Work Book

MS Access Queries for Database Quality Control for Time Series

GIS Exercise - 12th January 2004

Information Products for Nile Basin Water Resources Management
www.faonile.org

Table of Contents

Introduction

Requirements

This exercise requires a PC running Windows 95/98, NT/2000/XP, with MS Excel and MS Access. The necessary data are provided. The exercise presumes a basic knowledge of MS Access.

Objectives

The main objective of this exercise is to demonstrate how to use MS Access queries for basic database quality control for a time series dataset.

At the end of the exercise, the trainees will be able to:

- Identify duplicate and ambiguous records;
- Set proper primary key fields to avoid future duplicate or ambiguous records;
- Identify data gaps and records with erroneous zero values;
- Check the validity of a record set;
- Prepare time series plots;
- Identify duplicate data.

Task

Identify data errors in a sample data set.

A program for database quality control of time series data is presented in QC-1. It describes basic, advanced, and final quality control steps. In this exercise we will concentrate on the basic level. We will use MS Access to identify structural data errors.

Step 1: Check for Ambiguous Records

Exercise 1-1: Study QC-2: Ambiguous Records.

A sample dataset named "DBQC_Wshp_CourseData.mdb" is stored in the C_Data/DB_QC folder on the workshop CD. It contains two tables:

- DBQC_Rainfall_SampleDataSet
- DBQC_Stations_SampleDataSet

Exercise 1-2: Identify ambiguous records in the "DBQC_Rainfall_SampleDataSet" table.

The dataset contains a total of 6 ambiguous records. Save this dataset in a separate table using a "Make Table" query. For an actual database, one will have to check the original records to determine the true value for each individual ambiguous record.

Step 2: Check for Duplicate Records

Exercise 2-1: Study QC-3: Duplicate Records

Exercise 2-2: Identify all duplicate records in the "DBQC_Rainfall_SampleDataSet" table.

Step 3: Set Proper Primary Key.

Exercise 3-1: Copy the "DBQC_Rainfall_SampleDataSet" table and set a proper primary key in the new table to eliminate ambiguous and duplicate records. Please note that, for an actual database, one has to correct the ambiguous records as discussed above.

Step 4: Check for Empty Records

Exercise 4-1: Study QC-4: Systematically Identifying Empty Records

Exercise 4-2: Identify empty records in the "DBQC_Rainfall_SampleDataSet" table by making a cross-tab query for monthly totals of empty records per year and station.

The database manager has to decide whether or not to remove empty records from the dataset. As the size and completeness of a dataset is often expressed as number of station years, the manager should at least be aware of the existence of these empty records, which will exaggerate the available data.

Step 5: Check for Erroneous Zero Values

Identifying zero value records is similar to identifying empty records. In this case, however, use "0" as criteria instead of "Is Null".

Exercise 5-1: Make an inventory of the zero value records in the "DBQC_Rainfall_SampleDataSet" by making a cross-tab query for monthly totals of zero-value records per year and station.

Notice the years 1994 and 1995 for the 1290001 station, implying completely dry years, which clearly represent erroneous zero values.

Step 6: Identify Data Gaps

Exercise 6-1: Study QC-5: Identifying Data Gaps

Exercise 6-2: Determine the number of complete station years.

This info, however, could be improved by looking at the completeness of individual months.

Exercise 6-3: Make an inventory of systematic data gaps in the "DBQC_Rainfall_SampleDataSet" by making a cross-tab query for monthly record counts per year and station.

Note the structural data gap for the period August to December for station 1290004. This is a strong indication for a consistent data entry omission.

Step 7: Check Validity of a Record Set

Typical errors encountered in time-series data sets include using erroneous units. For example, a rainfall measurement was recorded in [cm], but is entered into a database as [mm]. This results in an error with a factor 10. A similar example is a rainfall event recorded in 0.1 [inch], and subsequently stored in the database as [mm]. The error factor in this case is 2.54, which is much more difficult to identify.

Exercise 7-1: Study QC-6: Validity Check
Exercise 7-2: Study QC-7: Plotting Time Series

Exercise 7-3: Check the "DBQC_Rainfall_SampleDataSet" for structural erroneous records.
Plot the various stations with daily rain events exceeding 100 mm to identify structural data entry mistakes.

Step 8: Prepare Time Series Plot per Station

Exercise 8-1: Plot the time series of two arbitrary stations and examine the results on possible data errors.

Step 9: Check for Duplicate Data

The existence of duplicate records forms another example of a structural error regularly encountered in a time-series data set. In this case, the same data has been entered for different stations or years. Checking the data set for identical monthly and annual totals is an effective method for identifying duplicate records.

Exercise 9-1: Study QC-8: Identical Monthly and Annual Totals

Exercise 9-2: Check the "DBQC_Rainfall_SampleDataSet" for duplicate records both per year, and per month

This concludes the basic database quality control program using MS Access queries. This needs to be followed by advanced control, as indicated in QC-1.

Exercise 10-1: Make an inventory of all structural data errors in "DBQC_Rainfall_SampleDataSet"

Miscellaneous

Naming Convention for Tables and Files: Database development is a continuous process. New records will be appended to existing tables, data will be reviewed and corrected, and stations will be established and closed.

It is therefore necessary to develop a naming system to keep track of changes in the data files and tables. Failure to establish and follow such system could lead to confusion and loss of data.

Exercise 11-1: Study QC-9: Naming Convention for Tables and Files

QC-1: Quality Control Program for Time Series Data

Basic Quality Control for Time Series Data

STEP 1: Check for ambiguous records; correct as necessary.

STEP 2: Check for duplicate records; remove as necessary.

STEP 3: Set proper primary key fields to avoid future duplicate or ambiguous records.

STEP 4: Check for empty records; remove as necessary.

STEP 5: Check for erroneous zero values; remove as necessary.

STEP 6: Identify data gaps; fill data gaps; priority should be given to important stations, and for those with long time series.

STEP 7: Check validity of record set; make corrections as necessary.

STEP 8: Prepare a time series plot per station; identify errors, and make corrections as necessary; this complements step 6.

STEP 9: Check for duplicate data through comparing monthly and annual totals; correct as necessary.

STEP 10: Add most recent data to complete/update time series.

Advanced Quality Control for Time Series Data

- Check for identical data values in consecutive days.
- Check for unrealistic changes of data value per time interval (especially for hourly data).
- Check for erroneous truncation of peak flows.
- Prepare double mass curves for neighboring stations.
- Prepare spatial presentation of monthly and annual values, both per time step and as averages.

Final Quality Control for Time Series Data

- Use data sets for hydro-meteorologic analysis and rainfall – runoff modeling.

QC-2: Ambiguous Records

What are ambiguous records?

Ambiguous records are two records referring to the same instant in time and the same station, but featuring two distinct values.

The figure below shows ambiguous records for 9 and 10 January 1992. Please note the distinct data value for the same date and station-ID.

HMRainfall_CorrectedMay03 : Table

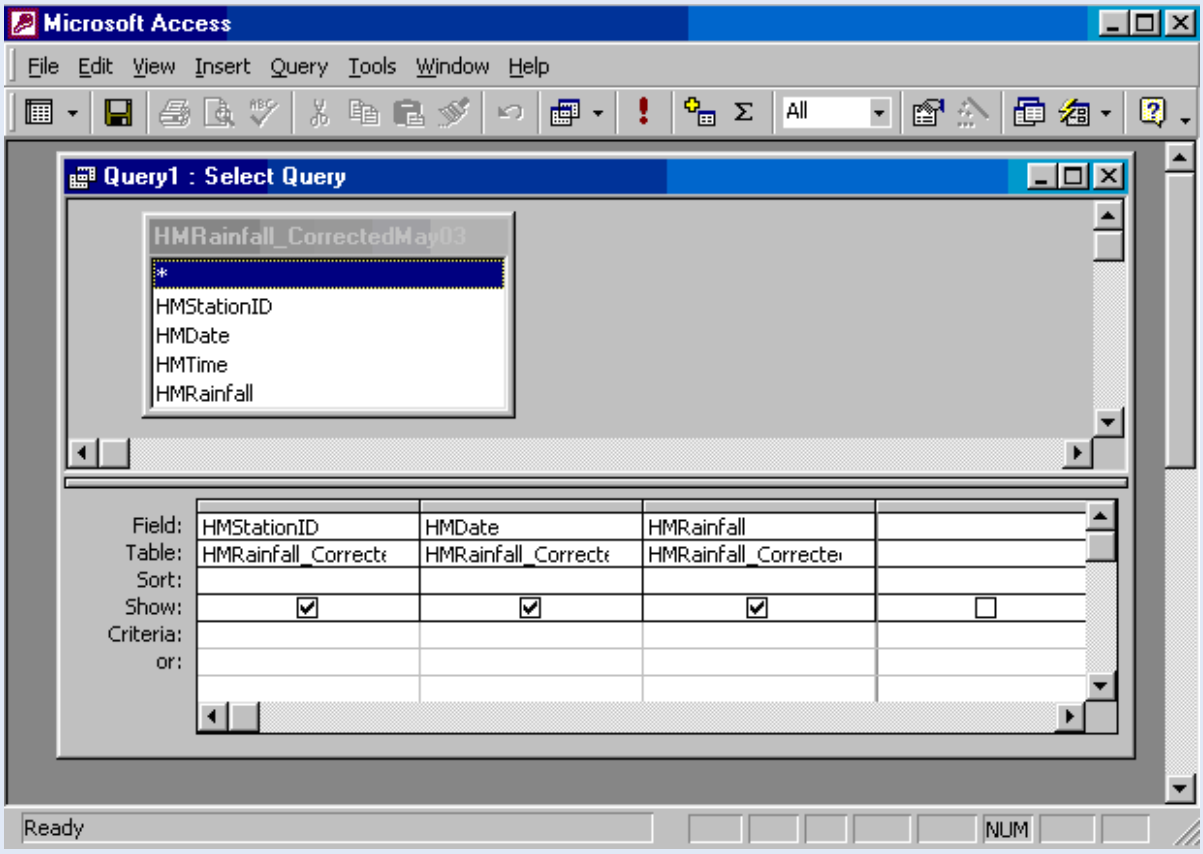
HMStationID	HMDate	HMTime	HMRainfall
10014	09-Jan-92	8:00 AM	2.8
10014	09-Jan-92	8:00 AM	4.0
10014	10-Jan-92	8:00 AM	0.2
10014	10-Jan-92	8:00 AM	0.9
10014	11-Jan-92	8:00 AM	0.0
10014	12-Jan-92	8:00 AM	1.1
10014	13-Jan-92	8:00 AM	0.2
10014	14-Jan-92	8:00 AM	0.2
10014	15-Jan-92	8:00 AM	15.0
10014	16-Jan-92	8:00 AM	16.5
10014	17-Jan-92	8:00 AM	5.5
10014	18-Jan-92	8:00 AM	0.0
10014	19-Jan-92	8:00 AM	0.5
10014	20-Jan-92	8:00 AM	0.0
10014	21-Jan-92	8:00 AM	1.8
10014	22-Jan-92	8:00 AM	0.0

Record: 103117 of 1204896

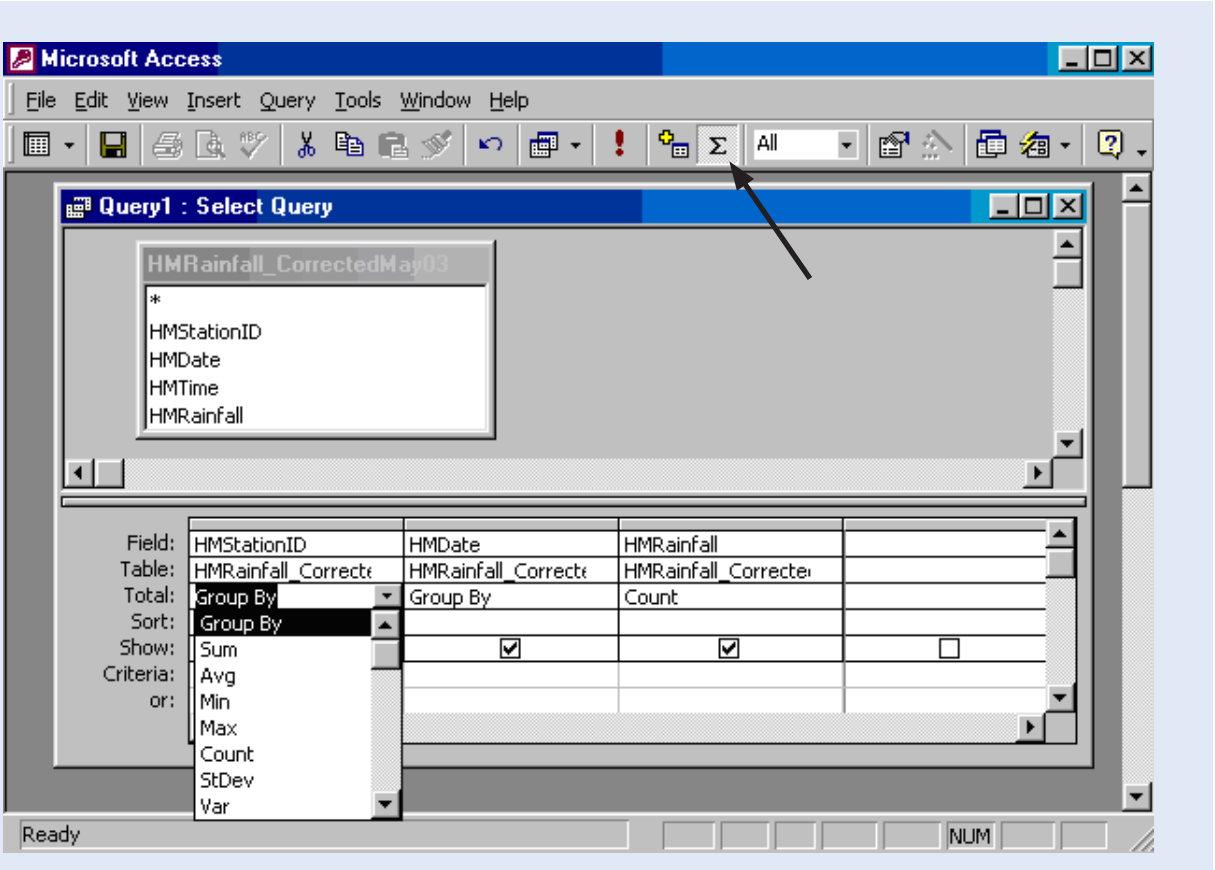
How to identify ambiguous records?

Through a Select Query using the Group By function.

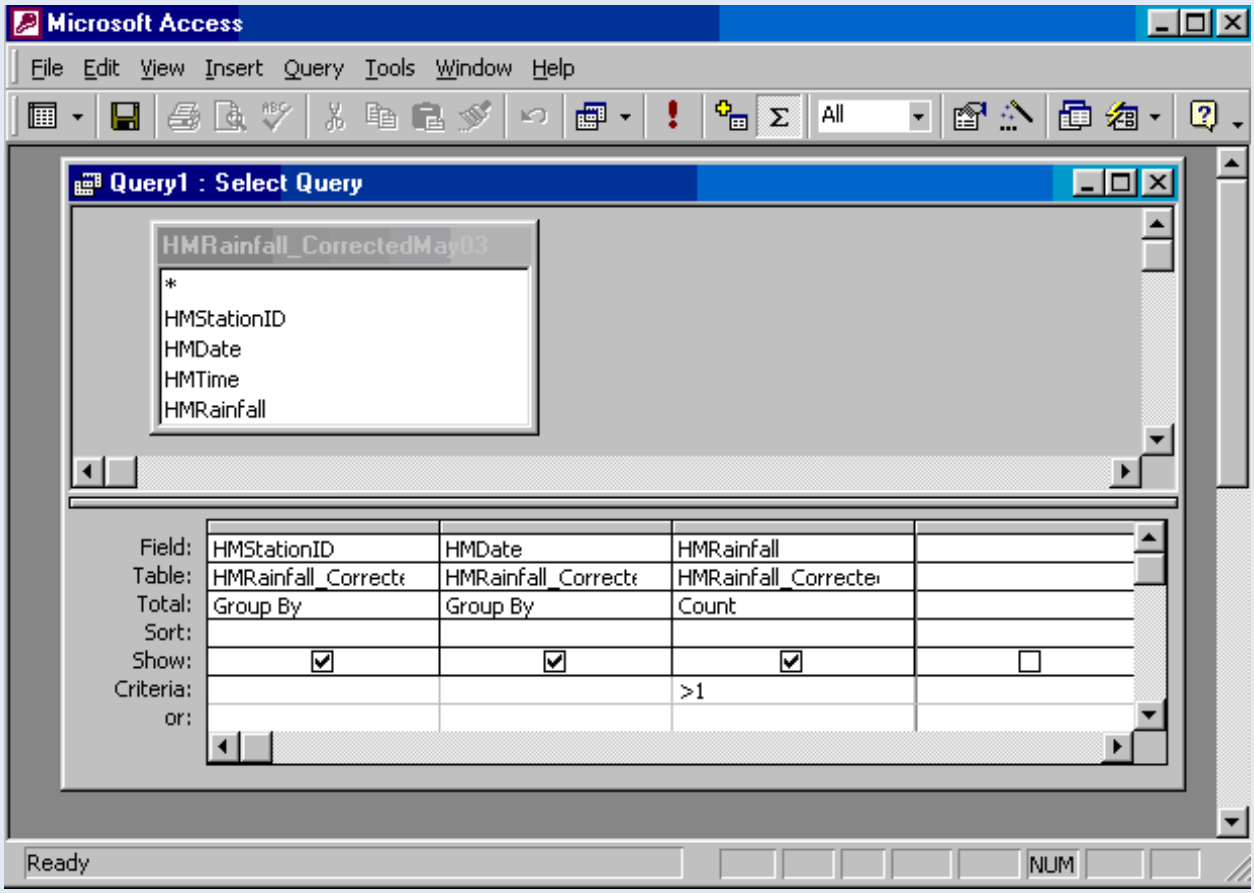
Step 1: Create a new query in design view; add the appropriate data table; choose "Select Query"; drag the "Station ID", "Date", and "Data" (in this example "HMRainfall") fields to the query window, as shown below.



