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

The designations employed and the presentation of material throughout this book do not imply the expression of any opinion whatsoever on the part of the Food and Agriculture Organization (FAO) concerning the legal or development status of any country, territory, city, or area or of its authorities, or concerning the delimitations of its frontiers or boundaries.

The authors are responsible for the choice and the presentation of the facts contained in this book and for the opinions expressed therein, which are not necessarily those of FAO and do not commit the Organization.

### Table of Contents

Intro	duction	4
QC1	Quality Control Program for Time Series Data	7
QC2	Ambiguous Records	8
QC3	Duplicate Records	13
QC4	Systematically Identifying Empty Records	20
QC5	Identifying Data Gaps	25
QC6	Validity Check	31
QC7	Plotting Time Series	34
QC8	Identical Monthly and Annual Totals	39
QC9	Naming Convention for Data Tables and Files	42

### 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 timeseries 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_Corre	ctedMay03 : Tabl	e	
	HMStationID	HMDate	HMTime	HMRainfall
Þ	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
Re	cord: 🚺 🔳	103117 🕨 🕨 🕨	f 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.

Microsoft Acc	ess				
<u> </u>	Insert Query Tools	<u>W</u> indow <u>H</u> elp			
🖿 🖌  🖶	B. 🌮 🗶 🖬 🖡	l 🚿 🗠 🖪 • 🛛	🗜 🍄 Σ 🔤 ΑΙ	🔹 🖻 🏠 🛅 🚈 🕻	2 🗸
Query1	: Select Query				-
HM *	Rainfall_CorrectedM	lay03		<u> </u>	U.
HMC	StationID Date Fime				
HMF	Rainfall				н.
Field:	HMStationID	HMDate	HMRainfall		
Table: Sort:	HMRainfall_Correct(	HMRainfall_Correct(	HMRainfall_Correcte		
Show: Criteria:					
or:					
	•				
					<b>T</b>
Ready					

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

🖉 Microsoft Acc	ess			
Eile Edit View	Insert Query Tools	<u>W</u> indow <u>H</u> elp		
] 🖩 🖌  🖨	🖪 🖑 👗 🛍 🖬	l 🚿 🗠 🖶 •	🚦 🗣 Σ 🔤 ΑΙ	• 🖻 🏠 🛅 🚈 🛛 🗸
E Query1 :	Select Query			
* HMS HMD HMT		ay03		
Field:	HMStationID	HMDate	HMRainfall	
Table:	HMRainfall_Correct(	HMRainfall_Correcte	HMRainfall_Correcte	
	Group By 📃 💌	Group By	Count	
Sort: Show:	Group By			
Criteria:	Avg			
or:	Min			
	Max			
	Count StDev			
	Var 🔻			
Ready				NUM

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.

• • •	Q. 🕫 🕺 🖪 🛙	2 🚿 🖻 🖬 🕶	l ο <sub>Ξ</sub> Σ Al ·	🖻 🏠 🗖	<b>⁄a</b> • 🛛
Query1	Select Query			_	
* HMS HMT	Rainfall_Corrected StationID Date Fime Rainfall	4ay03			
Field:	HMStationID	HMDate	HMRainfall		
Table:	HMRainfall_Correcte	HMRainfall_Correct(	HMRainfall_Correcte		
Total: Sort:	Group By	Group By	Count		_
Show:					-
DHOTH			>1		
Criteria:					- 21
	•				

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

Query1 : Select Q	X 🖻 🖻 🚿		, 🏹 🚡 🔽 🕯	₩,   ▶* ₩   [	〕 ⁄a ∙ - □ ×
HMStationID	HMDate	CountOfHMRai			
10014	09-Jan-92	2			
10014	10-Jan-92	2			
10014	06-Mar-95	2			
10036	17-Dec-91	2			
10044	27-Oct-95	2			
10049	09-Sep-91	2			
10049	13-Oct-91	2			
10049	21-Nov-92	2			
10049	20-Apr-95	2			
10061	09-Nov-92	2			
10061	07-Jan-93	2			
10061	08-Jan-93	2			
10061	09-Jan-93	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2			
10061	<u>1</u> 0-Jan-93				•
ecord: 🚺 🔳	1 🕨 🕨	* of 87			

#### 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_CorrectedMa	ay03 : Table		
	Field Name	Data Type	Descriptio	n 🔺
Ϋ́	HMStationID	Number	Regional identification code	for Nile Basin Mete
Ŷ	HMDate	Date/Time	Date of the rainfall reading	
	HMTime	Date/Time	Time of the rainfall reading	
	HMRainfall	Number	Rainfall value in millimiters	
				<u> </u>
		Field Pro	perties	
	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.

