon/Elevation data.
I’m trying to get a summary file with daily min, max and averages for Rx and Tx to include the point data, then draw a line between the points with line size & color to depend upon the error % and Rx/Tx thresholds (80/90/95%).

As a first step, I’ve got the daily stat’s file math functions working okay via the first query. The second query works to show the addition of the correct lat/lon for site-a. I’m having trouble with the third query.

Also, I’d like to get this part of the process in a single query.

Here’s the process:

My first query averages the daily file to produce an interim file with:
Site-A Name (“Router Name”), Site-B Name (“Remote Name”), Rx-Min, Rx-Max, Rx-Avg, Tx-Min, Tx-Max, Tx-Avg.

SELECT [D11Service].[Router_Name] as from_site, [D11Service].[PortSite_B] as to_site, 100*min(Rx_Percent) as Rx_Min, 100*max(Rx_Percent) as Rx_Max, round(100*avg(Rx_Percent)) as Rx_Avg, 100*min(Tx_percent) as Tx_Min, 100*max(Tx_percent) as Tx_Max, round(100*avg(Tx_Percent))as Tx_Avg into temp1 from [SiteLocation],[D11Service] group by D11Service.[Router_Name], D11Service.[PortSite_B];

My second query adds the lat/lon/elevation for Site-A Name to the interim file and displays the results.

SELECT temp1.*, sitelocation.[Latitude (I)] AS sitea_lat, sitelocation.[Longitude (I)] AS sitea_lon from temp1, sitelocation where Temp1.[from_site] = sitelocation.[PortLocation]

I’m trying to get a third query to add the remote-site lat/lon/elevation data to the file. (I’ll refrain from embarrassing anyone (mainly me) with my bad coding samples…!)

I’ve been trying to get this to work for three days,,, help please!
/David

adamw on 12/19/2005 6:40 AM (#17213)
Unless I am missing something, the third query would be analogous to the second:

SELECT temp2.*, sitelocation.[Latitude (I)] AS siteb_lat, sitelocation.[Longitude (I)] AS siteb_lon FROM temp2, sitelocation WHERE temp2.[to_site] = sitelocation.[portlocation];

All three queries joined together:
SELECT Router_Name AS from_site, A.[Latitude (I)] AS from_lat, A.[Longitude (I)] AS from_lon,
        PortSite_B AS to_site, B.[Latitude (I)] AS to_lat, B.[Longitude (I)] AS to_lon,
        100*Min(Rx_Percent) AS Rx_Min, 100*Max(Rx_Percent) AS Rx_Max,
        Round(100*Avg(Rx_Percent)) AS Rx_Avg,
        100*Min(Tx_percent) AS Tx_Min, 100*Max(Tx_percent) AS Tx_Max,
        Round(100*Avg(Tx_Percent)) AS Tx_Avg
FROM ((D11Service D
        INNER JOIN SiteLocation A ON D.Router_Name = A.PortLocation)
        INNER JOIN SiteLocation B ON D.PortSite_B = B.PortLocation)
GROUP BY Router_Name, PortSite_B

dprellwitz on 12/19/2005 8:57 AM (#17219)

Looks great!... having problems with the alias's "A" and "B", i.e., I'm not sure where (or how) they're being assigned... I'm assuming that 'A.[Latitude (I)]' means "D11Service.Latitude(I)" and same with A.[Lon...], B.[Lat...] and B.[Lon...]?
Is it at the INNER JOIN line where you have "....SiteLocation A on D.Router...."?
And "((D11Service D "

Thanks!
David

adamw on 12/19/2005 11:08 PM (#17255)

D11Service gets aliased as "D" ("D11Service D" is equivalent to "D11Service AS D"). The instance of SiteLocation used to deliver the lat/lon coordinates for Router_Name gets aliased as "A". The instance of SiteLocation used to deliver the lat/lon coordinates for PortSite_B gets aliased as "B".

Example 2:

From: Robert Sanson
Sent: September 27, 2006 10:42 PM
To: manifold-l@lists.directionsmag.com
Subject: [Manifold-l] UPDATE statement involving two tables
http://lists.directionsmag.com/discussion/read.php?f=29&i=42625&loc=0&t=42625

Can anyone help me? - I want to update a column in one table based on the values in a column in a second table. Both tables have a common field. I have tried the following and it doesn't work:

UPDATE [Table1]
INNER JOIN [Table2]
ON [Table1].[FARM_ID] = [Table2].[FARM_ID]
SET [Table1].[dist_km] = [Table2].[dist_km]
Many thanks,

Robert Sanson

From: vincent lacour
Sent: September 28, 2006 3:07 AM
To: 'Robert Sanson'; manifold-l@lists.directionsmag.com
Subject: RE: [Manifold-l] UPDATE statement involving two tables

You can try:

```
UPDATE (SELECT [Table 1].[Dist_km], [Table 2].[Dist_km] AS [d2]
FROM [Table 1] INNER JOIN [Table 2]
ON [Table 1].[Farm_ID] = [Table 2].[Farm_ID])
SET [Table 1].[Dist_km] = [d2];
```

Vincent

From: adamw@manifold.net
Sent: September 29, 2006 10:22 AM
To: manifold-l@lists.directionsmag.com
Subject: RE: [Manifold-l] UPDATE statement involving two tables

> Looks like a bug. The SQL below works in Access and it pretty much
> matches yours:
> >
> > UPDATE T1 INNER JOIN T2 ON T1.Farm_ID = T2.Farm_ID
> > SET [T1].[X]= [T2].[X]

This is not a bug but rather a syntactical difference.

There are plenty of flavors of SQL out there. The one used in Access (Jet SQL) is not exactly the
most logical one, and so while we strive to be as compatible with it as possible there are several
things that we do differently. Thus, we require that the join above is enclosed into a SELECT,
similarly to Vincent's variant.

Adam Wachowski
Pivot Tables

**Pivot Table Example 1 – Point Distance Matrix**

http://69.17.46.171/Site/Thread.aspx?id=13376&ti=632588268600000000

Johnny on 8/5/2005 6:42 AM (#13377)

I was fiddling around with a point layer and wanted to get a pivot table to show some kind of distance matrix.

Johnny on 8/5/2005 8:21 AM (#13386)

Thanks both Klaus and Art….
However I slightly changed the SQL and added the "sum()" aggregator - which doesn't harm the result but helps the TRANSFORM to run at all. So My code looks now like this:

```
Transform sum(distance(pt1.ID, pt2.ID)) Select pt1.ID
From points as pt1, points as pt2
WHERE pt1.ID <> pt2.ID
Group by pt1.ID
Pivot pt2.ID
```

**Pivot Table Example 2 – Sum of the Area Comprised by each Class within a Buffer around each Point**

http://69.17.46.171/Site/Thread.aspx?id=24759&ti=632894342921630000

chrismarx on 7/22/2006 6:52 PM (#24760)

Here is the scenario-
I have a drawing with vegetation type polygons and another drawing with points. I'd like to find the sum of the total area comprised by each vegetation class within a 100m buffer around each point. Ideally, I would like the pivot table to return the point name, along with 18 columns (number of different vegetation classes) each of which would have the sum of that vegetations area within the buffer.

adamw on 7/24/2006 8:59 AM (#24811)

Try:
```
--SQL
TRANSFORM Sum(V.[Area (I)])
SELECT P.[Name]
FROM [Points] P INNER JOIN [Vegetation Drawing] V
    ON Contains(Buffer(P.[Geom (I)], 100, "m"), V.[Geom (I)])
GROUP BY P.[Name]
PIVOT V.[class]
```

**Alias Variables**

**Alias Variables and SQL Processing Sequence**

http://69.17.46.171/Site/Thread.aspx?id=26220&ti=632926356630100000

Lorne at 8/21/2006 3:53 AM (#26221)

On August 18th, Adam suggested that we begin a thread devoted to SQL. I have attached a Word file containing some examples. Perhaps it could serve as a starting point.