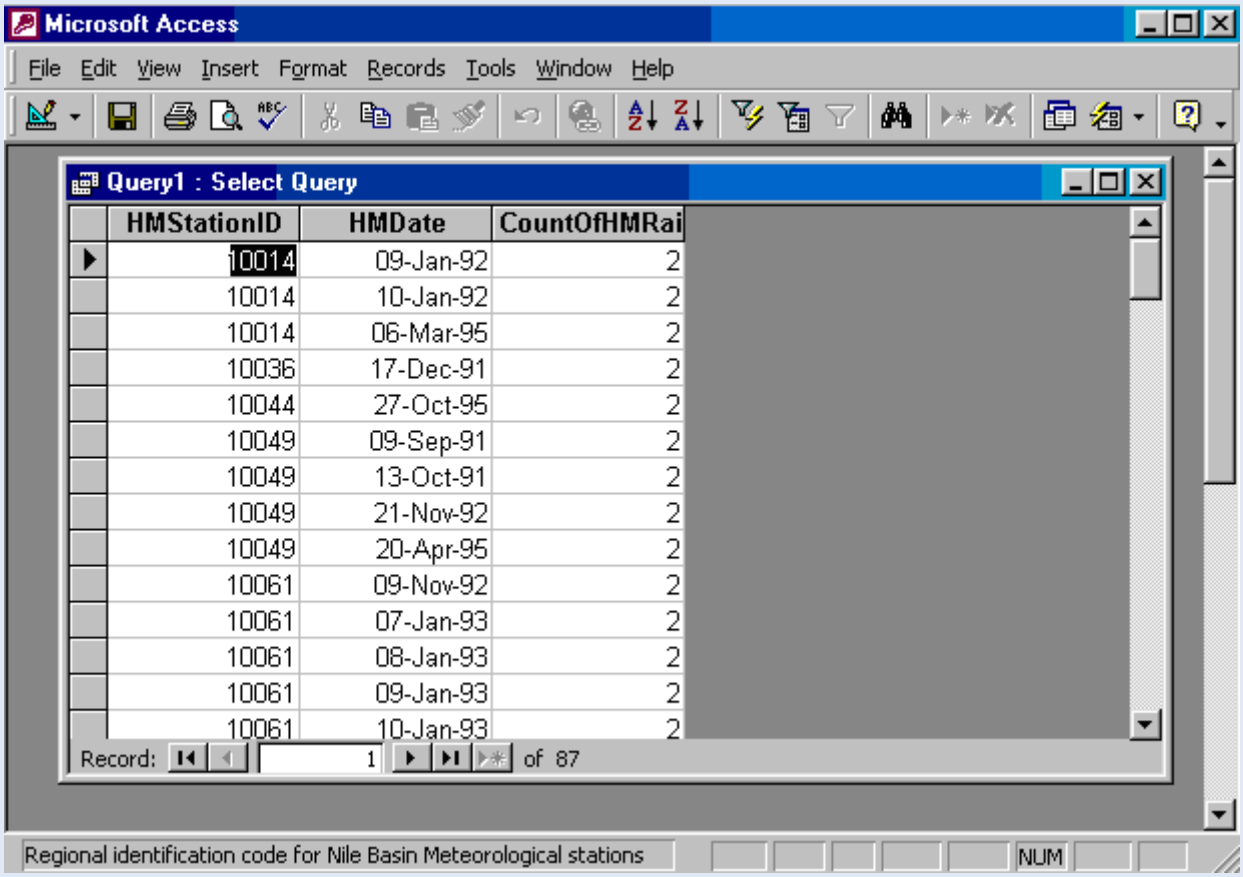
Step 2: Click the "Totals" button to activate the "Group By" functions, as shown below.



Step 3: select "Group By" for the Station ID field; select "Group By" for the Date field; select "Count" for the Data field (in this example "HMRainfall"). Enter ">1" in the criteria field, as shown below.



Step 4: Run the query. An example of the output screen is presented below. This particular data table contains 87 ambiguous or double records.

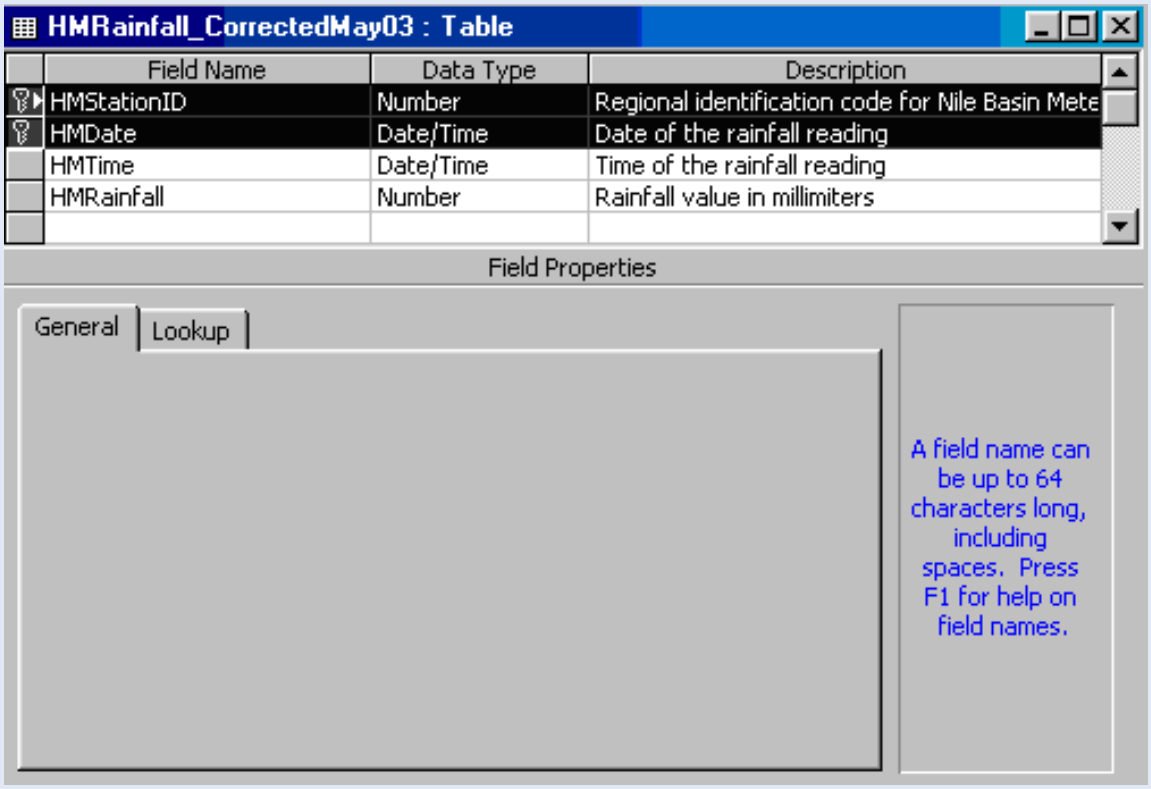


How to correct ambiguous records?

There is no automated method for correcting ambiguous records. The database manager needs to go back to the original paper records to determine the true data value. This is a tedious but unavoidable exercise.

How to avoid creating ambiguous records?

Through setting a multiple-field primary key consisting of the "Station ID" and "Date" fields. Entering two or more distinct data values is now no longer possible. This primary key setup is shown below.


HMRainfall_CorrectedMay03 : Table
Field Name
Data Type
Description
HMStationID
Number
Regional identification code for Nile Basin Mete
HMDate
Date/Time
Date of the rainfall reading
HMTime
Date/Time
Time of the rainfall reading
HMRainfall
Number
Rainfall value in millimiters
Field Properties
General
Lookup
A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

QC-3: Duplicate Records

What are duplicate records?

Duplicate records are two or more identical records in a data table. They refer to the same instant in time and the same station, and have an identical data value.

The figure below shows a duplicate record for 21 March 1991 at 6:00 AM.

HMRelativeHumidity : Table

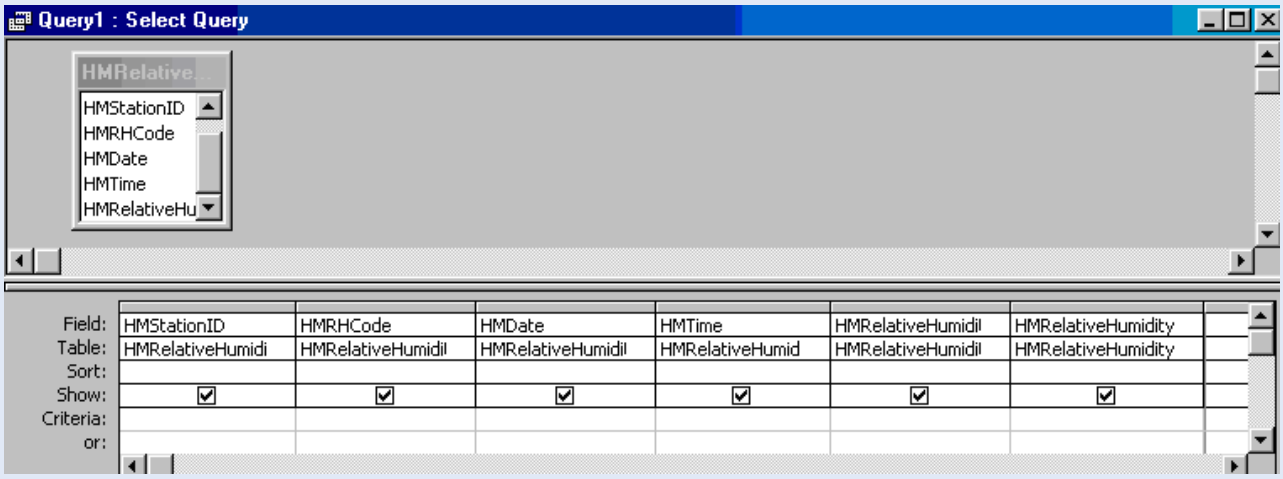
HMStationID	HMRHCode	HMDate	HMTime	HMRelativeHu
10011	1	21-Mar-91	1:00 PM	56
10011	1	21-Mar-91	11:00 AM	57
10011	1	21-Mar-91	9:00 AM	71
10011	1	21-Mar-91	8:00 AM	88
10011	1	21-Mar-91	7:00 AM	93
10011	1	21-Mar-91	6:00 AM	95
10011	1	21-Mar-91	6:00 AM	95
10011	1	21-Mar-91	12:00 AM	57
10011	1	22-Mar-91	9:00 AM	68
10011	1	22-Mar-91	4:00 PM	57
10011	1	22-Mar-91	5:00 PM	51
10011	1	22-Mar-91	3:00 PM	63
10011	1	22-Mar-91	2:00 PM	63
10011	1	22-Mar-91	1:00 PM	54

Record: 1038 of 1089445

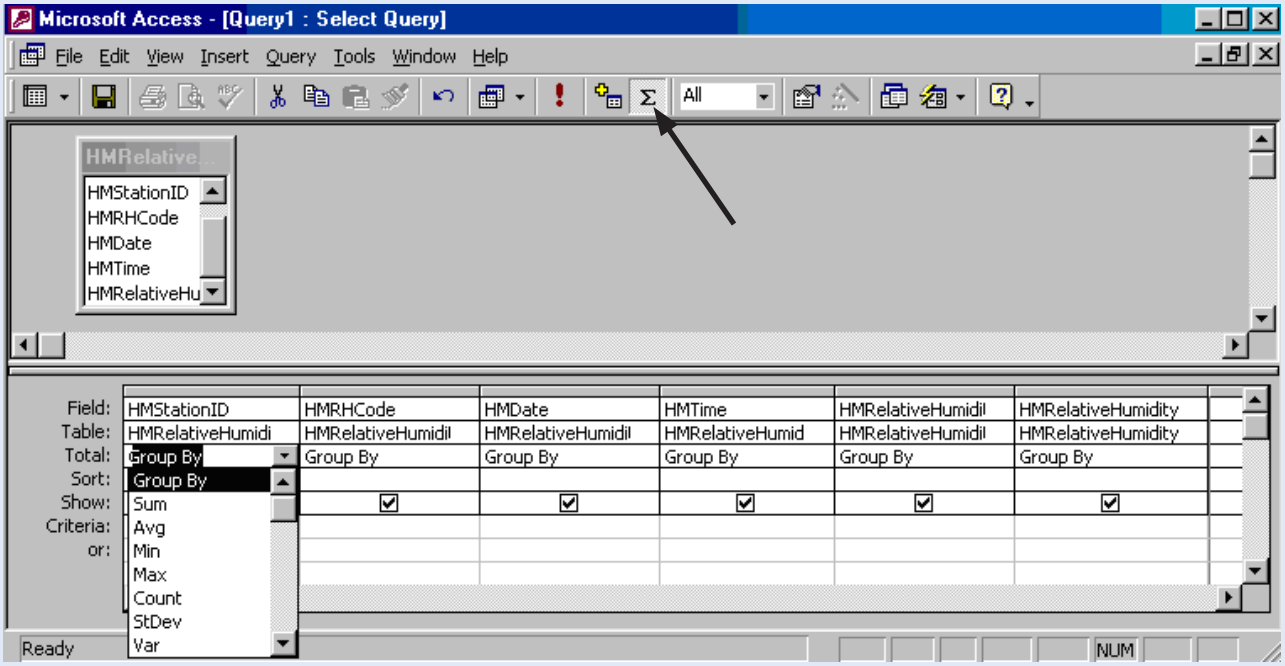
How to identify duplicate records?

Through a Select Query using the Group By function.

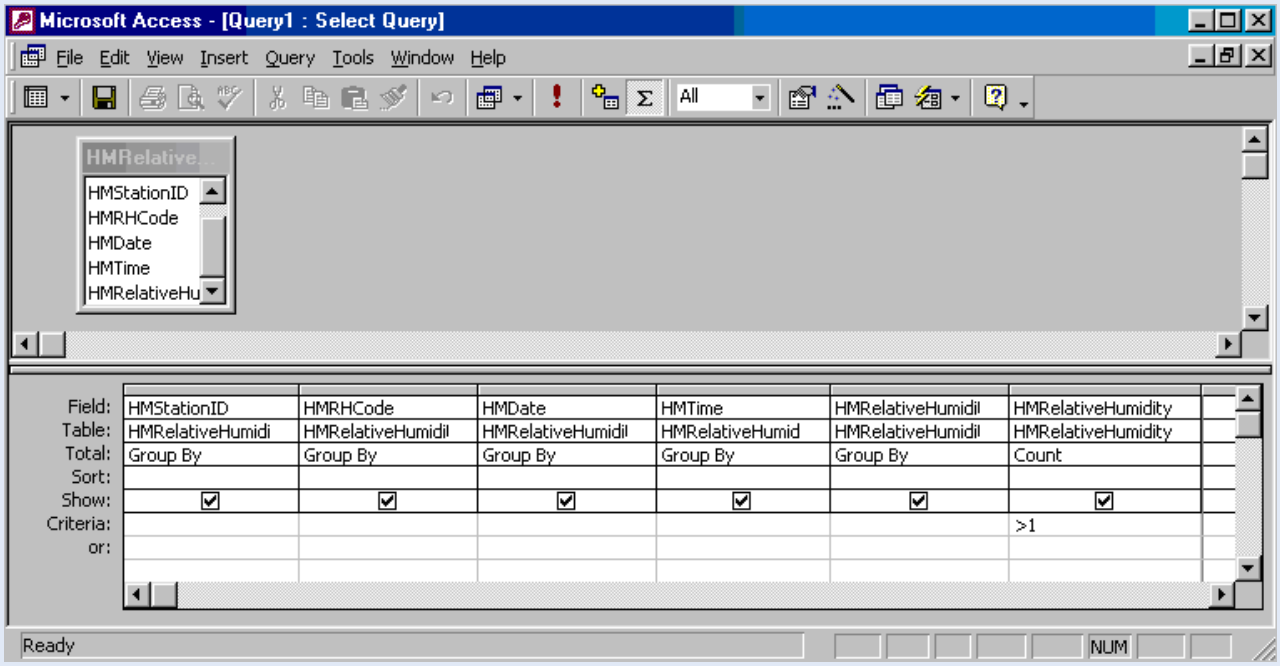
Step 1: Create a new query in design view; add the appropriate data table; choose "Select Query"; drag all table fields to the query window (in the below example: "HMStationID", "HMRHCode", "HMDate", "HMTime", and "HMRelativeHumidity"). Add a second instant of the data field (in this case the "HMRelativeHumidity" field). This is shown in the figure below.