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

#### 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.

нм	RHCode Date						
HM							
	Time RelativeHu ▼						
Field:	HMStationID	HMRHCode	HMDate	HMTime	HMRelativeHumidil	HMRelativeHumidity	F
	HMStationID HMRelativeHumidi	HMRHCode HMRelativeHumidil	HMDate HMRelativeHumidil	HMTime HMRelativeHumid	HMRelativeHumidil HMRelativeHumidil	HMRelativeHumidity	
Table:		HMRHCode HMRelativeHumidil	HMDate HMRelativeHumidi	HMTime HMRelativeHumid	HMRelativeHumidil HMRelativeHumidil	HMRelativeHumidity HMRelativeHumidity	

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

🗗 <u>E</u> ile <u>E</u> di	it <u>V</u> iew <u>I</u> nsert <u>Q</u> ue	ery <u>T</u> ools <u>W</u> indow	Help				_ 8
- 🖬	a 🛛 🖇	🖻 🖻 🚿 🗠	🖃 - 🚦 🐾 🛛	All 🔹 😭	🖄 🖬 ⁄a 🛛 🕻	Q •	
HMS HMR HMD							
	telativeHu						Þ
	elativeHu					1	
HMR	elativeHu	HMRHCode	HMDate	HMTime	HMRelativeHumidi	HMRelativeHumidity	
HMR Field: Table:	HMStationID HMRelativeHumidi	HMRelativeHumidil	HMRelativeHumidil	HMRelativeHumid	HMRelativeHumidil	HMRelativeHumidity	
Field: Table: Total:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	= =				
Field: Table: Total: Sort:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Group By	
Field: Table: Total: Sort: Show:	HMStationID HMRelativeHumidi Group By Sum	HMRelativeHumidil Group By	HMRelativeHumidil	HMRelativeHumid	HMRelativeHumidil	HMRelativeHumidity	
Field: Table: Total: Sort: Show:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Group By	
Field: Table: Total: Sort: Show: Criteria:	HMStationID HMRelativeHumidi Group By Sum Avg	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Group By	
Field: Table: Total: Sort: Show: Criteria:	HMStationID HMRelativeHumidi Group By Sum Avg Min	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Group By	

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.

I • 🔒	≝ La ♥   λ	B 🖬 💅 👳	💼 - 🚦 🔓 Σ	E All 👻 😭	🖄 🗇 🕭 📲 🖸	¥ -	
HMF	Relative						
	tationID						
	HCode						
HMD	ate						
HMTi	ime						
HMR	elativeHu						
	elativeHu						
	elativeHu						F
	elativeHu			1			•
Field:	elativeHu	HMRHCode	HMDate	HMTime	HMRelativeHumidi	HMRelativeHumidity	
Field:		HMRHCode HMRelativeHumidi	HMDate HMRelativeHumidil	HMTime HMRelativeHumid	HMRelativeHumidil HMRelativeHumidil	HMRelativeHumidity HMRelativeHumidity	
Field: Table: Total:	HMStationID						
Field: Table: Total: Sort:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Count	
Field: Table: Total: Sort: Show:	HMStationID HMRelativeHumidi	HMRelativeHumidil	HMRelativeHumidil	HMRelativeHumid	HMRelativeHumidil	HMRelativeHumidity Count	
Field: Table: Total: Sort: Show: Criteria:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Count	
Field: Table: Total: Sort: Show:	HMStationID HMRelativeHumidi Group By	HMRelativeHumidil Group By	HMRelativeHumidil Group By	HMRelativeHumid Group By	HMRelativeHumidil Group By	HMRelativeHumidity Count	