Based upon the SQL questions submitted on GeoReference and Manifold-L, I think there are many Manifold users who have difficulty with SQL. Personally, I find the simpler examples intuitive (to use Art's term). However, as the complexity ramps up and those elegant 8 line solutions from Adam roll in, I quickly lose my way. There are some general concepts for which I would love to see short, clear explanations. So, in addition to SQL examples, I would also like to see highly condensed SQL theory plus a few other items.

Following are some things that would really help me:

1. A SQL, URL list containing non-Manifold sites with high quality, complex SQL examples and explanations. I have not had much success finding such resources. Introductory topics are covered well in the Manifold help files. However, people may a few books covering advanced concepts that they would recommend.

2. I have not been able to get a good handle on how Manifold handles aliased variables – where they can and cannot be utilized. As my query grows longer, I begin to run into those messages indicating that Manifold cannot recognize an aliased field. There is some scattered help information on this but a one to two paragraph summary with complex examples would be great.

3. An explanation of how SQL engines process joins on multiple tables in deeply nested command strings. I have done amateur programming for a long time. Ideas like branches and loops are second nature; however, SQL does not operate along these lines. There was a thread from a few months back where someone asked how one can build a loop in SQL (I think some SQL engines actually have a construct that allows this). One of the responses indicated that SQL is best viewed as working on Sets. One of the attached Word examples contains an Adam solution to a question that I asked (Duplicating the GUI Explode Transform in SQL). Adam's SQL is extremely interesting and I can understand it after the fact – I could never have created it.
from scratch because I don't have a good grasp of the processing sequence for these multiple table queries.

Lorne

adamw at 8/23/2006 6:53 AM (#26362)

On 2:

Aliases are used for two main purposes: to resolve ambiguities and to simplify the text of the query and make it more manageable. Each alias has a scope which is determined by the place where it is defined. The keyword "AS" used to define an alias can be omitted.

Using aliases to resolve an ambiguity:

```
--SQL
SELECT A.[ID], B.[ID] FROM [Data] A, [Data] B
WHERE IsPoint(A.[ID]) AND IsArea(B.[ID]) AND Contains(B.[ID], A.[ID])
```

Using aliases to simplify the text of the query:

```
--SQL
SELECT A.[ID], Min(D) FROM
    (SELECT A.[ID], Distance(A.[ID], P.[ID]) D FROM [Areas] A, [Points] P)
GROUP BY A.[ID]
```
The "scope" of the alias that Adam mentions above is the key. An outer portion of a command cannot "look into" an inner selection enclosed in parentheses without some help. If you need to pass inner results to outer portions of the query, then you must name the inner virtual table and inner virtual columns as shown in the example below:

```sql
OPTIONS CoordSys("Latitude / Longitude");
Select X.XVal as X, Y.YVal as Y, AssignCoordSys(NewPoint(X.XVal, Y.YVal), CoordSys("Latitude / Longitude")) as Point_Geom From
  (Select 0 + [Index] * (-180-(0)) / (181-1) as XVal
   From [Mids] WHERE [Index] < 181) As X
  Cross Join
  (Select 0 + [Index] * (90-(0)) / (91-1) as YVal
   From [Mids] WHERE [Index] < 91) as Y

The example above is based upon rough outline suggested by SeaTrails
http://69.17.46.171/Site/Thread.aspx?id=28146&ti=632948576637400000

In the query sub-section shown below, the entire area between the parentheses will generate a virtual table which can be assigned a table alias name in the same manner as a simpler example such as:
Select D.[a] from [Table] as D  (D is the alias for [Table])

(Select 0 + [Index] * (-180-(0)) / (181-1) as XVal
 From [Mids] WHERE [Index] < 181) as X

(X is the alias for the virtual table defined by
(Select 0 + [Index] * (-180-(0)) / (181-1) as XVal From [Mids] WHERE [Index] < 181)

This makes the virtual table name (X) and the inner calculated value (XVal)
accessible to the portion of the query outside of those parentheses via X.XVal.

On the other hand, it appears that an inner selection enclosed in parentheses can see an outer table alias since the example below does work.

UPDATE [Drawing] as D1 Set D1.[Geom (I)] =
  ( SELECT Segments( D2.[Geom (I)], int( D2.[Length (I)]/50 ) )
   FROM [Drawing] as D2
   WHERE D1.[ID] = D2.[ID]
  )
On 3:
The exact algorithm used to process a join depends on what kind of join it is, how many records are within each joined record set and on other factors. Fastest joins are those on primary keys (ID) and spatial joins with "Distance(p, q) < x"-like criteria.

**Alias Theory and Examples**

1. willh on 6/27/2006 6:22 PM (#23429)

   http://69.17.46.171/Site/Thread.aspx?id=23351&ti=632870293446070000

   ```sql
   SELECT [the value], [the area], [summed area], (([the value] * [the area]) / [summed area]) as [weighted]
   From [drawing] d
   join (Select [the line segment], sum([the area]) as [summed area] From [drawing]
    Group By [the line segment]
  ) st on st.[the line segment] = d.[the line segment]
   ```

   .....As for table aliases, they are just like algebraic variables. In the query above, st is variable reference to the entire recordset from the aggregate sum subquery. In other words, we could have broken out the SELECT (that appears inside the parenthesis) into a seperate query component and called that query (known as a view in other flavored RDBMS) "st" and the query would have worked much the same way. Or, that same subquery could be written into a SELECT ... INTO [st] where we would get a new table and again, the query would work much the same way.


   Provided by: Klaus DE

   ```sql
   SELECT [UniquePCE$].[Well], [UniquePCE$].[Chemical], [UniquePCE$].[Concen], [UniquePCE$].[Date]
   FROM ( SELECT max([Date]) AS LastDate, [Well]
      FROM [PCE$] GROUP BY [Well] ) [LastResults]
   LEFT JOIN [UniquePCE$] ON
      ([UniquePCE$].[Well] = [LastResults].[Well]
    AND [UniquePCE$].[Date] = [LastResults].[LastDate]);
   ```


   http://69.17.46.171/Site/Thread.aspx?id=26220&ti=632926356630100000

   Aliases are used for two main purposes: to resolve ambiguities and to simplify the text of the query and make it more manageable. Each alias has a scope which is determined by the place where it is defined. The keyword "AS" used to define an alias can be omitted.

   Using aliases to resolve an ambiguity:
--SQL
SELECT A.[ID], B.[ID] FROM [Data] A, [Data] B
WHERE IsPoint(A.[ID]) AND IsArea(B.[ID]) AND Contains(B.[ID], A.[ID])

Using aliases to simplify the text of the query:

--SQL
SELECT A.[ID], Min(D) FROM
  (SELECT A.[ID], Distance(A.[ID], P.[ID]) D FROM [Areas] A, [Points] P)
GROUP BY A.[ID]

4. adamw@manifold.net – Manifold©-L Tue 07/04/2006 10:34 AM

Given the following auxiliary query named "digits":

VALUES (0), (1), (2), (3), (4), (5), (6), (7), (8), (9)

...this will take the drawing named "Drawing" (single-branch lines only, no filtering is being done) and explode each line into segments:

SELECT [ID],
  NewLine(Coord([Geom (I)], N.V), Coord([Geom (I)], N.V + 1)) FROM
  [Drawing]
  INNER JOIN
  (SELECT D1.[Column]*100 + D2.[Column]*10 + D3.[Column]*1 V
   FROM [Digits] D1, [Digits] D2, [Digits] D3
  ) N
  ON N.V < CoordCount([Geom (I)]) - 1

5. bhilton at 1/25/2006 2:26 PM (#18259)
http://69.17.46.171/Site/Thread.aspx?id=27305&ti=632933357340000000

PARAMETERS Address text, City text, State text, Zip text, [Search Radius in Miles] integer;
SELECT DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') as [Distance to Location], [Your Table].*
FROM [Your Table]
WHERE DistanceToAddress(ID, Address+', '+City+', '+State+', '+Zip, 'mi') < [Search Radius in Miles]
ORDER BY [Distance to Location];
**Select a Grid Cell Based Upon a Point Location that Uses a Different Projection**

Manifold©-L
Author: Tim Packard
Date: 12-22-2005 17:29
http://lists.directionsmag.com/discussion/read.php?f=29&i=41187&loc=0&t=41187

Manifold©-L
Author: Adam Wachowski
Date: 12-23-2005 07:30