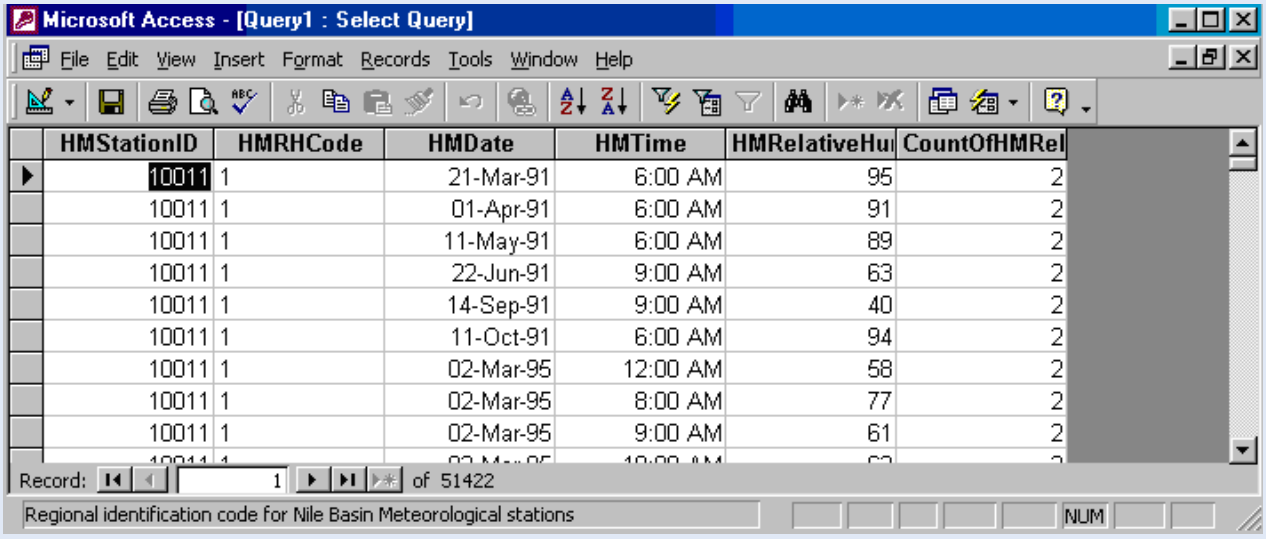
Step 2: Click the "Totals" button to activate the "Group By" functions, as shown in the figure below.



Step 3: Select "Group By" for all table fields; select "Count" for the second copy of the data field (in this case "HMRelativeHumidity"); enter ">1" in the criteria field, as shown below.



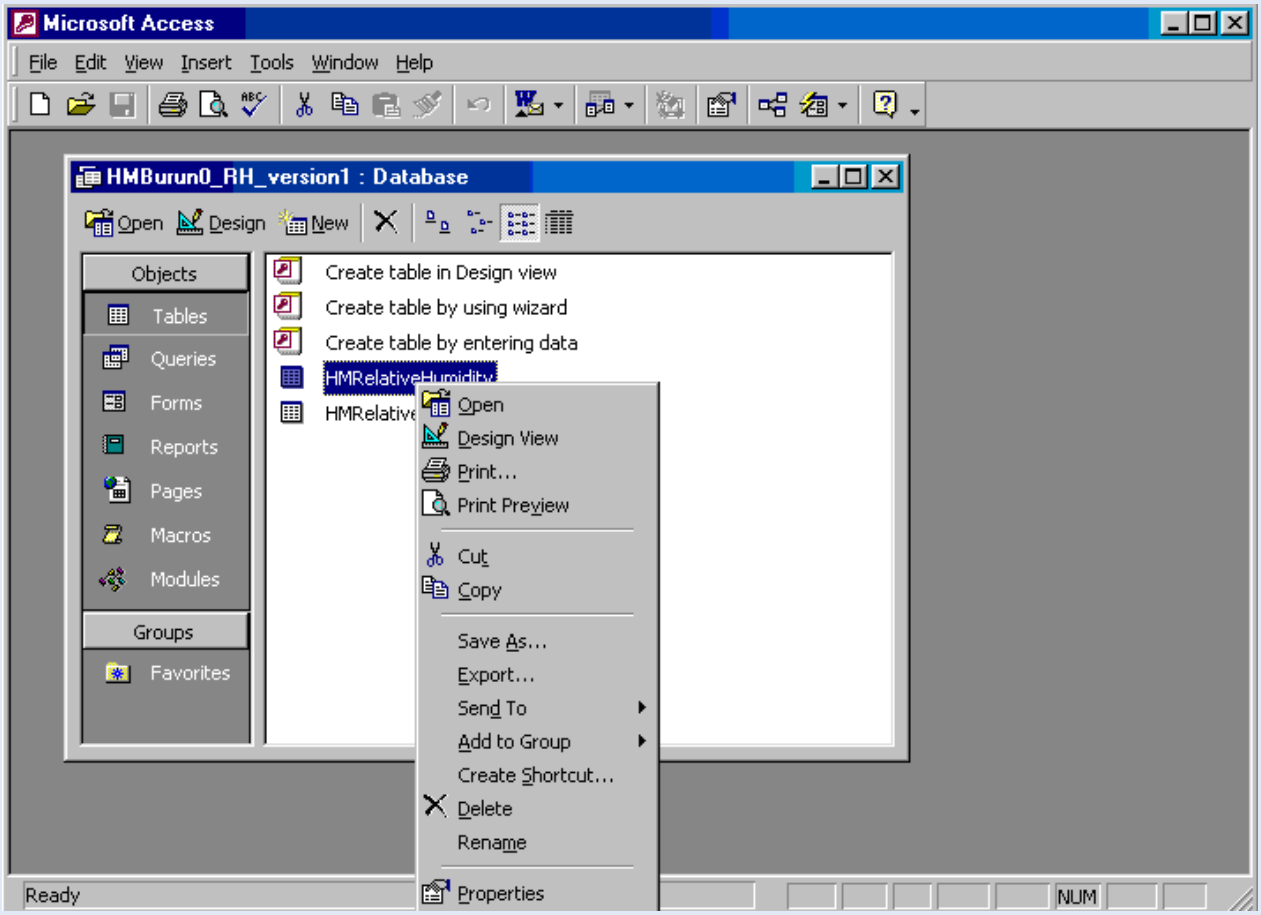
Step 4: Run the query. An example of the output screen is presented below. This particular data table contains 51422 duplicate records.



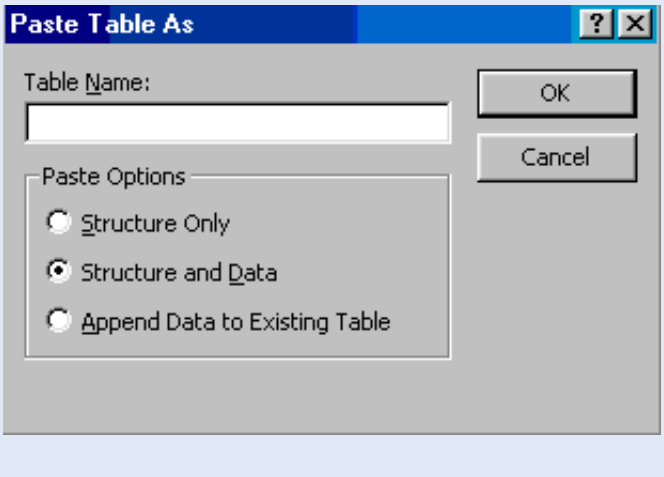
How to correct duplicate records?

By appending the entire data set to a new table with properly set primary key fields. The primary key is the field, or combination of fields, that is used to uniquely identify each record in the table.

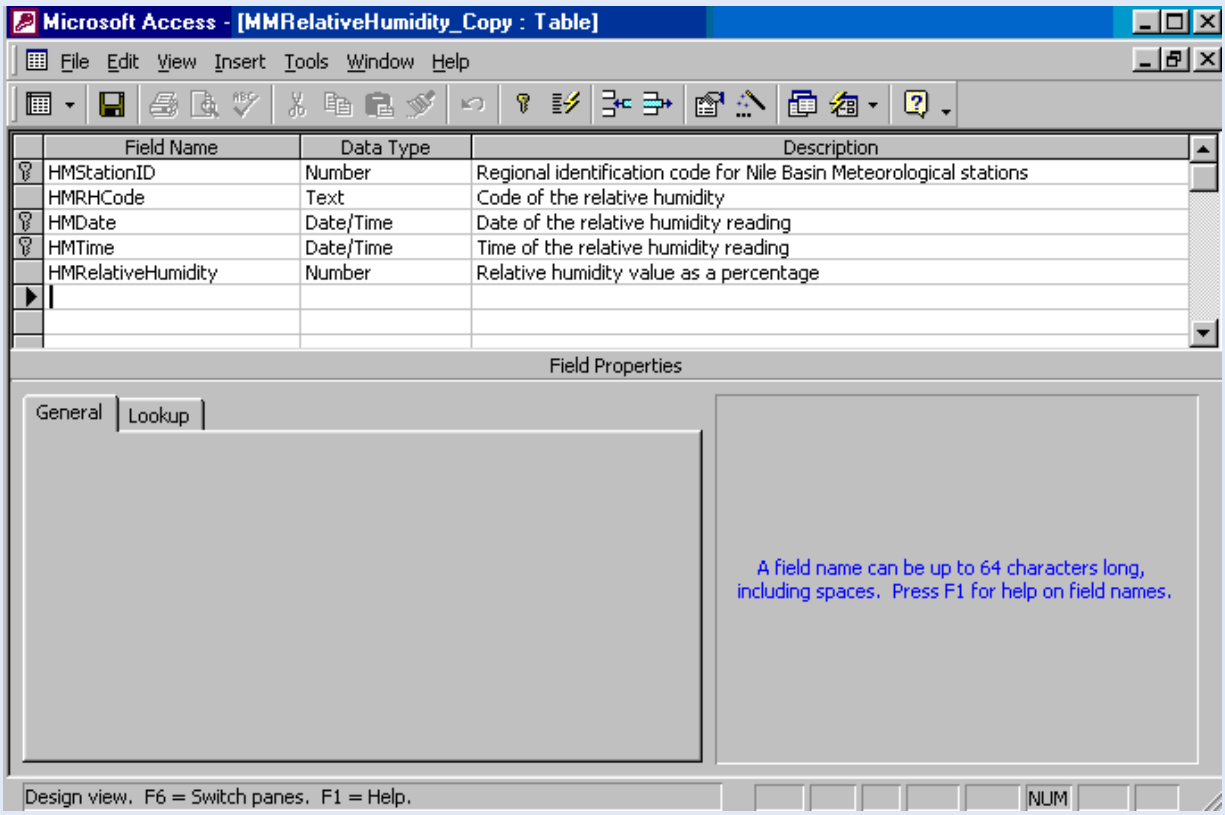
Step 1: Make a copy of the data table by clicking the 'right' mouse button and select "Copy", as shown in the figure below. Then select "Edit" from the menu bar and click "Past".



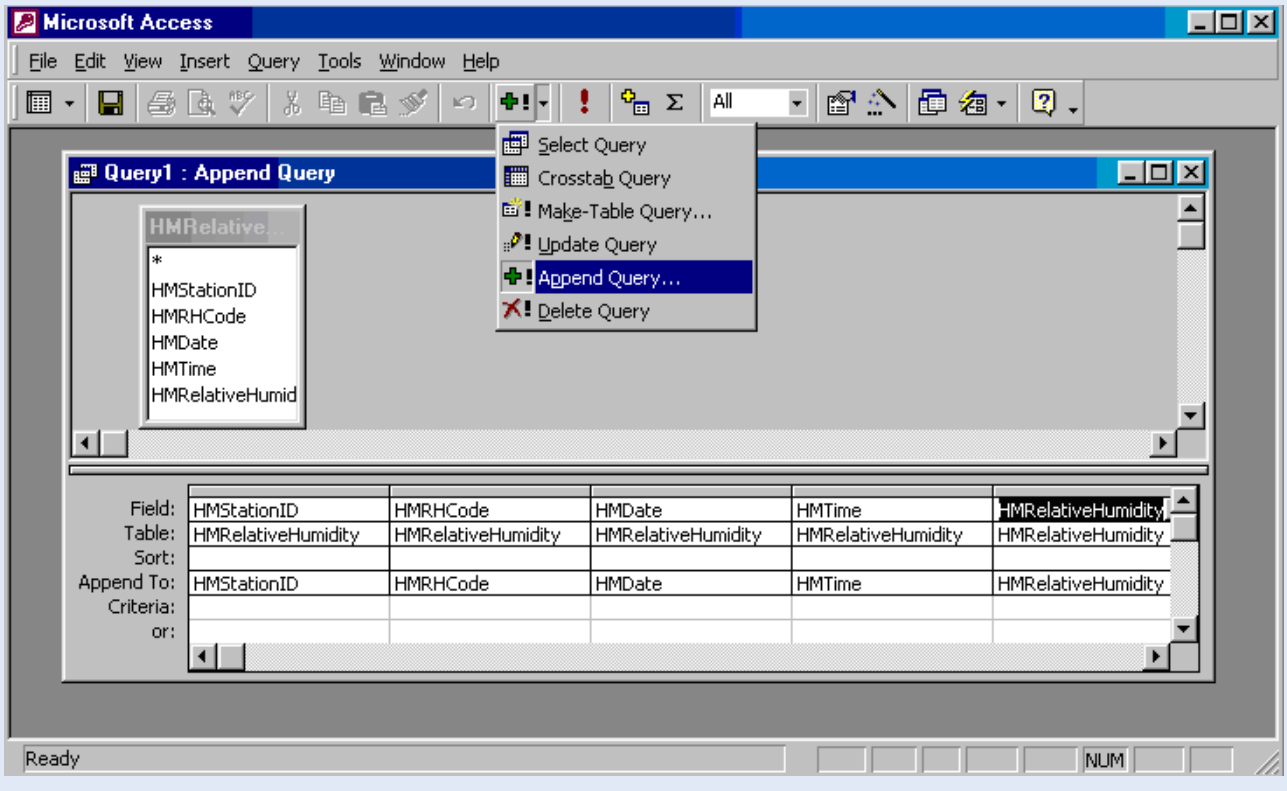
The following window appears. Select "Structure Only", and type in an appropriate name for the data table. The new table is empty and does not contain data. It has, however, an identical structure to the original data table.



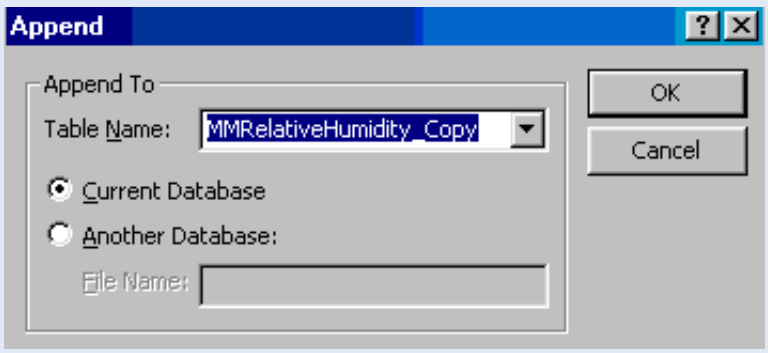
Step 2: Set the appropriate primary key fields. Open the new table in "Design View". Set the appropriate 'primary-key fields' by selecting the field, and then click the Primary Key button on the toolbar. Set a Primary Key for all fields needed to uniquely identify each record. For this example these are: "HMStationID", "HMDate", and "HMTime". Hold the "Ctrl" key to set a multi-field primary key. Save the table. The result is illustrated below.



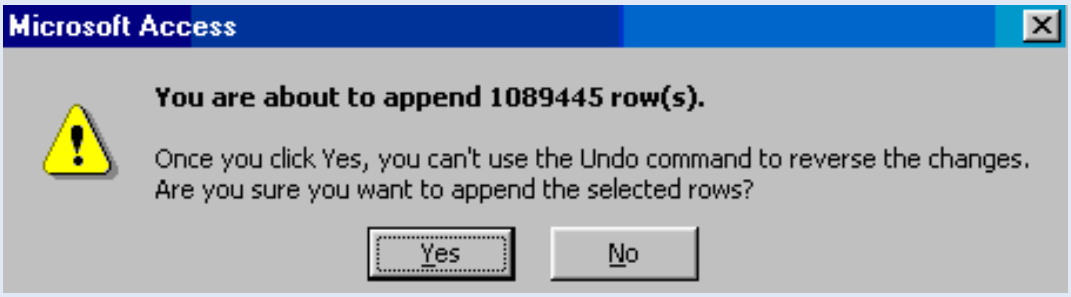
Step 3: Add data to new table through an "Append Query". Create a new query in design view. Add the original data table (containing the duplicate records). Drag all table fields to the query. Select "Append Query", as illustrated below.



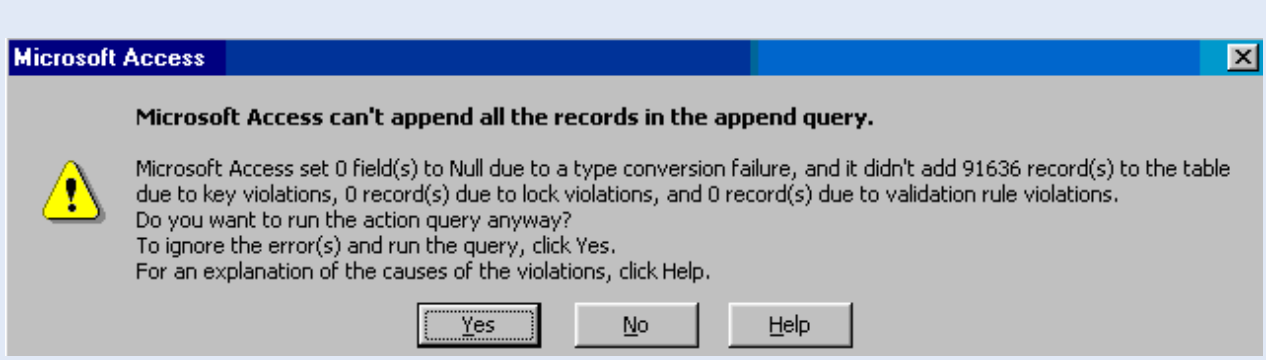
A window appears, as shown below, asking for the name of the table to append to. Select the newly created (empty) table with the appropriate primary key settings. Click OK.



Run the query. A window appears, as shown below, asking the user to append the selected rows. Click Yes.



In case of duplicate records, the following message appears:



This message indicates that 91636 records cannot be appended due to key-violations; these were the duplicate records. This number differs from the duplicate records identified in the query because some records have multiple duplicates. Click Yes.

Only unique records are now stored in the new table. Change this table name as appropriate.

The figure below shows that the double record for 21-March-91 has disappeared.

MMRelativeHumidity_Copy : Table

HMStationID	HMRHCode	HMDate	HMTime	HMRelativeHu
10011	1	21-Mar-91	1:00 PM	56
10011	1	21-Mar-91	2:00 PM	60
10011	1	21-Mar-91	3:00 PM	59
10011	1	21-Mar-91	4:00 PM	61
10011	1	21-Mar-91	5:00 PM	70
10011	1	21-Mar-91	6:00 PM	70
10011	1	22-Mar-91	12:00 AM	56
10011	1	22-Mar-91	6:00 AM	94
10011	1	22-Mar-91	7:00 AM	95
10011	1	22-Mar-91	8:00 AM	82
10011	1	22-Mar-91	9:00 AM	68

Record: 1039 of 997809

QC-4: Systematically Identifying Empty Records

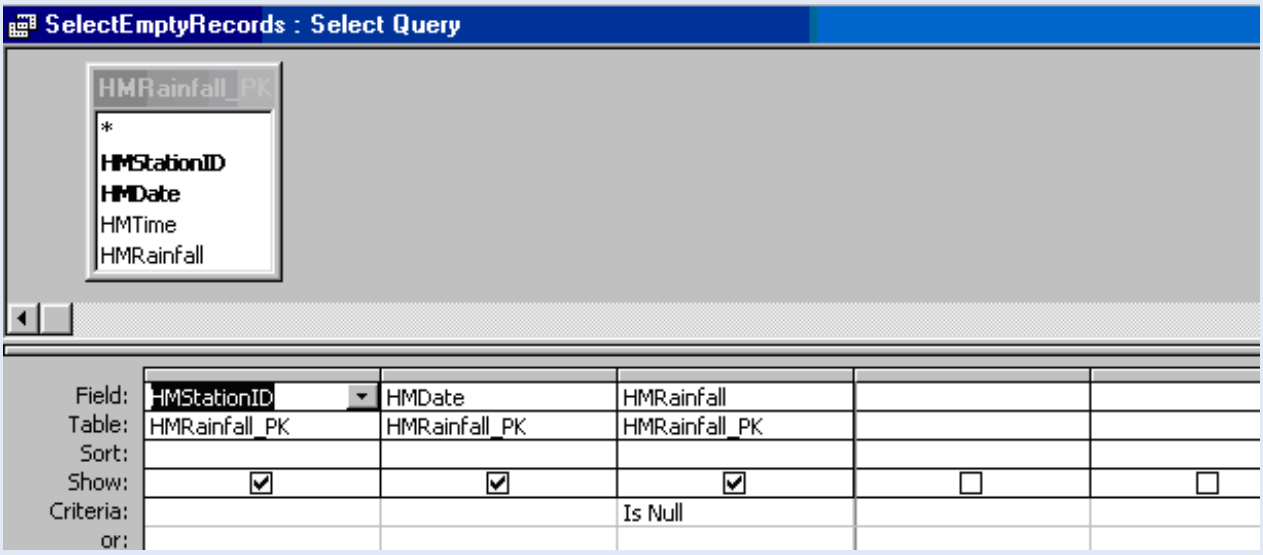
What are empty records?

An empty record is a record without data, containing the Null value. Such records can be included in the data set to indicate missing, unknown, or erroneous data values. Obviously, the Null value has an entirely different meaning than a "0" measurement of a hydro-meteorological phenomenon.

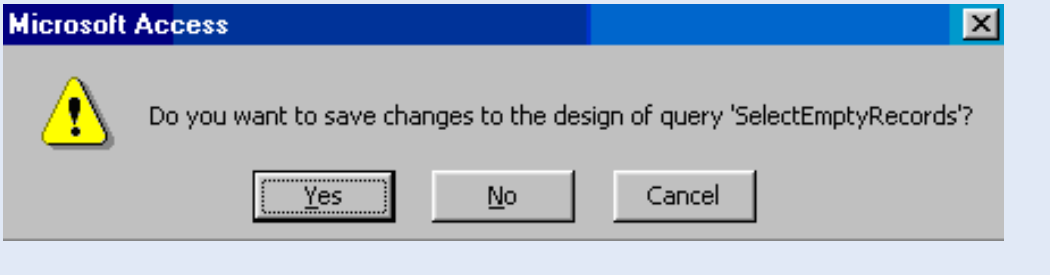
One is interested in systematically identifying empty records for quality control purposes and to check the completeness of a record set.

How to identify empty records

Step 1: create a new 'Select Query' in design view; add the data table that you want to analyze; add the relevant table fields to the query design grid, in this case "HMStationID", "HMDate", and "HMRainfall"; in the criteria cell of the appropriate data field (in this case "HMRainfall"), type the expression: Is Null. This is illustrated in the following figure.

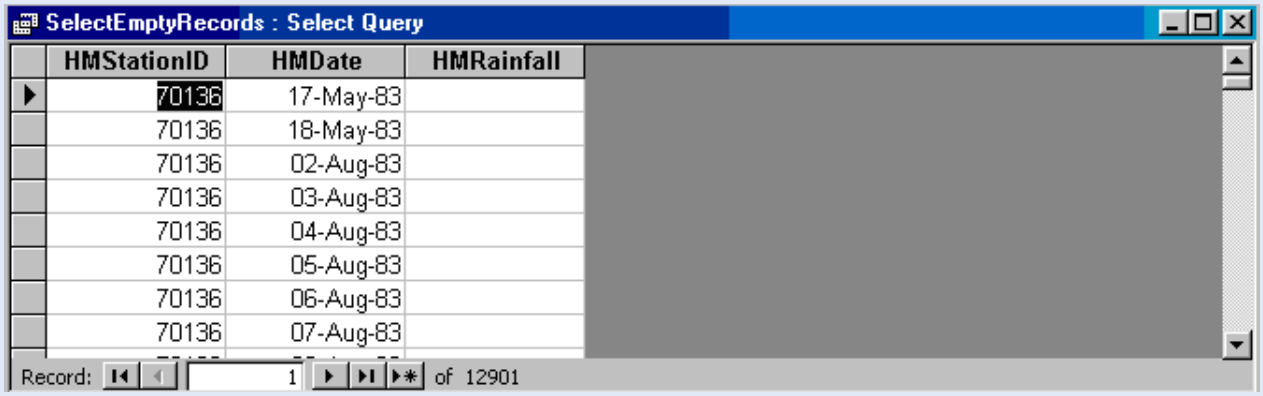


Step 2: save the query and give it an appropriate name (in this example "SelectEmptyRecords"), as shown below. Saving this query is necessary for the next analysis, which is making a systematic inventory of the empty records.



Step 3: run the query; a sample of the query results is presented below. It shows that this particular table contains 12,901 empty records. Although useful information, it will not assist a database manager to assess the completeness

of the various time series, and the nature of the missing values.

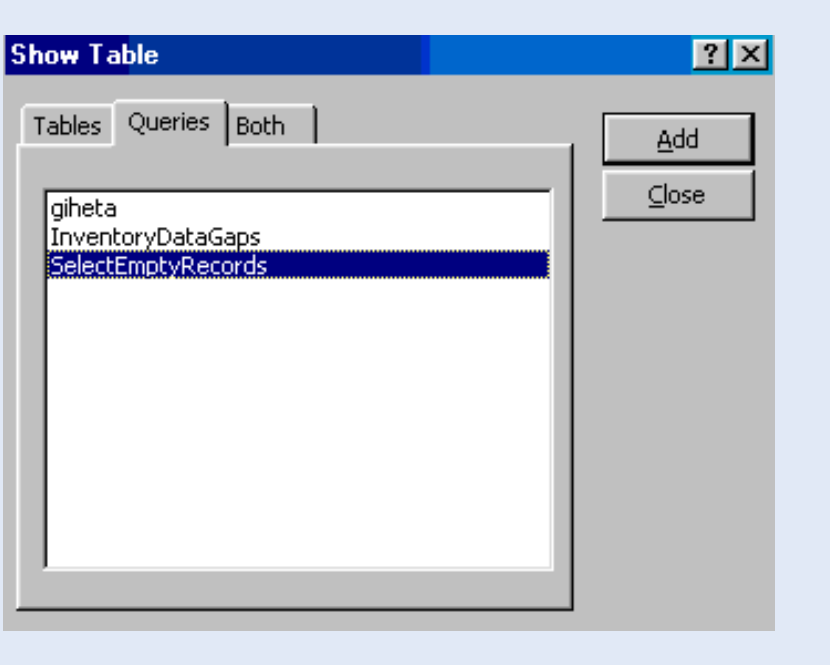


How to make an inventory of empty records

One obviously wants to know more about the nature of the empty records to assess the quality, and thus value, of the time series. Single empty records may indicate extreme events, while complete missing months may implicate data processing and entry difficulties or omissions.

An embedded cross-tab query using the "Format", "Group By", and "Nz" functions can provide useful information.

Step 1: create a new query in design view; add the "SelectEmptyRecords" query created in Step 2 of the above paragraph, shown below.



Step 2: drag the relevant table fields to the query window (in this particular case "HMStationID", two instances of "HMDate", and "HMRainfall").

Step 3: use the "Format" function to identify discrete time periods (in this case both months and years). The format function has the following syntax:

YEAR: Format([HMDate],"yyyy") : to identify the respective years in 4 digits

MONTH: Format([HMDate],"mm") : to identify the respective months in 2 digits

Step 4: use the "Nz" function to transfer the Null value to an integer code (for instance 9999). This is necessary because "Group By" functions, to be used later, cannot work on Null values. The syntax of the "Nz" function is as follows:

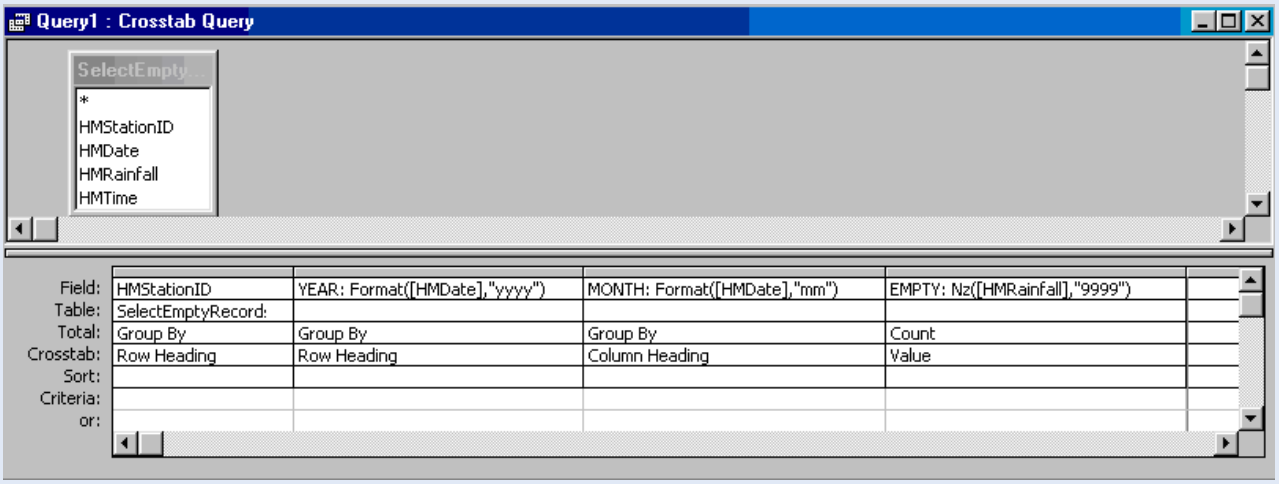
EMPTY:Nz([HMRainfall],"9999")

Step 5: use the "Totals" button to activate the "Group By" functions: in this case "Group By" for the "HMStationID", "YEAR", and "MONTH" fields, and "Count" for the "EMPTY" field.

Step 6: Change into a "Cross-tab Query" by selecting this query type from the "Query" menu.

Step 7: For each query field, select the appropriate "Cross-tab" settings, as follows:

"HMStationID" "Row Heading"
"YEAR" "Row Heading"
"MONTH" "Column Heading"
"EMPTRY" "Value"
The final query is shown in the below figure.



Step 8: run the query. A sample of the results is presented below.

It shows a comprehensive inventory of the numbers of empty records per month, per year, and per station. For example, the time series for station 70001 contains empty years for 1983, 1984, 1985, and part of 1986. Making an inventory without taking into account empty records would thus have created a false impression of a long time series. This is even worse for station 70015, which contain empty records from June 1980 to December 1986.

This systematic absence of complete data months points to data processing and entry difficulties. The data manager is advised to check the original paper recordings for the missing months.

By contrast, the erratic and irregular occurrence of mission values for station 70067 could point to instrument failure, extreme events, or sickness of the operator.

HMStationID	YEAR	01	02	03	04	05	06	07	08	09	10	11	12
70001	1983					31	30	31	31	30	31	30	31
70001	1984	31	29	31	30	31	30	31	31	30	31	30	31
70001	1985	31	28	31	30	31	30	31	31	30	31	30	31
70001	1986	31	28	31			30	31					
70001	1987						30						
70001	1988												31
70001	1991	31										30	
70001	1993									30	31		
70001	1994	31											
70015	1977			1									11
70015	1980						30	31	31	30	31	30	31
70015	1981		28	31	30	31	30	31	31	30	31	30	31
70015	1982	31	28	31	30	31	30	31	31	30	31	30	31
70015	1983	31	28	31	30	31	30	31	31	30	31	30	31
70015	1984	31	29	31	30	31	30	31	31	30	31	30	31
70015	1985	31	28	31	30	31	30	31	31	30	31	30	31
70015	1986	31	28	31	30	31	30	31	31	30	31	30	31
70037	1988	31	29	31	30	31	30	31					
70037	1989								31			30	
70037	1990										31		31
70043	1988	31	29	31	30	31							
70043	1989			15									
70045	1988									30			
70045	1990						30						
70048	1988	31	29	31	30								
70048	1989	31	28	31	30	31				2	31	30	31
70048	1990	31	28										
70048	1992								1				
70053	1988	31	29	31	30								
70053	1989							31	1				
70067	2000				14								
70067	2001	15				14							

Identifying Zero Values

A similar query as above can identify zero values. An example of the output of such query is presented below. It shows periods of several months on a row without rainfall, which is highly unlikely for the specific area. It is more probable that we have identified a structural data entry error in which the operators have entered zero values for months without data. In this case it is strongly advised to check the original records to establish if this prolonged dry spell has actually occurred.