```sql
SELECT * FROM [Drawing] WHERE
    (CentroidX(Project([Geom (I)], CoordSys("Surface" AS COMPONENT)))
     BETWEEN 33 AND 34) AND
    (CentroidY(Project([Geom (I)], CoordSys("Surface" AS COMPONENT)))
     BETWEEN 25 AND 26)
```

**Finding the Center for a Large Set of Points**

http://69.17.46.171/Site/Thread.aspx?id=26625&ti=632924802508400000

johnrobot at 8/29/2006 7:56 AM (#26626)
I have a large set of points, say about 1000, and I would like to find the center coordinates for all........ Magnus

seatrails at 8/29/2006 8:31 AM (#26628)

Select Centroid(AllCoords(Geom(ID))) from Drawing

**Return Points where One Line Intersects Other Lines**

http://69.17.46.171/Site/Thread.aspx?id=18257&ti=632742887502000000

njengler at 1/25/2006 2:05 PM (#18258)
The following SQL returns a set of points where a line (ID=147547) from one drawing intersects the lines in another drawing:

```sql
SELECT IntersectionPoint(147547,[Table].[Geom (I)], [Table].[ID]) _
     FROM [Table] WHERE Intersects(147547,[Table].[Geom (I)]) = TRUE;
```
The interesting/devastating bug is that if the line (ID=147547) is drawn with the Snap on (in this case I used 'snap to lines' and others) then no intersection points are generated. If the line is drawn without Snap, the SQL works as expected.

Does anyone have any insight into this?

adamw on 1/31/2006 7:19 AM (#18483)
Does it work if you replace Intersects with Touches?

--SQL
SELECT IntersectionPoint(147547,[Table].[Geom (I)]), [Table].[ID]
FROM [Table] WHERE Touches(147547,[Table].[Geom (I)]) = TRUE;

A line is considered to intersect another object if one of its interior points (not a line end) belongs to that object.

Return Surface Height at Every Node along a Line

http://69.17.46.171/Site/Thread.aspx?id=18948&ti=632754365264370000

njengler at 2/13/2006 1:32 PM (#18973)
I have is a surface and a drawing on top of it with lines. What I am looking for is a query that can automatically return the elevation (height) value from a surface at every inflection point on a line. For example, if my line has 10 inflection points, I want to return 10 records from the query where each record contains the elevation value found at the x,y location of the inflection point.

I have the following SQL which returns each inflection point in my line. What I want to add do now is extract the Height value of a surface at each of these points (in this SQL 80689 is the [ID] of my line):

SELECT a.[g] FROM
(SELECT [g] FROM [Drawing] WHERE [ID] = 80689 SPLIT BY Coords(Segments(80689,375)) AS [g]) a

Does this make sense?

N

njengler at 2/13/2006 2:08 PM (#18974)
Solved it!
CREATE POINTS AT THE MIDDLE OF LINES

http://69.17.46.171/Site/Thread.aspx?id=20740&ti=63282401239330000

NY-Mapper at 4/30/2006 9:57 AM (#20741)
I'm trying to create a point in the middle of the road length so that I can use it for horizontal labels. I know I saw in here a way to do it, but can't find it right now.

KlausDE at 4/30/2006 2:09 PM (#20742)
Link a Drawing to the following Query:

--SQL
SELECT [Streetname],
LinePoint([ID],[Length (I)]/2) AS CentralPoint
FROM [Streets]

--Klaus

willh at 4/30/2006 2:24 PM (#20743)
Elegant, but you could also INSERT the points instead of linking in a drawing to a query. Maybe one day soon we'll be able to do a SELECT INTO [new drawing]; it would be very cool and I know it would make my life easier :)

NY-Mapper at 4/30/2006 7:47 PM (#20744)
Ok, so how do I insert the points instead of linking them.

willh at 4/30/2006 8:04 PM (#20745)
Yeah, I figured that would be coming..

Insert into [drawing] ([street name], [Geom (I)])
(select [street name], LinePoint([geom (i)], [Length (I)]/2) AS [Geom (I)]
FROM [drawing] where isline([Geom (I)]))

KlausDE at 5/1/2006 1:01 AM (#20749)
Keep in mind that the INSERT INTO or the eagerly awaited SELECT INTO [new drawing] are static solutions whereas with the additional two permanent components [query] and [linked drawing] you get the Name of a street added newly to the [streets drawing] automatically appear in the labels component as soon as the query is updated.
Transferring the Selection from a Drawing to an Image

Manifold-L
Original Question From:
Tim Packard
Date: 01-05-2006 20:11

Answer:
Author: Adam Wachowski (NA)
Date: 01-09-2006 09:54

> I need to select all pixels that touch a drawing object. I have created a grid and then selected a
> cell of the grid and transferred the selection to the surface underneath. When inspecting the
> pixels that are selected, the ones that are split with a grid line appear to not be selected.

Transferring the selection from a drawing to an image or surface selects only the pixels whose
centers are under the selected objects.

To select all pixels touching the selected objects, use the following query:

```
UPDATE (SELECT S.[Selection (I)] AS [Selection]
FROM [Surface] S, [Drawing] D
WHERE D.[Selection (I)] AND Touches(D.[ID],
AssignCoordSys(CGeom(CGeomWKB(
    "POLYGON ((" &
    S.[X (I)] & " " & S.[Y (I)] & "," &
    (S.[X (I)]+1) & " " & S.[Y (I)] & "," &
    (S.[X (I)]+1) & " " & (S.[Y (I)]+1) & "," &
    S.[X (I)] & " " & (S.[Y (I)]+1) & "," &
    S.[X (I)] & " " & S.[Y (I)] & ")))",
    CoordSys("Surface" AS COMPONENT))))
SET [Selection] = True;
```

Adam W.
Using the CASE operator in SQL Strings

Manifold-L
Author: Adam Wachowski
Date: 02-08-2006 00:55

PARAMETERS [y1] TEXT;
SELECT [Color].*, (CASE [y1] WHEN "User1" THEN [User1] ELSE 0 END) FROM [Color];

Manifold-L
Author: Adam Wachowski
Date: 02-08-2006 01:29

PARAMETERS [UserNo] INT;
UPDATE (SELECT * FROM [Region] INNER JOIN [Color] ON [Region].[Key] = [Color].[Key])
SET [Region].[User] = (CASE [UserNo]
    WHEN 1 THEN [Color].[User1]
    WHEN 2 THEN [Color].[User2]
    ...
    ELSE [Color].[User99] END)

johnrobot on 9/7/2006 7:22 AM (#27208)
http://69.17.46.171/Site/Thread.aspx?id=27205&ti=632932365368600000

UPDATE [elevation]
SET [Height (I)] =
(CASE
    WHEN [Height (I)] BETWEEN 0 AND 1000 THEN 1
    WHEN [Height (I)] BETWEEN 1000 AND 2000 THEN 1000
    ELSE ([Height (I)]*1)
    END)

Combining Spatial Operators to Extend Spatial Queries

http://69.17.46.171/Site/Thread.aspx?id=19930&ti=632792746611000000

artlembo at 3/29/2006 7:20 AM (#19980)
........ This is the SQL you are looking for:
SELECT b.* FROM a,b
WHERE
CONTAINS(a.id,b.id)
AND
TOUCHES(boundary(a.id),b.id)

here we are making sure that the "B" polygon is both inside (contained), and touching the boundary or the "A" polygon.

That is why spatial SQL is so great. It's enormously flexible. You can even add things like:

AND B.PropertyValue > 50000, etc.

So, Manifold doesn't need a contained operator like ArcGIS. Manifold gives you the flexibility to do just about anything you want, if you learn a little spatial SQL.

Selecting Duplicate Objects

http://69.17.46.171/Site/Thread.aspx?id=20092&ti=632803720515300000

adamw at 4/10/2006 7:07 AM (#20353)
An alternative idea is to create a query to select the duplicate objects, eg:

--SQL
SELECT * FROM [Drawing] WHERE [ID] NOT IN
  (SELECT First([ID]) FROM [Drawing] GROUP BY [Column1], [Column2], ...)

Extracting / Exporting Node Coordinates

http://69.17.46.171/Site/Thread.aspx?id=20579&ti=632811612619900000

I have a drawing and want the geometry to exist in a CSV file. Specifically, I'm trying to extract all points so coordinates will appear--either concatenated in a cell, or each row associated with a long/lat pair.

jkelly at 4/20/2006 7:55 PM (#20589)
1. A fairly crude way is to extract the geometry with a query using

SELECT CGeomWKB(Geom(ID)) AS [Geom] INTO [WKB Table] FROM [yourDrawing];
Then right click on the Geom column and change the format to WKT. This then gives a listing of each coordinate pair that goes to make up the shape. You could then export this out as a CSV and alter the pairs so that each one is a new line. Pretty laborious though, especially without coding.

Upon thinking about this more, there is a much easier way if you want every vertex as a separate point.

2. Use the Points operator in the transform toolbar, to create points, then export the selection as a CSV.

---

**Problem: String Fraction "7 1/2" is seen as a Date "#7/1/2002#"**

Manifold®-L  
Author: Michael Henricks  
Date: 04-28-2006 05:15  

(This has been fixed but Adam's solution is still interesting – LAK)

When query with

```
SELECT [FENAME] FROM [myTable] WHERE [FENAME]="7 1/2"
```

and FENAME is a string field

I get a column type mismatch "#7/1/2002#"

How do I prevent the query from converting 7 1/2 to a date?!

Michael Henricks  
mike@ameripages.com

---------------------------------

Manifold®-L  
Author: Adam Wachowski  
Date: 04-28-2006 09:13  

Try:

```
SELECT [FENAME] FROM [myTable] WHERE [FENAME] = "7 1/" & "2"
```

This is going to be fixed in the next update.
Adam Wachowski

**Rotation Angle for the Point Labels in Box Style**

http://69.17.46.171/Site/Thread.aspx?id=17554&ti=632724422103770000

KlausDE at 1/7/2006 3:45 PM (#17641)

To get rotation angle for the Point Labels in box style you could

1. create the centroids as Adam suggested and add the line ID or create them by a query

   Query LineCenter:
   OPTIONS COORDSYS("Lines Drawing" AS COMPONENT);
   SELECT LinePoint([ID],[Length (I)]/2) As LCenter, [ID] AS LineID, [Labelling]
   FROM [Lines Drawing]

   I want to see what happens and linked a Drawing to this query. I duplicated this Drawing so it's
   no longer bound to the Query, editable and label positions can be moved around on the line. The
   Default name of this Drawing will be [LineCenter Data 2]

2. Buffer the LineCenters and create borders of the buffers

3. Create Intersection points from the lines and the buffer borders of the same Line ID

   SELECT IntersectionPoint( Boundary(Buffer([LineCenter Data 2].[ID], 30)),
   [Lines Drawing].[ID]) AS IGeom, [LineID]
   FROM [LineCenter Data 2], [Lines Drawing]
   WHERE [LineID] = [Lines Drawing].[ID]

   4. Use (the two) points for each buffer to calculate the rotation angle. Create and update a
   rotation column in the unlinked Drawing of points and create Labels with the Text [Labelling]
   and thematic formatting of rotation using column [rotation].

   Well, that's how it should be but I've not solved step 4 because step 3 did not (only) what I
   expected. May be easier scripted than done using SQL.

Klaus

KlausDE at 1/8/2006 12:58 PM (#17656)

……The second Query above created a Buffer (with radius 30 - change to what is appropriate
for your data) and used the border of this buffer to create intersection points with the line. Let's
call that query [2-IntersectionPoints] as it's the second query and with numbered queries its
easier to keep track of what we're doing in the project pane. In column [IGeom] it carries a set of
points but we need the set decomposed to single points, because there is something curious with
these points. May be there is a way of doing this with SQL that I don't know, but linking a
Drawing to this query shows decomposed records, one for each point. Displaying this linked Drawing with the default name [2-IntersectionPoints Data] I found there always is one additional point at angel 0 of the buffer. Don't ask me why. We have to get rid of it by the next query called (see Adam's answer at the end - LAK)

3-ReduceIPoints:
--SQL
SELECT [2-IntersectionPoints Data].* FROM [2-IntersectionPoints Data], [Lines Drawing]
WHERE Touches([2-IntersectionPoints Data].[ID], [Lines Drawing].[ID])

The next task is to combine all data that we need to calculate the text angle. First permanently manually add a column [rotation], type INTEGER, to the drawing of label points, [LineCenter Data 2] from above, the unbound, editable version. It's this column that is to be updated with the text angles.

The query to collect all necessary info is named

4-CombineSlopeData
--SQL
SELECT P1.[LineID],
   (P1.[Y (I)] - P2.[Y (I)]) As DifY,
   (P1.[X (I)] - P2.[X (I)]) As DifX,
   [Rotation]
FROM [3-ReduceIPoints] P1, [3-ReduceIPoints] P2, [LineCenter Data 2]
WHERE P2.ID > P1.ID AND
   P1.[LineID] = P2.[LineID] AND
   P1.[LineID] = [LineCenter Data 2].[LineID]

The query uses the target table [LineCenter Data 2] and two instances of the real intersection points, one aliased P1, the other P2. They are combined by sharing the same LineID and P1 and P2 differ in the running intrinsic [ID] of the points. So there is only one record for each Label. To calculate the slope of the text we need the difference in X and Y of the startpoint and the endpoint of the secant through the buffer.

The last query finally calculates the rotation angle. It's an action query and you will have to confirm running it every time you have changed the position of a label point in the drawing [LineCenter Data 2] and want the labels to align to the slope of the line in the new buffer area. It's the only query you have to run. It will invoke all the other logical layers (queries and linked drawings). They only have to be there with correct names. This query is called
5-UpdateRotation

--SQL
UPDATE [4-CombineSlopeData] SET [rotation] = IIF(DifX*DifY > 0, 360 - Rad2Deg(Atn(DifY/DifX)), 180 - Rad2Deg(3.14159 + Atn(DifY/DifX)) )

That's it. Now the column [rotation] carries the necessary angles. Two versions of the formula for X>0 or X<0. With X=0 there is a little problem. There will be no record for such conditions where the label is positioned at the end of a line and there is no or only one true intersection point with the border of the buffer. So these Labels will keep the rotation they had before. Default is 0 which may be OK for very short lines.

That was real fun for the weekend to solve it in the SQL-based way of manifold :-) If I had to script it, I would call these queries by the script.

( Learn how to make use of this [rotation] in thematic formatting of Labels at the end of Help topic 'Rotating Labels'. The trick is 'Continuous shading' and values 0 -> 360 )

Klaus

adamw at 1/9/2006 10:23 PM (#17705)
I have been able to reproduce the bug with the IntersectionPoints function creating extra points, and will file a request to fix it. Thanks for finding this!

Removing Interior Holes from Complex Objects

From: adamw@manifold.net
To: manifold-l@lists.directionsmag.com
Sent: Tue 05/23/2006 10:18 AM
Subject: RE: [Manifold-l] removing interior boundaries from single drawing object after trace
> After tracing areas on a scanned image to populate a new drawing, I get drawing area objects
> that have numerous "interior" boundaries. In other words, some of the drawing's polygons have
> complex holes in them.
> I'm trying to figure out the most efficient way to eliminate all the interior boundaries in a given
> area object.

You could use SQL.

First do Normalize Topology, to decompose areas with multiple islands into separate objects and to make the branch corresponding to the outer contour of each area the first branch. Then create a new query with the text given below, and link it as a drawing:
SELECT First(G) FROM (  
    SELECT G, ID FROM [Drawing] SPLIT BY Branches([Geom (I)]) G  
) GROUP BY ID

Adam Wachowski

**Using a Different Coordinate System in a Query**

http://69.17.46.171/Site/Thread.aspx?id=21493&ti=632846602694770000

adamw at 5/23/2006 6:40 AM (#21555)

Yes, you can change the query to use UTM 34N instead of Orthographic:

```sql
--SQL
OPTIONS CoordSys("Universal Transverse Mercator - Zone 34 (N)");
SELECT [Yesterday].*, AssignCoordSys(NewPoint(CAST([Easting] AS DOUBLE),
CAST([Northing] AS DOUBLE)),
    CoordSys("Universal Transverse Mercator - Zone 34 (N)")) [GeomData]
FROM [Yesterday];
```

**Complicated Example of Select Top 3 Records**

http://69.17.46.171/Site/Thread.aspx?id=22550&ti=632863934569830000

chrismarx at 6/13/2006 7:40 AM (#22551)

Hi to all,

I have a large Drawing table with 400 points and the distances between each point and every other point in the drawing, resulting in something that looks like this

<table>
<thead>
<tr>
<th>Point1</th>
<th>Point2</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5.6</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4.2</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7.2</td>
</tr>
</tbody>
</table>

I would like to be able to select the top 3 minimum distance values from the table for each point (group of points) in the first column.
Try this (uses a single drawing named "D"):

```sql
--SQL
SELECT A.[ID], B.[ID], Distance(A.[ID], B.[ID])
FROM [D] A INNER JOIN [D] B ON B.[ID]
  IN
    (SELECT TOP 3 C.[ID] FROM [D] C WHERE C.[ID] <> A.[ID]
      ORDER BY Distance(A.[ID], C.[ID]))
```

adamw on 6/19/2006 6:24 AM (#22870)
What is the meaning of values within the "time travelled" field? Is it the amount of time required to travel to some fixed location?

adamw on 6/20/2006 8:07 AM (#22960)
I then end up with a table that has all the routes from one point to all other points (for all points) and the time it took to take that route.

Say, you have 10 points. Do you end up with a table with 10 records and 10 columns, or with a table with 100 (or 90) records and 3 columns (from, to, time).

In the latter case, you could use:
--SQL
SELECT A.[ID], B.[To], B.[Time]
C.[Time])

chrismarx on 6/20/2006 9:30 AM (#22970)
Yes, I would end up with a table with 90 records and 3 columns…..

Auto-Number / Sequential Number into an Empty Table Column

http://69.17.46.171/Site/Thread.aspx?id=23625&ti=632872452349670000

pcardoso at 6/30/2006 3:43 AM (#23626)
Is it possible to add autonumbers (increasing order) into an empty column?

KlausDE at 6/30/2006 4:31 AM (#23634)

UPDATE [Table] Set [recNo] = (SELECT Max([recNo]) + 1 FROM [Table] T2)

Duplicating the GUI Explode Transform in SQL/Script

From: Lorne [mailto:lketch@hfx.eastlink.ca]
Sent: July 3, 2006 2:08 PM
To: manifold-l@lists.directionsmag.com'
Subject: Duplicating the GUI Explode Transform in SQL

I have a drawing containing single branch polylines. You can apply the Explode transform from
the GUI to split the polylines into their constituent line segments. I can also do this via a script
by working on the pLine objects but this is slow to execute on a large drawing. SQL transforms
execute very fast in Manifold. I would like to be able to script an SQL clause that provides the
same functionality as GUI Explode. I had been experimenting with the "Split By" clause and
have looked at a number of "split by" examples on GeoReference and Manifold-L but have not
had much success. It thought I would turn to the Pros :) Is there an SQL approach to this
problem?

Lorne
I would like to be able to script an SQL clause that provides the same functionality as GUI Explode.

Given the following auxiliary query named "digits":

VALUES (0), (1), (2), (3), (4), (5), (6), (7), (8), (9)

...this will take the drawing named "Drawing" (single-branch lines only, no filtering is being done) and explode each line into segments:

```
SELECT [ID],
       NewLine(Coord([Geom (I)], N.V), Coord([Geom (I)], N.V + 1)) FROM [Drawing]
   INNER JOIN (SELECT D1.[Column]*100 + D2.[Column]*10 + D3.[Column]*1 V FROM [Digits] D1, [Digits] D2, [Digits] D3)
     ON N.V < CoordCount([Geom (I)]) - 1
```

The inner query creates a temporary table filled with numbers from 0 to 999.

Adam Wachowski

Any chance there will be more split functions added in the future? Something like:

```
SELECT [ID], [P] FROM [D] SPLIT BY LineSegments([ID]) AS [P] WHEREGeomType(ID)= 2;
```

Yes, absolutely. This is in the wishlist.

--

Adam Wachowski

Art Lembo Provided a Script Version using the Analyzer Object
Not SQL but interesting
artlembo on 9/10/2006 6:16 PM (#27363)
http://69.17.46.171/Site/Thread.aspx?id=27362&ti=632937183439630000
Creating, Reading & Writing Geoms: NewPoint vs NewPointLatLon vs CGeoWKB

http://69.17.46.171/Site/Thread.aspx?id=25835&ti=632911331541100000

njengler at 8/12/2006 9:09 AM (#25836)
Can anyone please explain to me why this works:

```
INSERT INTO [Test01_liths] ([Geom (I)], [LOC_CODE], [GSC_CODE],
[DEPTH_TOP_m], [DEPTH_BOTTOM_m])
VALUES (NewLine(NewPoint(10300,414),NewPoint(10300,390)), "9999999", "99",
0.23, 40.1)
```

But this does not:

```
INSERT INTO [Test01_liths] ([Geom (I)], [LOC_CODE], [GSC_CODE],
[DEPTH_TOP_m], [DEPTH_BOTTOM_m])
VALUES (CGeoWKB("LINESTRING(10300 414, 10300 390)")", "9999999", "99",
0.23, 40.1)
```

I have been looking at this for an hour, and can't see any reason why the WKT should not work, but Manifold returns "Can't Run Query"

Thanks

N

KlausDE at 8/12/2006 10:12 AM (#25837)
I'm not sure but I believe it's because CGeoWKB(...) is equivalent to
NEWPOINTLATLON() as long as you do not explicitly assign a projection to the Geom. It's not
NEWPOINT(). But your coordinates are astonishing for LatLon.

So Add:
ASSIGNCOORDSYS( CGeoWKB(...)), COORDSYS("Test01_liths" AS COMPONENT) 

Something I often must be reminded: The projection is set to a component and we're used to see this projection in every single Geom we create by GUI. But in scripting and SQL we must be aware that every single Geom has projection info and that it's possible to have a collection of
Geoms with many different projections in one drawing that all are displayed in the projection of the drawing component.

Klaus

KlausDE at 8/12/2006 12:22 PM (#25841)
I don't know if a special detail of my explanation is true. You can have different projections in the Geoms stored in a table with geometry. But I don't know if the Geoms of a drawing are synchronized to the projection of the drawing on import. This projection then would be the moment, when the error is generated. A case for adamw.

adamw at 8/14/2006 6:19 AM (#25878)
Writing to the [Geom (I)] field of a particular drawing will accept values in any projection, but reading from the field will always return values in the projection of the drawing.

You can have a table with a Geom field, or a query that returns a Geom field, and have each value in that field be in a different projection though.

CLOSING LINES IMPORTED FROM DXF

http://69.17.46.171/Site/Thread.aspx?id=8451&ti=632598330600000000

mikedufty at 2/10/2005 2:51 AM (#8452)
Is there any way of automatically closing a line imported from dxf so that bounded areas can be made.

KlausDE at 2/10/2005 4:22 AM (#8457)
No need to close the lines. But Normalize Topology with a reasonable precision matching the gaps and then Join Lines before creating Bounded Areas.

WillH at 2/10/2005 10:16 PM (#8481)
Run this query (after making appropriate changes), then File > Import > Drawing and choose "This project" from file type and choose the new table in the following dialog.

SELECT IIF(ISCLOSED([id]) = false,
AssignCoordSys(CGeom(CGeomWKB(
TRIM(TRAILING ")" FROM CStr(CGeomWKB(Geom(ID)))) & "," &
CentroidX(STARTPOINT(Geom(ID)))" &" &CentroidY(STARTPOINT(Geom(ID))) & ")"
)), AssignCoordSys(Geom([id]),CoordSys("my dxf" AS COMPONENT))
), AssignCoordSys(Geom([id]),CoordSys("my dxf" AS COMPONENT)))
as [Geom (I)] INTO [fixed dxf geom table] FROM [my dxf]

WillH on 8/16/2005 11:51 PM (#13821)
Ahh, and here is a slightly easier way to go about it.
UPDATE [drawing] SET [Geom (I)] = Boundary(ConvertToArea([Geom (I)])) WHERE CoordCount(id) > 2

**Using Voronoi Functions**

http://69.17.46.171/Site/Thread.aspx?id=26050&ti=632918264606100000

adamw at 8/22/2006 6:54 AM (#26281)
The Voronoi functions take a geom containing all coordinates in the desired point set and return another geom with the relevant portion of the Voronoi diagram of that point set.

This query takes the coordinates of all objects within a drawing named "D", creates a Voronoi diagram of the resulting point set, and returns it as a single geom with a lot of branches:

--SQL
SELECT Voronoi(AllCoords([Geom (I)])) FROM [D]

This query does the same but splits the returned geom into multiple geoms:

--SQL
SELECT ConvertToArea(G) FROM
(SELECT Voronoi(AllCoords([Geom (I)])) V FROM [D])
SPLIT BY Branches(V) G

This query does the same but joins data from the original drawing (works fine for point drawings):

--SQL
SELECT A.[Geom], B.[ID] FROM
(SELECT ConvertToArea(G) [Geom] FROM
(SELECT Voronoi(AllCoords([Geom (I)])) V FROM [D])
SPLIT BY Branches(V) G) A
INNER JOIN [D] B ON Contains(A.[Geom], B.[Geom (I)])

**Locating Dangle Nodes or Unconnected Lines**

http://69.17.46.171/Site/Thread.aspx?id=12859&ti=632572847400000000

antoniocarlos on 7/14/2005 10:13 AM (#12860)
Hi
Is there an easy way to identify dangle nodes or un-connected lines with one of the transforms or SQL? …..

artlembo on 7/14/2005 3:04 PM (#12876)
Well, here is how you can do it in SQL:

```
SELECT pts.ID, count(*) AS totallines
FROM
(SELECT * from [drawing] WHERE Cstr([Type (i)]) = "Point") as pts,
(SELECT * from [drawing] WHERE Cstr([Type (i)]) = "Line") as lines
WHERE TOUCHES(pts.ID,lines.ID)
GROUP by pts.ID
```

this will get you all the nodes and the number of lines they touch. Nodes with only one touch mean they are a dangle, right?

Now, you can modify it to only select those where totallines = 1, and since you are getting the ID for each point you can then find the individual nodes that are the problem.

Art

WillH on 7/14/2005 8:38 PM (#12902)
artlembo - 2005-07-14 3:04 PM

Now, you can modify it to only select those where totallines = 1, and since you are getting the ID for each point you can then find the individual nodes that are the problem.

and that would look like this:

```
SELECT pts.ID, Count(*) AS totallines
FROM
(SELECT ID FROM [drawing] WHERE IsPoint(ID)) AS pts,
(SELECT ID FROM [drawing] WHERE IsLine(ID)) AS lines
WHERE TOUCHES(pts.ID,lines.ID)
GROUP BY pts.ID
HAVING Count(*) = 1
```

and if you wanted to delete those points (I don't know if that's what you really want, just putting it out there :-) ) in one fell swoop, you could use the following:
DELETE FROM [drawing] WHERE [Drawing].[ID] IN
(SELECT pts.ID
FROM
(SELECT * FROM [drawing] WHERE IsPoint(ID)) as pts,
(SELECT * FROM [drawing] WHERE IsLine(ID)) as lines
WHERE TOUCHES(pts.ID,lines.ID)
GROUP by pts.ID
HAVING Count(*) = 1){/code - temp tag to be replaced}

artlembo on 7/14/2005 9:57 PM (#12905)
Well, I've got nothin' to do tonight, just getting ready for vacation, so here's a way to delete the
dangles by modifying some of Will's code:

delete from drawing where drawing.ID in
(select drawing.ID from drawing, (  
SELECT geom(pts.ID) as g, Count(*) AS totallines
FROM
(SELECT ID FROM [drawing] WHERE IsPoint(ID)) AS pts,
(SELECT ID FROM [drawing] WHERE IsLine(ID)) AS lines
WHERE TOUCHES(pts.ID,lines.ID)
GROUP BY pts.ID
HAVING Count(*) = 1) as Q
where touches(Q.g, drawing.ID))

basically, Will's first query returned the ID, so what I've done there is wrapped his query in parenthesis and returned the geom object instead of the ID. I called the result Q, and then did a touches clause.

So, you can see again that it's relatively easy to keep adding more sophistication. Notice that what is in parenthesis is basically a table. Will originally created it, and with Manifold instead of entering a table name, we can enter the result of a query, like the one Will had made.

artlembo on 7/14/2005 9:59 PM (#12906)
and, adding another line at the end will only delete those lines that are below a certain distance:
delete from drawing where drawing.ID in
(select drawing.ID from drawing, ( 
SELECT geom(pts.ID) as g, Count(*) AS totallines
FROM
(SELECT ID FROM [drawing] WHERE IsPoint(ID)) AS pts,
(SELECT ID FROM [drawing] WHERE IsLine(ID)) AS lines
WHERE TOUCHES(pts.ID,lines.ID)
GROUP BY pts.ID
HAVING Count(*) = 1) as Q
where touches(Q.g, drawing.ID))
and drawing.[length (i)] < 50

WillH on 7/15/2005 12:15 AM (#12909)
Nice stuff here, Art. And now for something completely different (Ni!):

SELECT pts.ID FROM
(SELECT ID FROM dangle WHERE IsPoint(ID)) AS pts,
(SELECT startpoint(ID) as g FROM dangle WHERE IsLine(ID)) AS s,
(SELECT endpoint(ID) as g FROM dangle WHERE isline(ID)) AS e
WHERE touches(pts.ID,s.g) OR Touches(pts.ID,e.g)
GROUP BY pts.ID

Create a Surface in which the Z-Factor is a Calculated Distance from the Nearest Point(s)

http://69.17.46.171/Site/Thread.aspx?id=8433&ti=632442790200000000

Scott on 2/9/2005 4:19 PM (#8434)
Is it possible to create a distance surface from points?

Scott on 2/14/2005 10:14 AM (#8565)
I need to create a surface in which the z-factor is not a height but a calculated distance from the nearest point or points. Does/Can Manifold calculate the distance a pixel is from the point file from which the surface was created?

adamw on 2/15/2005 12:01 AM (#8592)
Scott - 2005-02-14 7:14 AM
I need to create a surface in which the z-factor is not a height but a calculated distance from the nearest point or points. Does/Can Manifold calculate the distance a pixel is from the point file the surface was created?
You can do this by executing the following query and then copying and pasting the resulting table as a surface:

```sql
SELECT S.[X (I)] X, S.[Y (I)] Y,
(SELECT Min(Distance(
AssignCoordSys(NewPoint(S.[X (I)], S.[Y (I)]), CoordSys("Surface" AS COMPONENT)),
Geom(D.[ID]))) FROM [Drawing] D)
INTO T FROM [Surface] S
```

### Cross Tabulation Matrix to Compare Landcover Types (Pivot Table)

http://69.17.46.171/Site/Thread.aspx?id=1830&ti=632152165200000000

**artlembo on 3/17/2004 3:05 PM (#1831)**
I am attempting to create a cross tabulation matrix to compare land cover types ......
but its a virtual table. I have been trying to turn it into a real table so I can copy the values to Excel to compute the land cover change.

**mdsumner on 3/17/2004 10:29 PM (#1839)**
I'm well out of my depth here, but I think I got what you want by using your original query text in component "Query":

```
TRANSFORM int(sum([Area (I)])/10000) SELECT lc
FROM [Drawing]
WHERE luse_gen <> "" AND lc <> ""
GROUP BY lc
PIVOT luse_gen
```

And then running a second query with Dave's tableNew request

component "Query 2":

```
SELECT * INTO tableNew FROM (SELECT * FROM [Query]);
```

**artlembo on 3/18/2004 8:42 AM (#1851)**
Mike,
That worked perfectly! Thanks. I wanted to do it in one SQL statement, but can't seem to get it to work. But, this way is just as easy. Thanks.

Art
Find Lines that are Adjacent to a Polygon, and then Sum the Results

http://69.17.46.171/Site/Thread.aspx?id=1308&ti=632119324800000000

artlembo on 2/7/2004 1:37 PM (#1309)
.....Someone had a question on Manifold-L regarding how to find the lines that are adjacent to a polygon, and then sum the results.

There is actually a simpler way of doing it in SQL, if you have boundaries and actual areas contained in a single drawing (say "p"). This is assuming that the person normalized topology, or split the lines as Jim indicated.

```sql
SELECT * 
FROM p, p as pb 
WHERE contains (pb.id,p.id) 
AND p.parcel_id <> pb.parcel_id 
AND isline(p.id) AND isarea(pb.id)
```

Or, to just get the fields the person asked for:

```sql
SELECT p.parcel_id, pb.parcel_id, p.[Length (I)] 
FROM p, p as pb 
WHERE contains (pb.id,p.id) 
AND p.parcel_id <> pb.parcel_id 
AND isline(p.id) AND isarea(pb.id)
```

Art

Select a Unioned Geom and an Area Sum for Each Unique Attribute

http://69.17.46.171/Site/Thread.aspx?id=16161&ti=632670468940000000

dmbrubac on 11/3/2005 10:13 PM (#16173)

SELECT UnionAll(Geom(ID)), Sum([Area (I)]) AS [Area] 
FROM [A Drawing] 
GROUP BY [An Attribute]

This should give you a query that returns a Geom and an area sum for each unique attribute.
Split a line with SQL

http://69.17.46.171/Site/Thread.aspx?id=27362&ti=632935450371200000

artlembo on 9/10/2006 6:16 PM (10 hours ago) (#27363)
I want to break a bunch of lines into 50' segments. The following SQL query works when I have one line:

Update D Set [Geom (I)] =
   (Select Segments([Geom (I)],int([Length (I)]/50)) from D)

but, it doesn't work when I have more than one line. However, the "select" statement works with more than one line (it creates multiple statements). So, it appears that the UPDATE part of the statement isn't working when there is more than one line.

L. Ketch

I don't believe that Art ever posted the fix for this. I did some experimentation and found that the following SQL works for multiple lines. I believe it was just a problem with the SQL engine not being able to identify which group of segments went with which [Geom (I)] so the UPDATE failed without any warnings. The WHERE clause below resolves the issue.

UPDATE [Drawing] as D1 Set D1.[Geom (I)] =
   ( SELECT Segments( D2.[Geom (I)], int( D2.[Length (I)]/50 ) )
      FROM [Drawing] as D2
      WHERE D1.[ID] = D2.[ID]
   )

Interpolate Heights across a Series of Points that Represent a Stream Segment

http://69.17.46.171/Site/Thread.aspx?id=21111&ti=632834458707900000

grmapper on 5/15/2006 2:35 PM (#21112)
Hello All.

I could use some help. I am trying to interpolate heights across a series of points that represent a stream segment. These points have a unique primary ID (generated by Manifold), a common ID2 (generated using an active column), a bounding elevation at each endpoint of the stream segment point set and a number of internal points with an elevation currently of 0.
My initial step I think is to identify the first endpoint elevation \([\text{First}_\text{Elevation}]\), the last endpoint elevation \([\text{Last}_\text{Elevation}]\) and the total number of points \([\text{Count}]\) that make up the total stream segment based on a common identifier \([\text{ID2}]\).

The query:

```sql
SELECT First([\text{Elevation}]) AS \[\text{First}_\text{Elevation}\], Last([\text{Elevation}]) AS \[\text{Last}_\text{Elevation}\],
Count([\text{Elevation}]) AS \[\text{Count}\] FROM [\text{river}] WHERE \[\text{ID2}\] = 15057375 AND
[\text{Elevation}] >= 0
```

I have tried to do what you want using nothing but queries and here is what I ended up with:

Suppose we have a drawing named "Drawing", with the following columns: the height column ("Height"), the column which can be used to group points that belong to the same line together ("ID2"), and the column which provides the order of points on the line ("Order").

Run the following query to collect the minimum and maximum height, and the number of points in each line into a separate table:

```sql
--SQL
SELECT \[\text{id2}\], First([\text{height}]) \[\text{minh}\], Last([\text{height}]) \[\text{maxh}\], Count(*) \[\text{total}\]
INTO \[\text{stats}\] FROM \[\text{drawing}\]
GROUP BY \[\text{id2}\] ORDER BY \[\text{order}\]
```

Run the following query to interpolate the heights in intermediate line points:

```sql
--SQL
UPDATE
(SELECT f.*, c.[\text{height}], \[\text{minh}\], \[\text{maxh}\], \[\text{total}\] FROM
(SELECT d.id, d.id2, Count(*)-1 \[\text{seq}\]
FROM \[\text{drawing}\] d INNER JOIN \[\text{drawing}\] e ON d.id2=e.id2 AND
d.[\text{order}]>=e.[\text{order}]
GROUP BY d.id, d.id2) f
INNER JOIN \[\text{drawing}\] c ON c.[\text{id}]=f.[\text{id}]
INNER JOIN \[\text{stats}\] ON c.[\text{id2}]=\[\text{stats}.[\text{id2}]
SET [\text{height}] = ([\text{maxh}] - [\text{minh}]) / ([\text{total}]-1)*[\text{seq}] + [\text{minh}]
WHERE [\text{seq}]>0 AND [\text{seq}]<[\text{total}]-1
```

The innermost SELECT generates sequential numbers 0, 1, 2, ... along each line (the \[\text{seq}\] column). The next SELECT adds a writable height column from the original table as well as the columns generated by the previous query. The SET moves numbers around between columns.
It would perhaps have been easier to write a script, but what the heck... :-(

### Average Value of an Attribute Shared by a Group of Polygons, Weighted by their Areas

http://69.17.46.171/Site/Thread.aspx?id=23351&ti=632870293446070000

**chrismarx on 6/26/2006 10:03 PM (#23352)**

Hi,
I'm trying to put together a query that will give me the average value of an attribute shared by a group of polygons, weighted by their areas. It's akin to a set of variously sized polygons, each with a property value, and I'm trying to find the average value of a section of buffered road on my map, but weighted by the contribution of those polygons based on its size.

**willh on 6/27/2006 6:41 AM (#23386)**

You need to keep in mind that you're working with sets. The way you've tried to combine the two queries at the end, you're actually dividing the numerator for each record by the entire table of results of the aggregate sum table. What you need is a way to say divide this numerator by its appropriate denominator (here, the line segment determines that)

```
SELECT [the value], [the area], [summed area], (([the value] * [the area]) / [summed area]) as [weighted]
From [drawing] d
Join (Select [the line segment], sum([the area]) as [summed area] From [drawing] Group By [the line segment]
) st on st.[the line segment] = d.[the line segment]
```

### Obtaining the Lat/Long Coordinates of an Image in Relationship to the Map Display Containing that Image

**chu_man_fu on 9/27/2006 5:50 AM (#28211)**

Hello,

Is there anyway of finding the Lat/Long of an Image.
I am creating an Image like so:

```
Rect rectCAT = appCAT.NewRect(-6841902.500000, 6232438.500000,351603.343750, 4688170.500000);
mapCAT.RenderAreaTo("Image1", 1588,1088, rectCAT, true);
```
The rectCAT hasn't got the same dimensions as the image dimensions, so Manifold automatically sizes the image accordingly. I need to find out what the MaxX/Y and MinX/Y of the image is. I will then clip objects on the Map using the acquired coordinates.

Some help on this would be great.
Cheers

chu_man_fu on 9/27/2006 8:28 AM (#28229)

Hi,

Thanks in advance to anyone who was looking into this query.
I have since come up with a solution.

I created a query to select the Lat/Long values from the image from the 0, 0 pixel to the ImageWidth-1, ImageHeight-1 pixel:

```
SELECT
[TempImage].[X Offset (I)] AS OFFX, [TempImage].[Latitude (I)] AS LAT,
[TempImage].[Y Offset (I)] AS OFFY, [TempImage].[Longitude (I)] AS LON
FROM [TempImage] WHERE
( [TempImage].[X Offset (I)] = 1587 AND [TempImage].[Y Offset (I)] = 1087 )
OR
( [TempImage].[X Offset (I)] = 0 AND [TempImage].[Y Offset (I)] = 0 )
```

Cheers.

Non-SQL Approach
seatrails on 9/27/2006 8:44 AM (#28233)
seatrails on 9/28/2006 7:58 AM (#28269)

1) Grab the Image component from the ComponentSet using the image name.
2) Grab the PixelSet property from the image object
3) Grab the box property of the pixelset
4) Grab XMax, XMin, YMax, YMin properties from the Box
5) Create a new cc= Application.NewCoordinateConverter()
6) Prepare the cc object with the 'from coord sys' of the image and the 'to coord sys' of the map image
7) Create a new point from your minX, minY
8) Convert that point with the cc object.
9) Do same for max