Hydromet_ID	YEAR	01	02	03	04	05	06	07	08	09	10	11	12
8634002	1987	30	24	27	24	18	23	29	28	28	31	30	31
8634002	1989	31	27	25	23	22	27	18	28	21	31	30	31
8731015	1986	31	28	31	30	31	30	31	31	22	21	20	30
8733019	1977	23	23	20	17	14	16	13	11	14	14	16	31
8733019	1978	31	19	20	16	16	15	19	12	23	17	29	25
8733019	1979	25	21	23	30	31	30	24	21	30	31	17	31
8733019	1981	31	28	31	30	31	30	24	20	30	31	30	31
8733021	1977	26	28	31	30	31	30	31	31	18	14	29	31
8733021	1978	30	19	23	13	15	16	19	11	30	31	30	23
8733023	1977	21	25	24	20	13	22	30	15	19	13	29	31
8733023	1978	29	19	23	15	14	14	17	7	30	31	30	23
8830005	1977	31	28	25	22	16	22	19	27	24	28	16	31
8830006	1977	26	21	15	16	14	13	21	12	7	12	10	20
8830006	1978	30	16	8	11	18	23	25	14	17	8	15	18
8830006	1979	22	28	31	30	31	30	31	31	30	31	30	31
8830006	1983	31	28	31	30	31	30	31	31	30	31	16	21
8830006	1984	30	20	21	11	18	19	18	17	18	7	3	23

QC-5: Identifying Data Gaps

The need for complete records

The usefulness of a time-series data set depends to a large extent on its length and completeness. A typical hydro-meteorological analysis to identify drought and flood magnitudes and frequencies requires long and complete data series. This in particular because missing data often represents extreme events, which are difficult to monitor. Hence, data gaps greatly reduce the value of the dataset. It is therefore important to assess the completeness of the record set and the nature of the data gaps. Individual missing values may indicate the occurrence of an extreme event, while an entire missing month could point to difficulties in data processing.

How to identify data gaps?

One can identify data gaps by plotting the records, or by querying the data set using the "Group By" and "Count" functions.

Analysis 1: count number of records per year

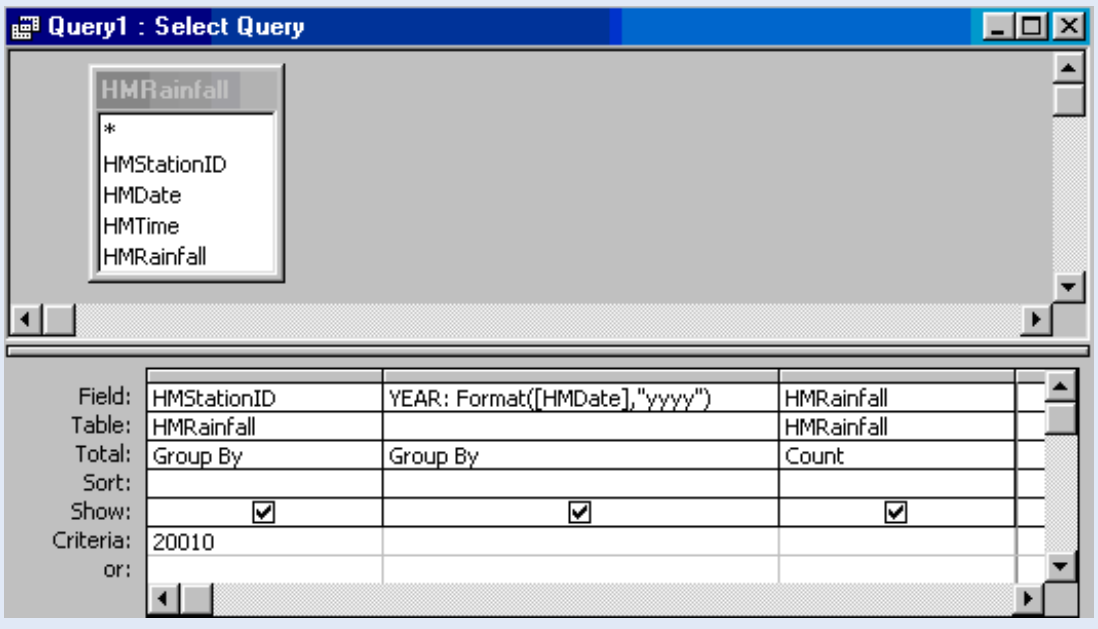
Step 1: Create a new query in design view; add the appropriate data table; choose "Select Query"; drag all relevant table fields to the query window (in this case "HMStationID", "HMDate", and "HMRainfall"). Type the appropriate station code in the criteria field.

Step 2: Use the "format" function to identify individual time periods, in this case years. The "format" function has the following syntax:

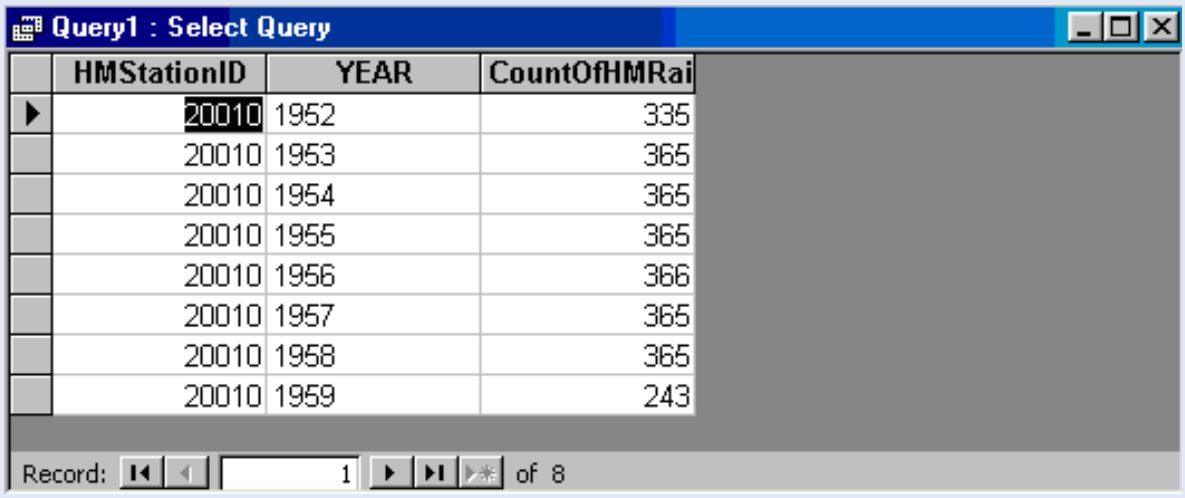
Format([table field],"code identifying time period, in this case "yyyy" for a year presented in 4 digits"). An example of the proper syntax is given below.

YEAR: Format([HMDate],"yyyy")

Step 3: Click the "Totals" button to activate the "Group By" functions; select "Group By" for the "HMStationID" and "format function" fields; select "Count" for the data field, in this case "HMRainfall". An example of such query is shown below.



Step 4: Run the query. The follow window shows the results.



The query indicates that the years 1952 and 1959 are incomplete. However, it is not possible to distill the nature of the data gap from these results. More information may be obtained by assessing the completeness of the individual months.

Analysis 2: count number of records per month

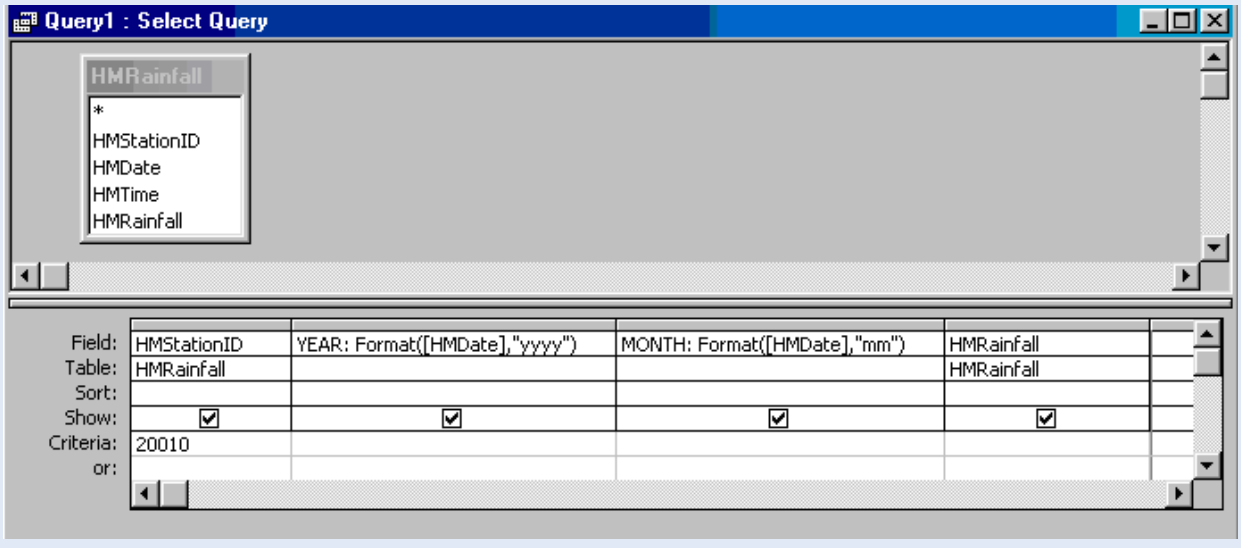
Step 1: Create a new query in design view; add the appropriate data table; drag the station-ID and data fields to the query (in this case "HMStationID" and "HMRainfall"), as well as two instances of the date field (in this case the "HMDate" field).

Step 2: Use the "Format" function to identify individual months and years. To this end, add the following codes to the date fields:

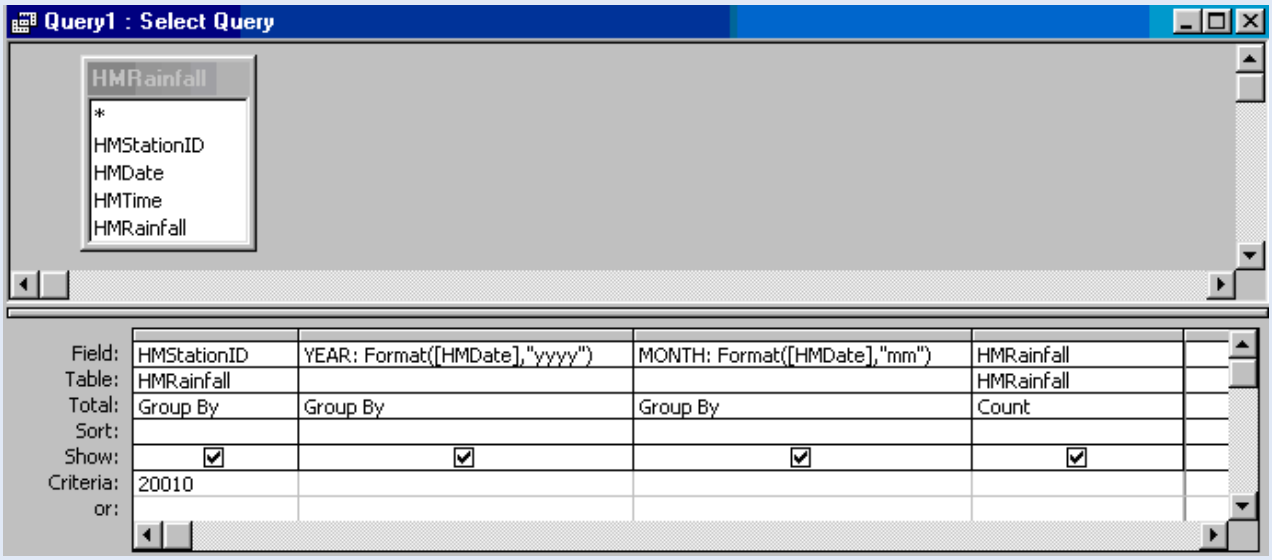
YEAR: Format([HMDate],"yyyy") : to identify the respective years in 4 digits

MONTH: Format([HMDate],"mm") : to identify the respective months in 2 digits

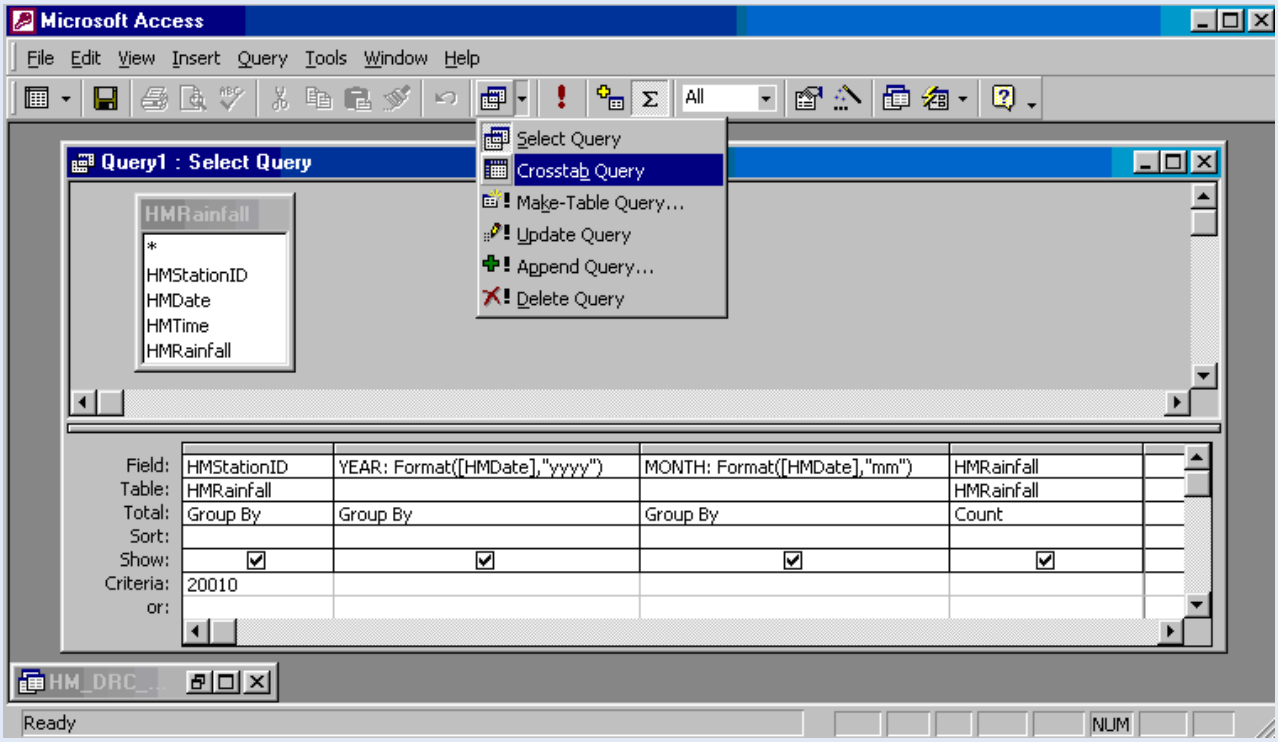
This is illustrated below.



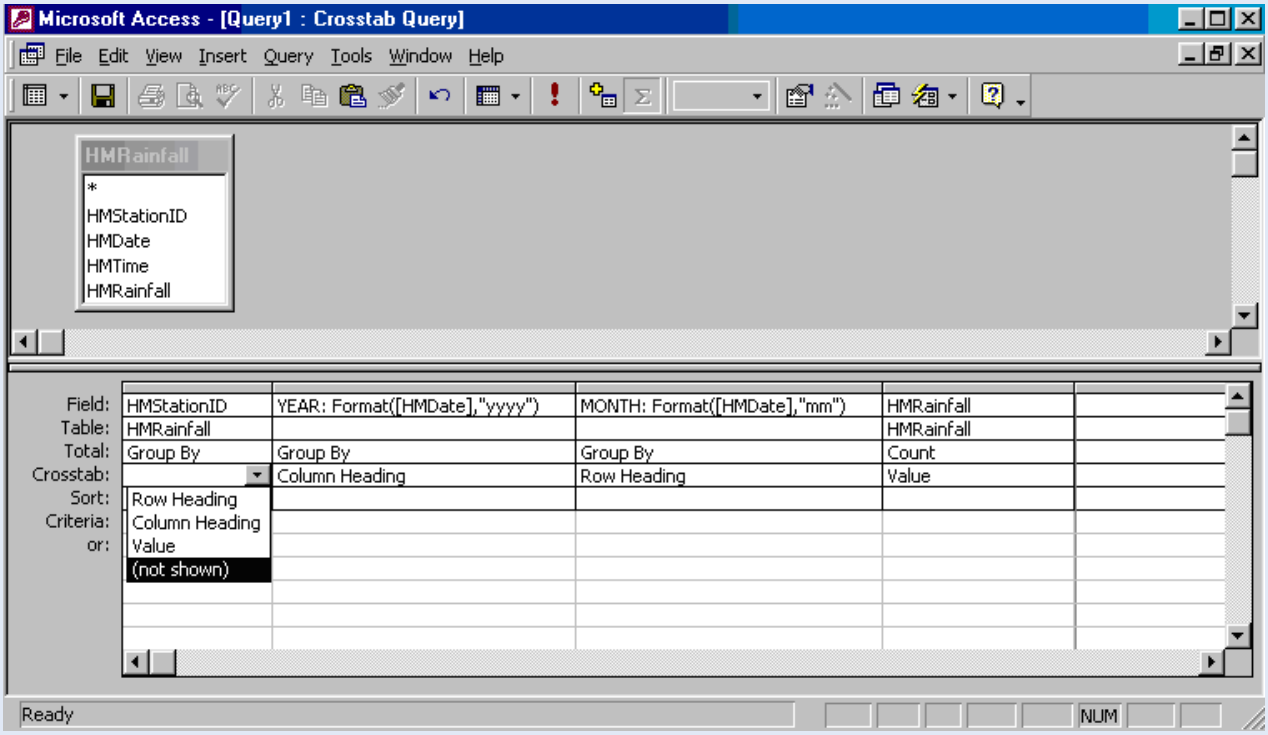
Step 3: Use the "Totals" button to set the "Group By" functions; in this case "Group By" for the Station-ID, "YEAR", and "MONTH" query fields, and "Count" for the "HMRainfall" data field. This is illustrated below.



Step 4: Change into a "Crosstab Query" by selecting this query type from the "Query" menu, as indicated below.



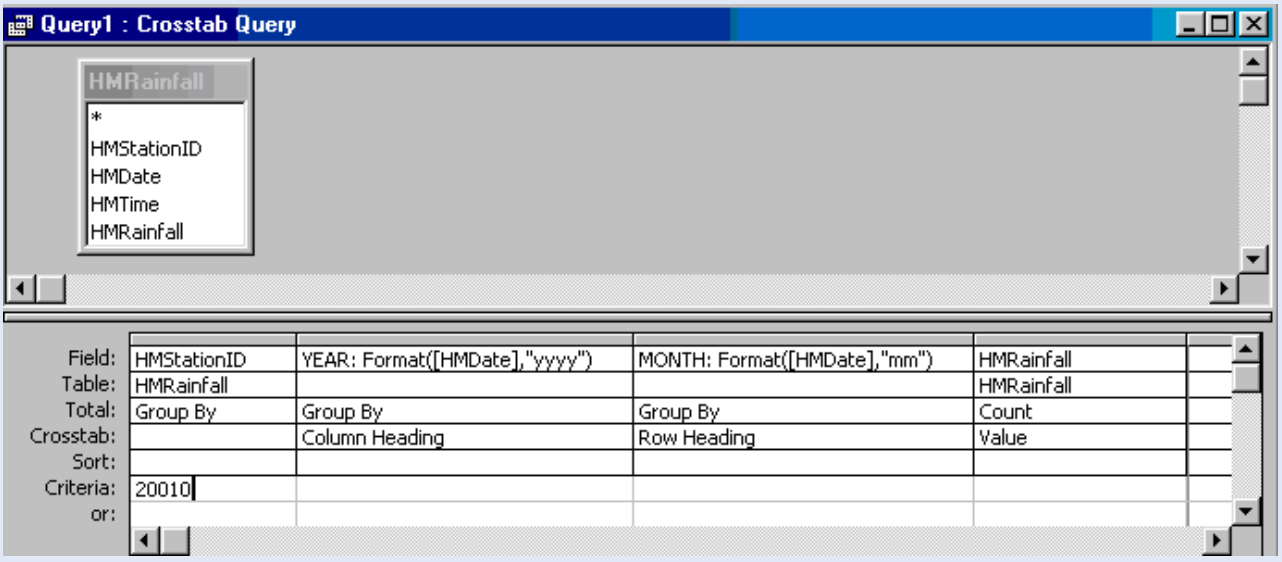
Step 5: For each query field, select the appropriate "Crosstab" settings; to this end, place the cursor in the "Crosstab" field and activate the drop down list, as shown in the below illustration.



Step 6: select the appropriate setting, as follows:

"HMStationID"	"(not shown)"
"YEAR"	"Column Heading"
"MONTH"	"Row Heading"
"HMRainfall"	"Value"

The final query is shown in the following figure



Step 7: Run the query. The results are presented below.

Query1 : Crosstab Query

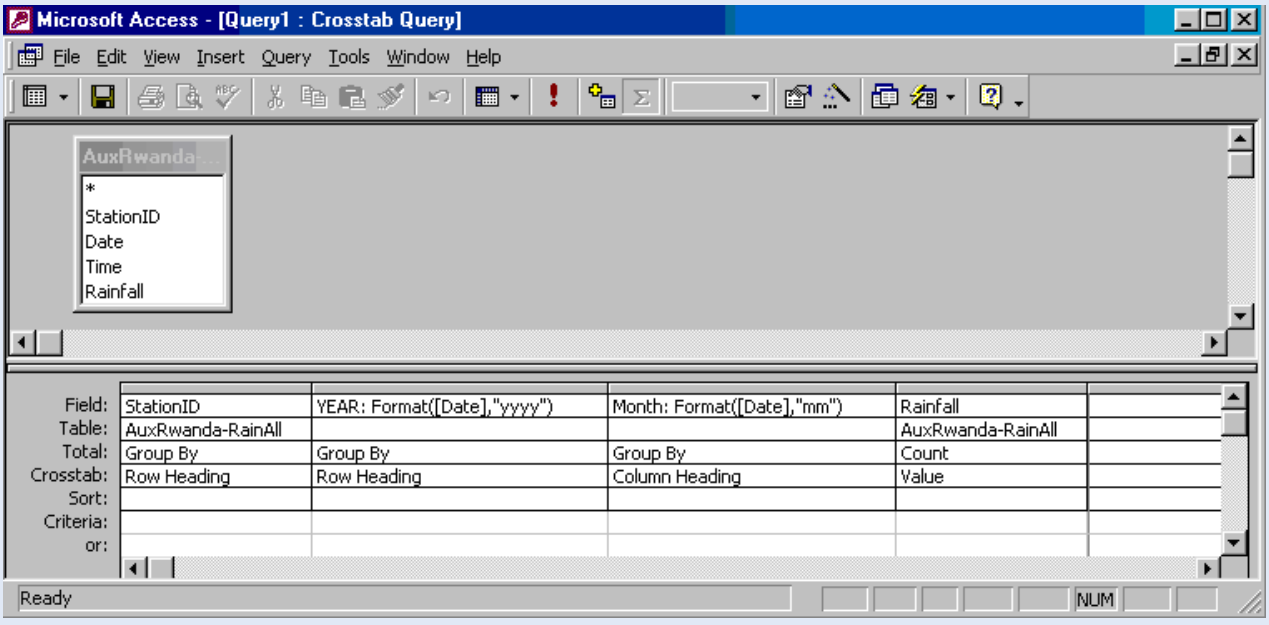
MONTH	1952	1953	1954	1955	1956	1957	1958	1959
01	31	31	31	31	31	31	31	31
02	29	28	28	28	29	28	28	28
03	31	31	31	31	31	31	31	31
04	30	30	30	30	30	30	30	30
05	31	31	31	31	31	31	31	31
06	30	30	30	30	30	30	30	30
07	31	31	31	31	31	31	31	31
08		31	31	31	31	31	31	31
09	30	30	30	30	30	30	30	
10	31	31	31	31	31	31	31	
11	30	30	30	30	30	30	30	
12	31	31	31	31	31	31	31	

Record: 8 of 12

This particular example shows a complete missing month for August 1952, which could indicate a data processing error. In this case it is advised to check the original paper records if this missing month could still be located.

Analysis 3: count number of records per month for all stations

The above query could be slightly modified to get a full picture of the data gaps for all stations in the record set. To this end, make changes, as shown below, to the query described in Step 6. Row Headings have now been set for "Station-ID" and "YEAR" fields, while Column Heading is set to the "Month" field. As no criterion is entered for the station ID, all stations are taken into account.



Running this query results in the following output.

Microsoft Access - [Query1 : Crosstab Query]

File Edit View Insert Format Records Tools Window Help

StationID	YEAR	01	02	03	04	05	06	07	08	09	10	11	12
70009	1995	31	28	31	30	31	30	31	31	30	31	30	31
70009	1996	31	29	31	30	31	30	31	31	30	31	30	31
70009	1997	31	28	31	30	31	30	31	31	30	31	30	31
70009	1998	31	28	31	30	31	30	31	31	30	31	30	31
70009	1999	31	28	31	30	31	30	31	31	30	31	30	31
70009	2000	31	29	31	30	31	30	31	31	30	31	30	31
70009	2001	31	28	31	30	31	30	31	31	30	31	30	31
70009	2002	31	28	31	30	31	30	31	31	30	31	30	31
70013	1998	31	28	31	30	31		31	31	30	31	30	31
70013	1999								31	30	31	30	31
70035	1996		29	31					31	30		30	31
70035	1997		28	31			30	31	31	30			31
70035	1998	31	28	31	30	31	30	31	31	30	31	30	
70035	1999							31	31	30	31	30	31
70078	1996												31
70078	1997				30	31							
70078	1998											30	31
70078	1999			31		31	30	31	31	30		30	31
70085	1997	31	28	31		31	30	31					
70085	1998	31	28	32	30	31	30	31	31	30	31	30	31
70085	1999	31	28	31	30	31	30	31	31	30	31	30	32
70124	1996							31	31	30	31	30	32
70124	1997	31	28	31	30	31	30	31	31	30	31	30	32
70166	1998		28	31	30	31	30	31	31	30	31	30	
70166	1999	31	28	31	30	31	30	31	31	30	31	30	31

Record: 1 of 25

Datasheet View NUM

It shows a comprehensive inventory of the number of records per month for all stations. For example, the time series for station 70009 in the above example is complete and could thus represent a valuable data set. By contrast, station 70035 experiences numerous missing months and is therefore of limited value.

QC-6: Validity Check

What is a reasonable data range?

The data within the reasonable range represent the physical element's expected value. The range's minimum and maximum bounds are a function of the local environmental conditions, and determined by experience. They can vary per location and season. For example, daily rainfall on the Lake plateau typically ranges between 0 and 100 mm per day. An event exceeding 100 mm is unlikely.

Data outside the reasonable range is not automatically rejected, but flagged as questionable. Its validity has to be checked before being used in hydro-meteorological analysis, or being discarded as being incorrect.

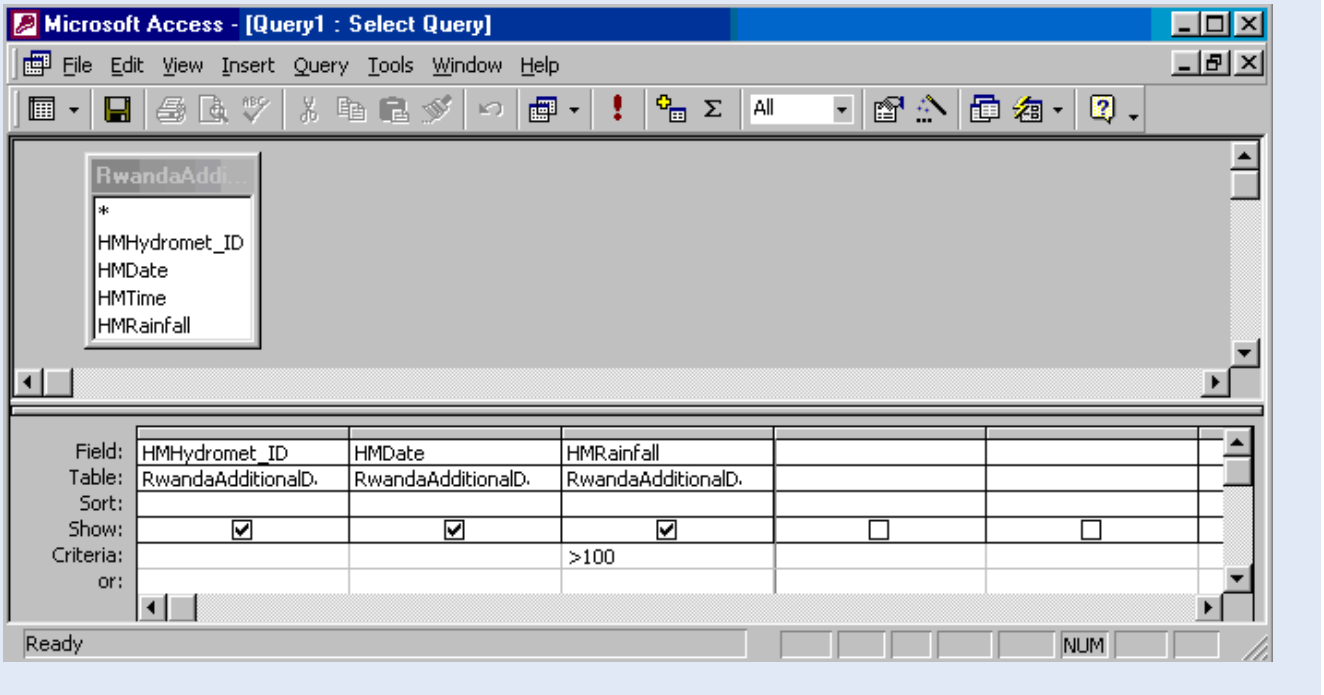
How to check for data values outside the reasonable range

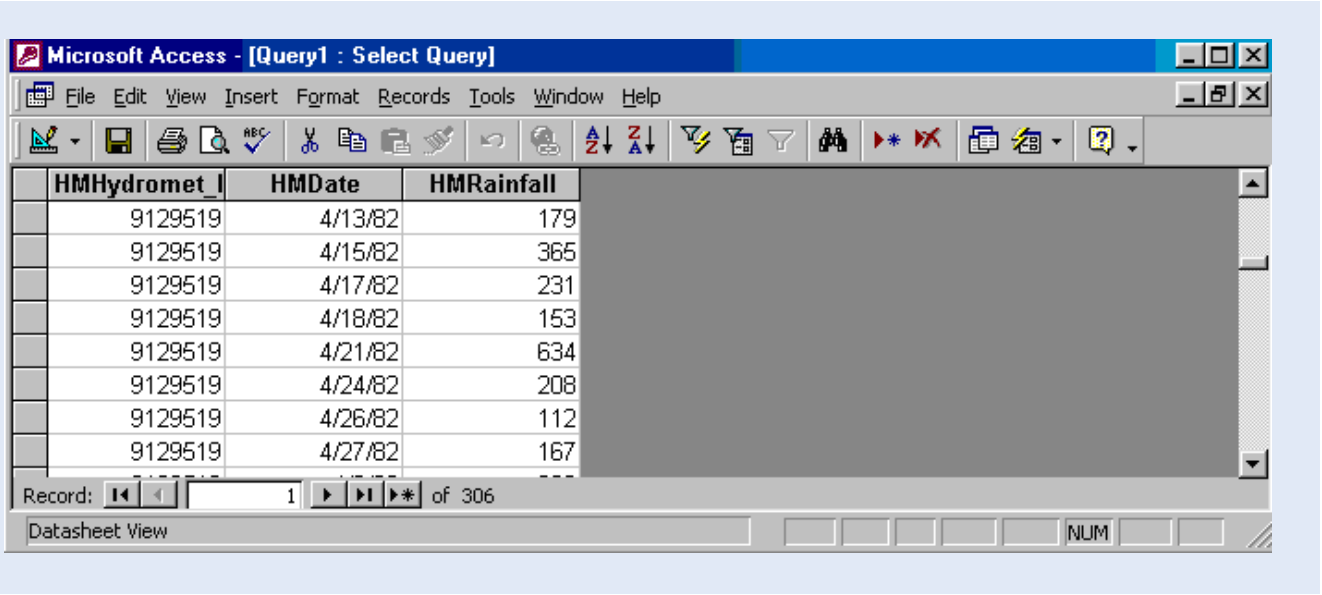
By a simple "select query" setting an appropriate criterion.

Step 1: Start a new 'Select Query' in design view; add the table containing the data records.

Step 2: Drag the "Station-ID", "Date", and "Value" field to the query window; in this particular case these are: "HMHydromet_ID", "HMDate", and "HMRainfall".

Step 3: Set an appropriate criterion in the "Data" field, in this case ">100" [mm] in the "HMRainfall" query field; run the query. Both query and query results are shown below.



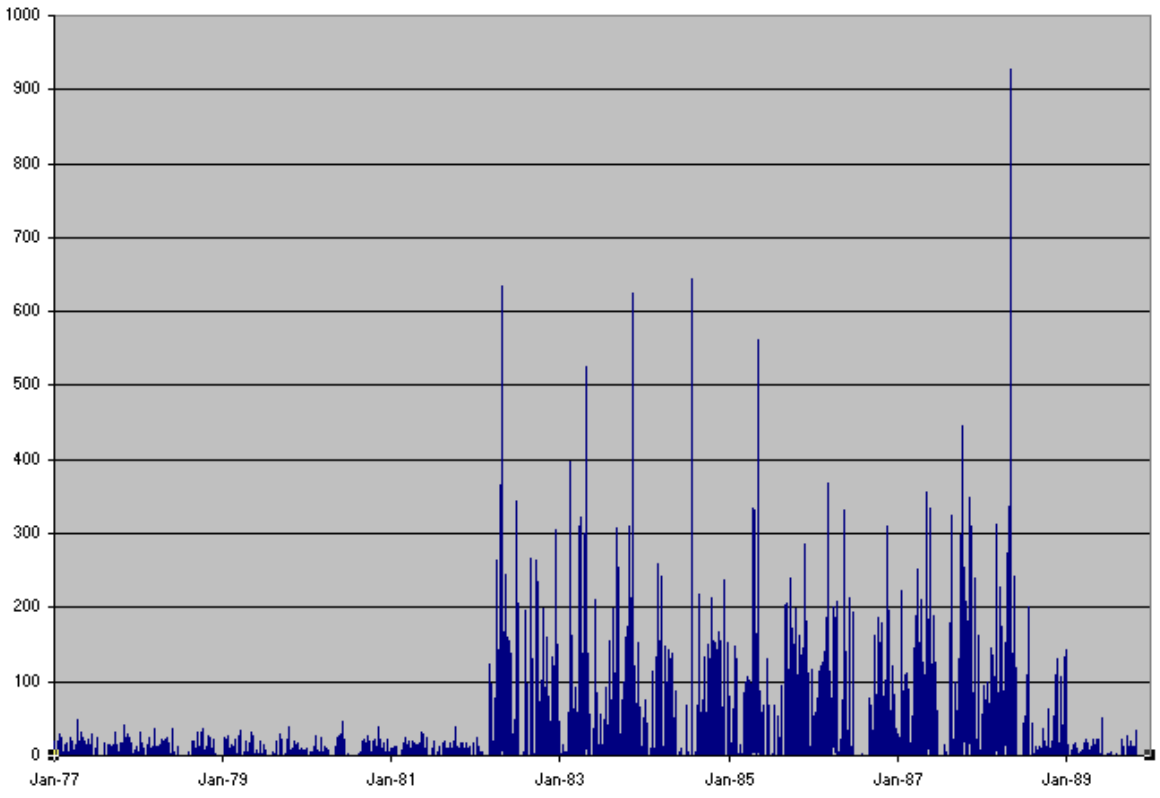


In this particular situation, the record set contains 306 instances of daily rainfall exceeding 100 mm. As can be seen in the illustration above, a group of these questionable events all take place in the same month for the same station (in this case Station 9129519 for the month of April 1982). This could indicate a data entry error, most likely by an erroneous setting of the comma.

How to check for outliers within the reasonable range

By plotting the complete time series for one station in a graph.

This has been done for station 9129519, which was indicated above as likely having erroneous records. The "Chart Wizard" in MS Excel was used. The results are shown in the following figure.



This figure clearly shows a systematic data error from March 1982 to December 1988.

How to correct possible data errors

The only way to correct possible data errors is by comparing the computerized values with the original paper recordings. This is a tedious but indispensable undertaking. Proven data errors should be corrected or, if not possible, replaced by the missing value code.

QC-7: Plotting Time Series

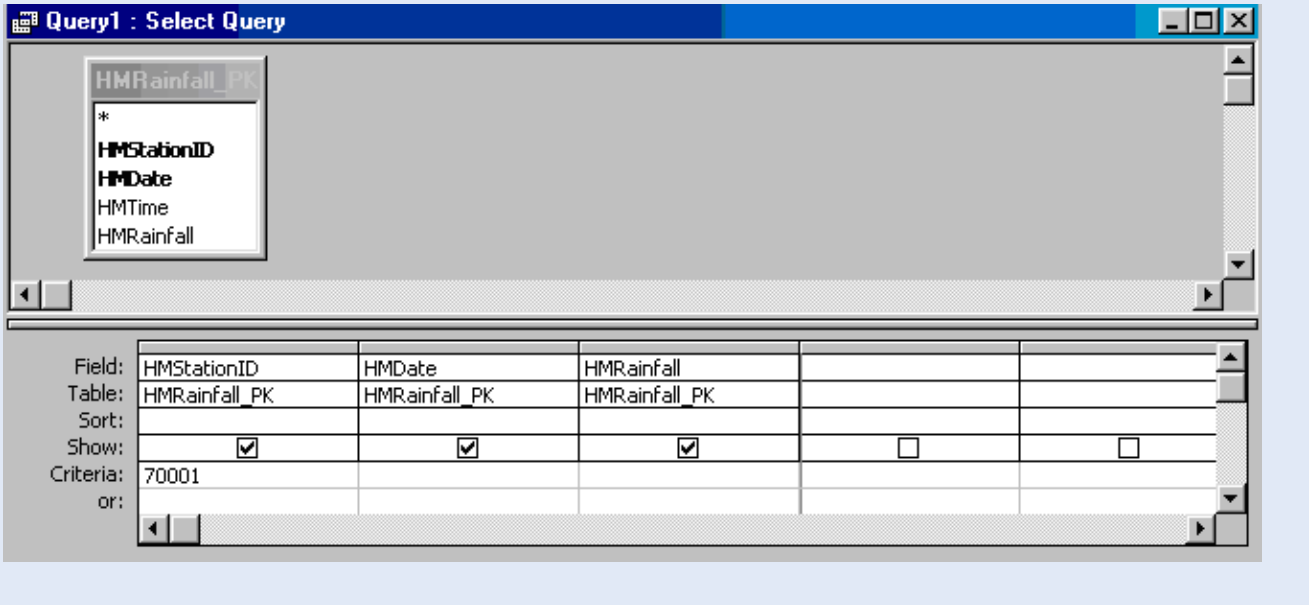
Why plotting time series?

Plotting a time series in a graph has proven an excellent and straightforward means to check the completeness and consistency of a record set. Spikes, data gaps, and systematic data entry errors are easily identified.

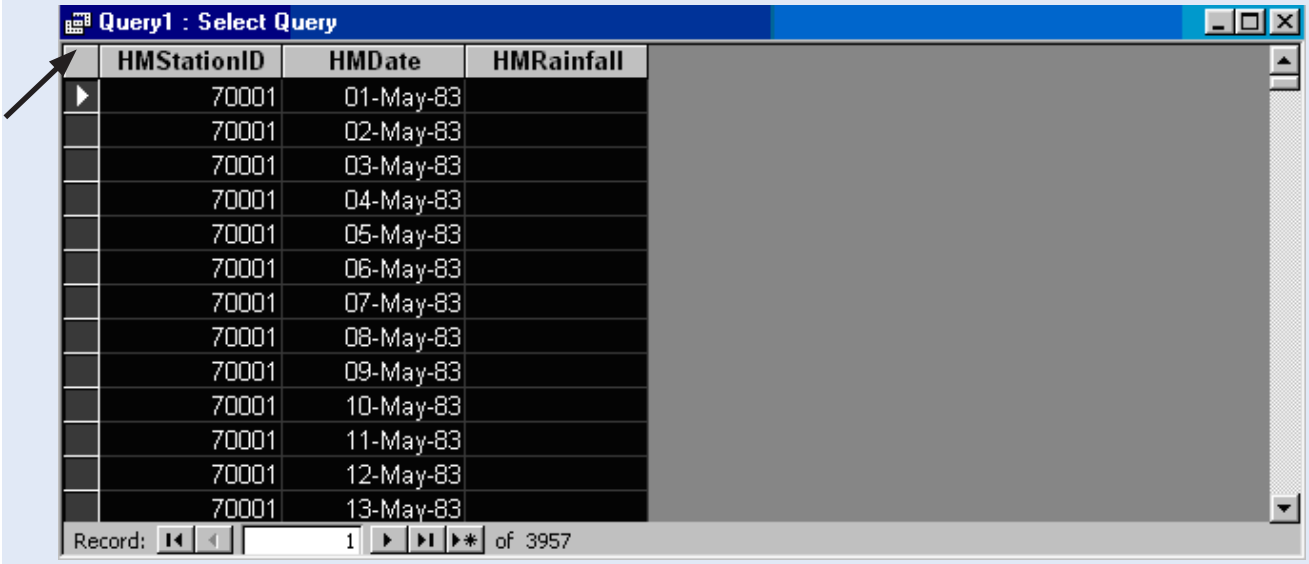
The combined use of MS Access and MS Excel makes time series plotting a simple and quick exercise.

How to make a time series plot?

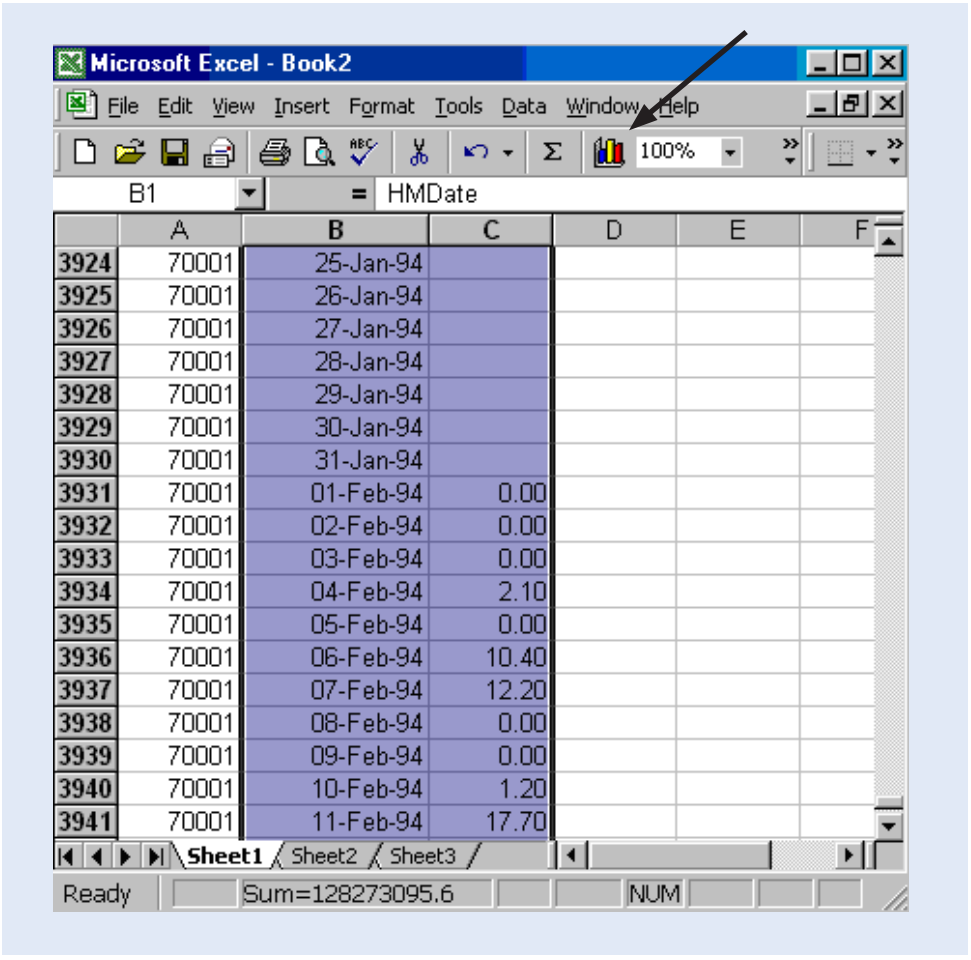
Step 1: in MS Access, create a new 'Select Query' in design view; add the relevant data table; drag the "Station-ID", "Date", and "Value" fields to the query window; set the appropriate criterion to extract the data for the requested station; this query is shown in the figure below.



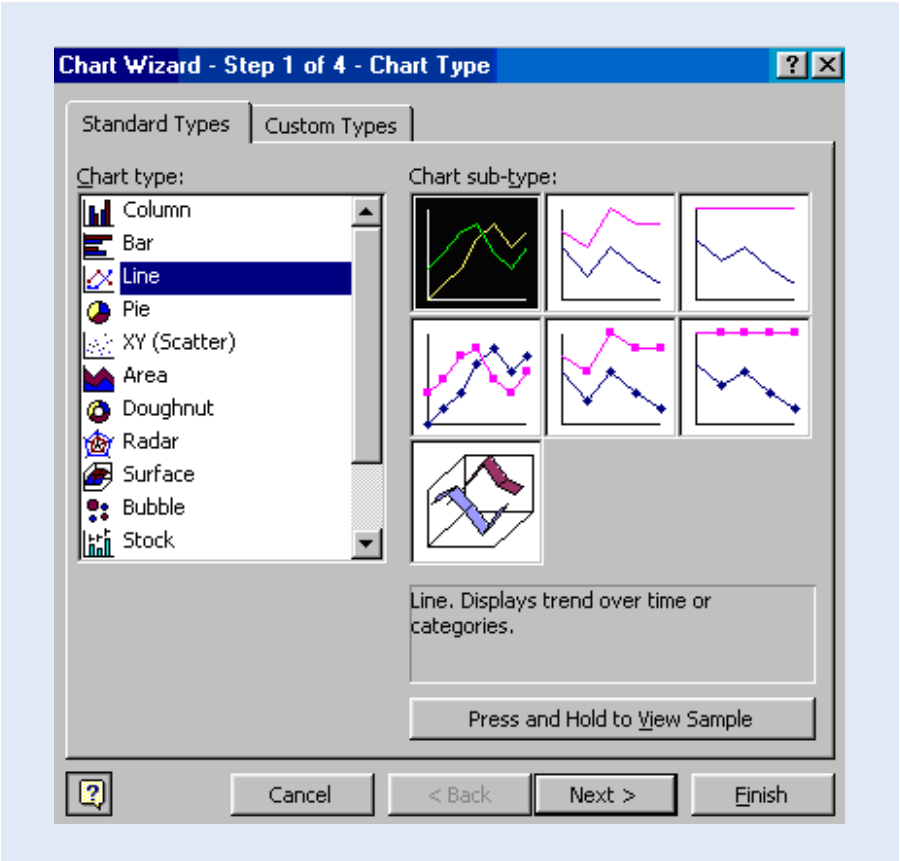
Step 2: run the query; in the resulting table, click the top-left square (indicated in the below figure) to select the complete data set; Click "Copy" from the Edit menu;



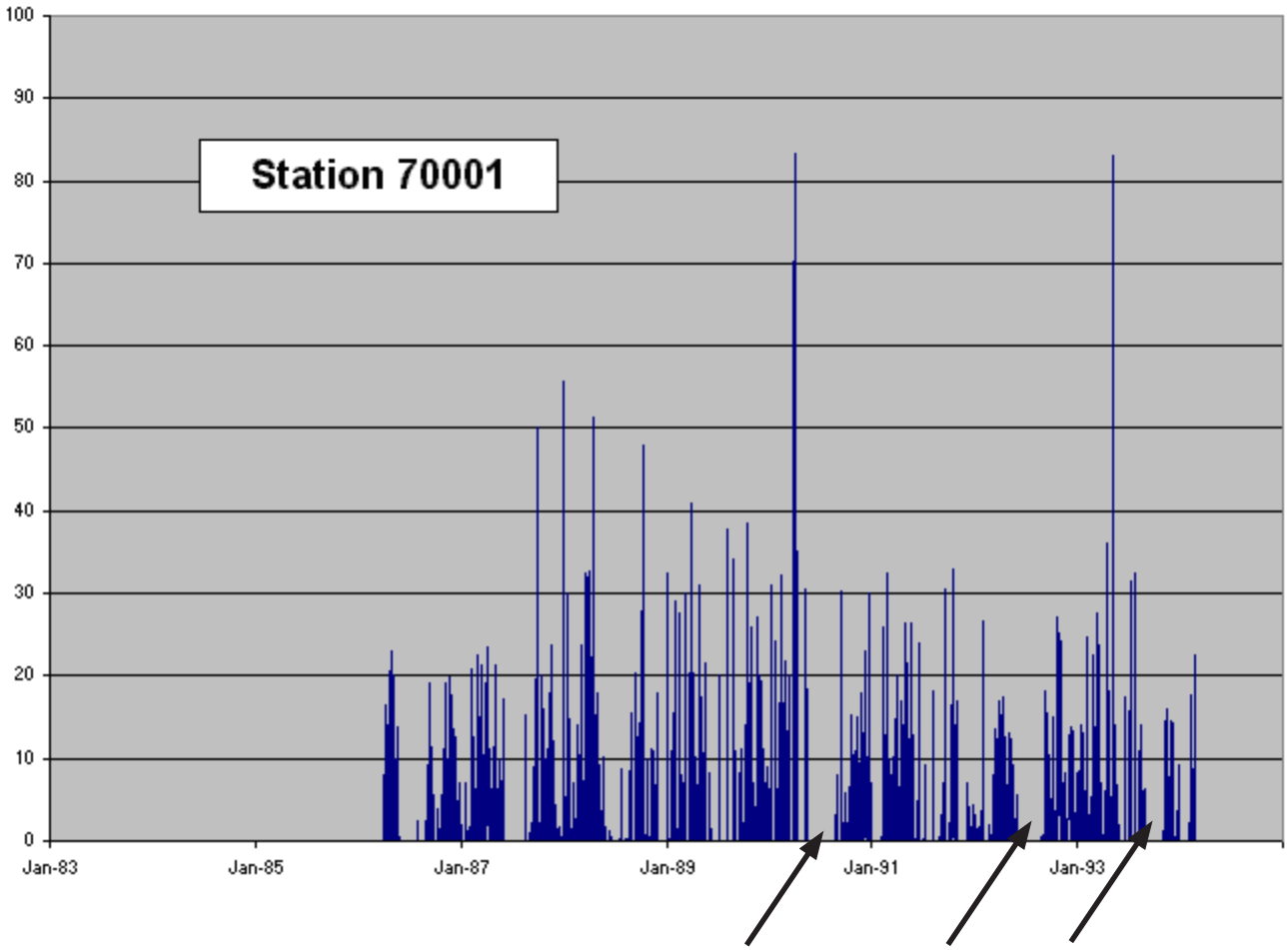
Step 3: open a new 'workbook" in MS Excel; select 'Paste' from the Edit menu; the time series for the selected station is now transferred to Excel; select all records in the 'date' and 'value' columns (by holding the 'Shift' key and navigating downwards); click the 'Chart Wizard' icon, as shown below.



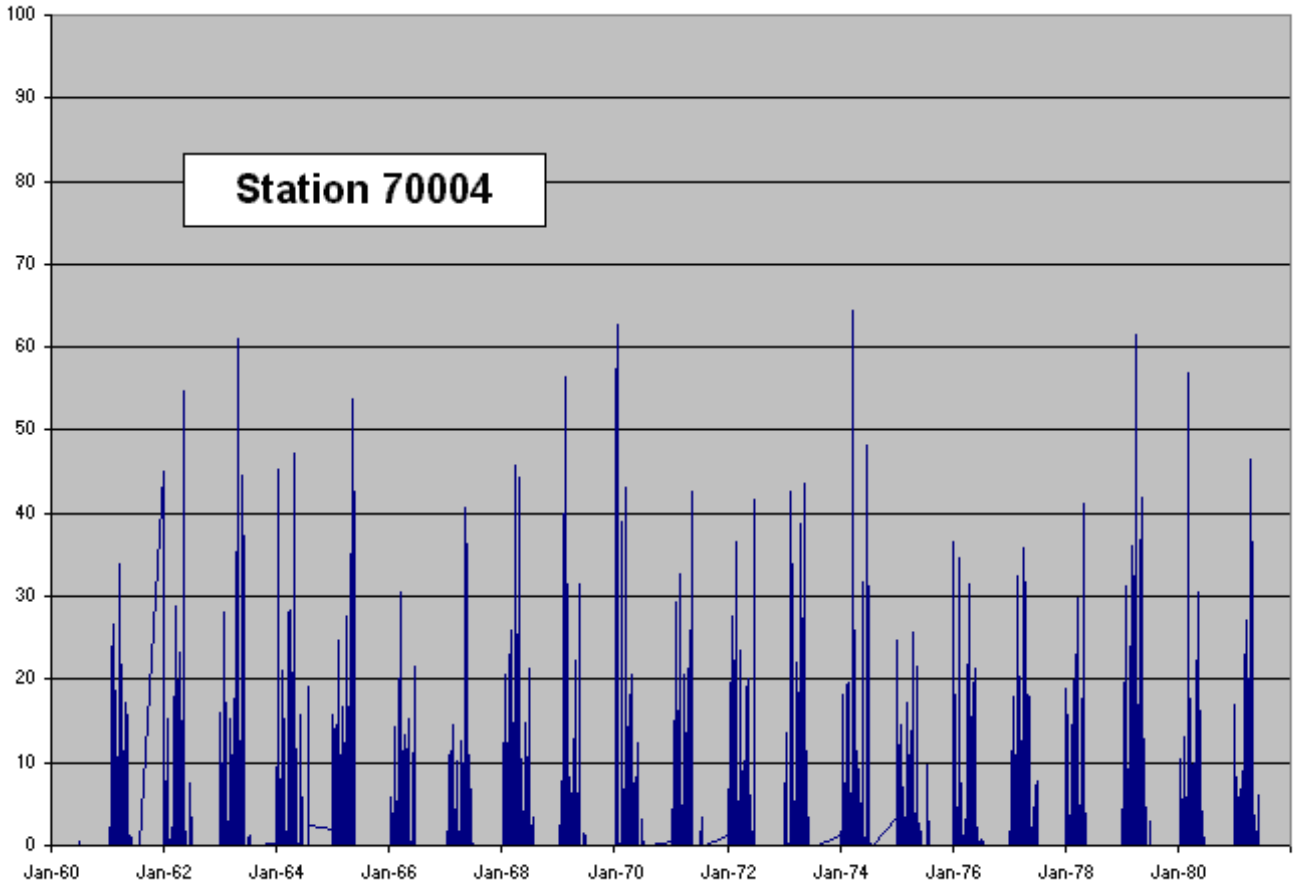
Step 4: select "Line"; follow the Chart Wizard instructions.



The below figure presents an example of a time series graph. In this particular case it shows the daily rainfall record for station 70001. The range of rainfall values is plausible. Nevertheless, it is advised to check the original paper records for the two outliers (rainfall events above 80 mm/day) as they have a significant effect on the record set's statistical parameters. The plot also shows that the years 1983 to 1986 contain empty records. There is no reason to keep these records in the data set, as they give a false impression of a long data series and only take up hard disk space. Several other blocks of empty records are seen, for example in 1990, 1992, and 1993, as indicated with arrows. These are most likely complete missing months, and it is advised to check the original paper records for these data. Their inclusion would certainly enhance the value of the data set.



A similar graph has been prepared for station 70004 (presented in the figure below). This plot clearly implies a systematic error. The months August to December consistently contain empty records. A quick comparison with the nearby station 70001 indicates that, in this particular region, rainfall occurs throughout the year. The missing August-December periods therefore represent a serious data gap, making the record set for station 70004 effectively useless. However, the consistency of these data gaps points to a systematic data entry error, and it is advised to check the original paper recordings for the missing months.


Station 70004
100
90
80
70
60
50
40
30
20
10
0
Jan-60
Jan-62
Jan-64
Jan-66
Jan-68
Jan-70
Jan-72
Jan-74
Jan-76
Jan-78
Jan-80

QC-8: Identical Monthly and Annual Totals

Why do identical monthly and annual totals point to possible data errors?

By their nature hydro-meteorological parameters are subject to spatial and temporal variability. Although climatic events can have similar characteristics for large parts of a country or region, the incidence of identical values is rare, even for neighboring stations. The probability of identical monthly or annual totals is even lower, and their occurrence therefore suggests a structural data error.

How to identify identical monthly and annual sums?

Through a select query using the aggregate and format functions.

Step 1: create a new select query in design view; add the appropriate data table.

Step 2: drag the relevant fields into the query window (in this particular case "HMStationID", 2 instances of "HMDate", and "HMRainfall").

Step 3: use the "Format" function to identify discrete time periods (in this case both months and years). The format function has the following syntax:

YEAR: Format([HMDate],"yyyy") : to identify the respective years in 4 digits

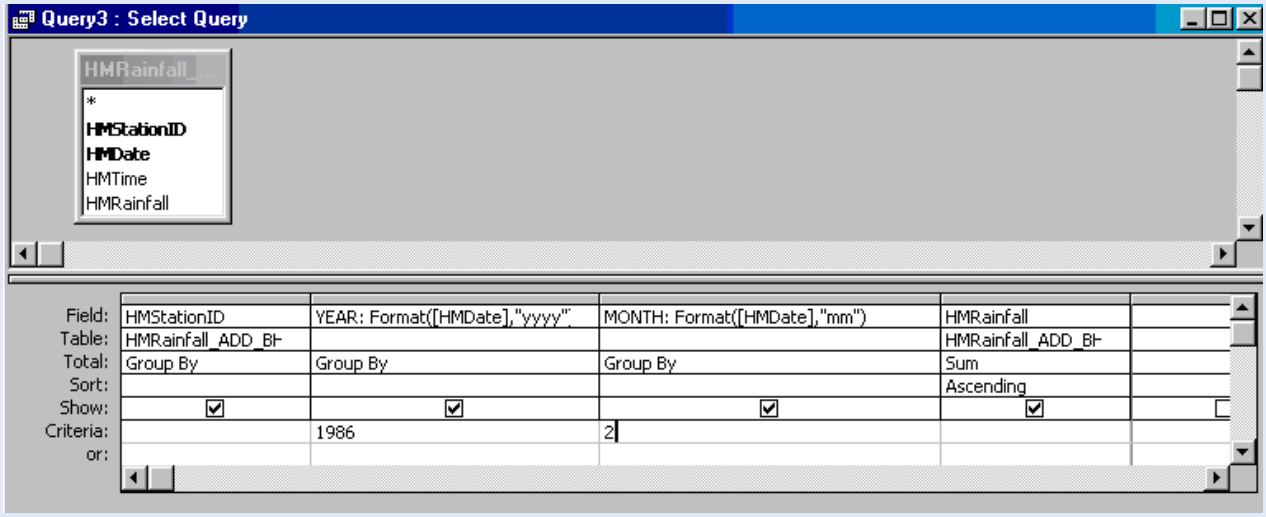
MONTH: Format([HMDate],"mm") : to identify the respective months in 2 digits

Step 4: For each query field, set the appropriate aggregate function, as follows:

"HMStationID" :Group By
"YEAR: Format([HMDate],"yyyy")" :Group By
"MONTH: Format([HMDate],"mm")' :Group By
"HMRainfall" :Sum

Step 5: Set the combination month-year for which to check for identical totals.

An example of the final query is presented in the below figure.



The query found identical monthly totals for stations 70054 and 70055, and for stations 70045 and 70046, as shown below. As stated above, this is unlikely and it is advised to check the original records.

Query3 : Select Query

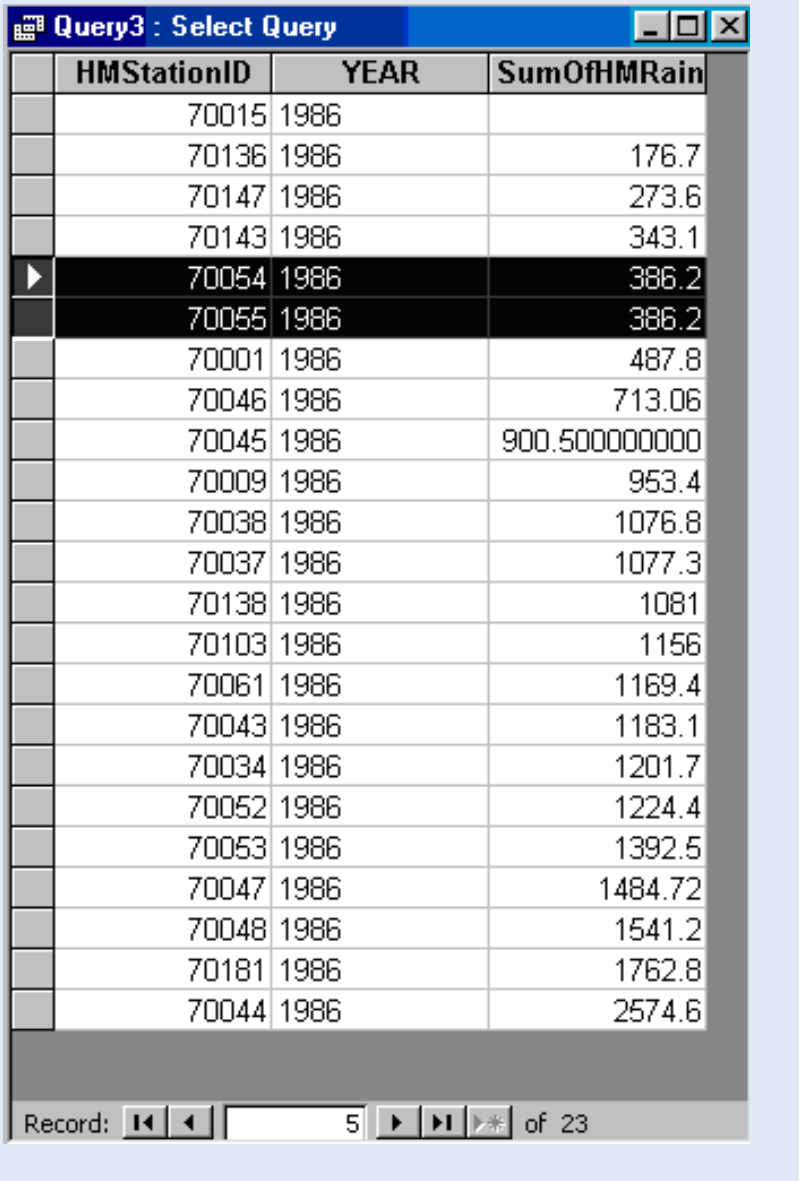
HMStationID	YEAR	MONTH	SumOfHMRain
70143	1986	02	
70015	1986	02	
70001	1986	02	
70055	1986	02	31.7
70054	1986	02	31.7
70136	1986	02	36
70053	1986	02	70.1
70138	1986	02	79.2
70147	1986	02	81
70061	1986	02	84.5
70052	1986	02	85.5
70038	1986	02	91.3
70181	1986	02	91.5
70103	1986	02	95.3
70048	1986	02	97.6
70009	1986	02	103.6
70034	1986	02	114.5
70044	1986	02	118.2
70043	1986	02	127.3
70047	1986	02	128.1
70045	1986	02	137
70046	1986	02	137
70037	1986	02	166

Record: 4 of 23

The above query should be repeated for each month in each data year. However, a short cut is to check the annual sums. This is accomplished through the same query as above, but without including the MONTH field. The result of running this query is shown below.

Station 70054 and 70055 show identical yearly totals, suggesting that the same record set has been entered twice for two different stations. By contrast, station 70045 and 70046 have different annual accumulated values, pointing to a less structural data error.

The absence of identical yearly sums does, of course, not mean that individual months have not been entered twice for different stations. The user is still advised to check each month in each data year. However, this is a rather quick exercise.



How to correct duplicated data?

By checking the original paper recordings. This is a tedious and time consuming, but unavoidable exercise.

QC-9: Naming Convention for Data Tables and Files

Why is there a need for a naming convention?

Database development is a continuing process. New records will be appended to existing data tables, data will be reviewed and corrected, sensors added, and stations will be established and closed.

It is necessary to develop a naming system to keep track of the modifications in the data files and tables. Failure to establish and follow such system could lead to confusion, loss of data, or duplication of work.

Proposed Naming Convention

The following syntax is proposed:

DATANAME_CODE1-CODE2_NAMEOPERATOR_DATE

In which:

DATANAME is the original name of the data table or file, e.g. "HMRwand0".

CODE depends on the action performed:

ADD	data added to fill gaps
APP	new records appended
COR	data corrected
DEL	data or table deleted
FIN	final data set
NWT	new table added
ORG	original data file before modification
PK	primary key added
REC	double and ambiguous records removed

Use multiple codes when multiple actions have been performed.

NAMEOPERATOR is the name the database manager having performed the action. It is proposed to use an abbreviation, e.g. JM for Jetty Masongole.

DATE is the date when the action was performed, using 3 characters to identify a month (JAN, FEB, MAR, etc.), and 2 digits each for date and year. For example 03Aug03 points to 3 August 2003. The date field is a principal element in following the history of the data set.

Examples

HMRainfall_COR-PK-REC_JM_03Aug03

HMRainfall table has been corrected, ambiguous or double records removed, and a primary key added; by Jetty Masongole, on 3 August 2003.

HMRainfall_APP_SAK_04Aug03

New records have been appended to the HMRainfall table; by Shaukat Ali Khan, on 4 August 2003.

HMRwand0_COR-NWT_BH_05August03

A new table has been added to the HMRwand0 file, while data have been corrected in this file; by Bart Hilhorst, on 5 August 2002.