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

∃ <u>F</u> ile Edit ⊻iew Insert F <u>o</u> rmat	<u>R</u> ecords <u>T</u> ools <u>W</u> ind	ow <u>H</u> elp			_ 8
🖌 🔒 🎒 🖪 🖤 🐰 🖻	2 🖉 🗠 😫	21 XI 🏹 🚡	7 🖊 🕨 🕅	🗇 ⁄a 🔹 🙎	•
HMStationID HMRHCode	e HMDate	HMTime	HMRelativeHu	CountOfHMRel	
10011 1	21-Mar-91	6:00 AM	95	2	
10011 1	01-Apr-91	6:00 AM	91	2	
10011 1	11-May-91	6:00 AM	89	2	
10011 1	22-Jun-91	9:00 AM	63	2	
10011 1	14-Sep-91	9:00 AM	40	2	
10011 1	11-Oct-91	6:00 AM	94	2	
10011 1	02-Mar-95	12:00 AM	58	2	
10011 1	02-Mar-95	8:00 AM	77	2	
10011 1	02-Mar-95	9:00 AM	61	2	
	00 M-+ 05 ▶≉ of 51422	40.00 0.84		2	

#### 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".

Microsoft Access	
<u>File Edit Yiew Insert Tools Window H</u> elp	p
🗋 🗅 😅 🖬 🎒 🖪 🖤 👗 🖻 🖻 🚿	ダ い 🏂 - 闘 - 独 🗃 🖷 🦛 2 - 🕄 -
HMBurun0_RH_version1 : Databa	base
🛱 Open 🔛 Design 🏪 New 🗙 鸟	⁰ ፦ ☷ ∰
Objects 🖉 Create table i	e in Design view
	e by using wizard
Bull Bar Oueries	e by entering data
	Brint
Dense l	
🖾 Macros	Х сu <u>t</u>
	ao ≌ ⊆opy
Groups	
Favorites	Save <u>A</u> s Export
	Send To
	Add to Group
	Create Shortcut
×	X <u>D</u> elete
Ready	Properties NUM

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.

iable <u>N</u> ame:	ОК
-Paste Options	Cancel
C Structure Only	
Structure and <u>Data</u>	
C Append Data to Existing Table	

Step 2: Set the appropriate primary key fields. Open the new table in "Design View". Set the appropriate 'primarykey 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.

Field Name         Data Type         Description           HMStationID         Number         Regional identification code for Nile Basin Meteorological stations           HMRHCode         Text         Code of the relative humidity           HMDate         Date/Time         Date of the relative humidity reading           HMTime         Date/Time         Time of the relative humidity reading           HMRelativeHumidity         Number         Relative humidity value as a percentage	III Eile Edit Yiew Insert	RelativeHumidity Tools Window He		;]					
HMStationID       Number       Regional identification code for Nile Basin Meteorological stations         HMRHCode       Text       Code of the relative humidity         HMDate       Date/Time       Date of the relative humidity reading         HMTime       Date/Time       Time of the relative humidity reading         HMRelativeHumidity       Number       Relative humidity value as a percentage         Field Properties       Field Properties	• • 🖬 🏼 🖧 🖤	X 🖻 🖻 🚿	60 🕴 🛃	]*• ⊒•	🖻 🏠	🗗 ⁄a •	2.		
HMRHCode       Text       Code of the relative humidity         HMDate       Date/Time       Date of the relative humidity reading         HMTime       Date/Time       Time of the relative humidity reading         HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: Field Properties       Field Properties         General       Lookup       A field name can be up to 64 characters long,		Data Type							
HMDate       Date/Time       Date of the relative humidity reading         HMTime       Date/Time       Time of the relative humidity reading         HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Ima						Basin Meteor	rological	stations	
HMTime       Date/Time       Time of the relative humidity reading         HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: Field Properties       Image: Field Properties         General       Lookup       Image: Field Properties         Image: A field name can be up to 64 characters long,       A field name can be up to 64 characters long,									
HMRelativeHumidity       Number       Relative humidity value as a percentage         Image: Properties       Image: Properties         General       Lookup         A field name can be up to 64 characters long,									
Field Properties  General Lookup  A field name can be up to 64 characters long,									
General Lookup A field name can be up to 64 characters long,	HMRelativeHumidity	Number	Relative hum	idity value	as a percer	ntage			
General Lookup A field name can be up to 64 characters long,	<u> </u>								
General Lookup A field name can be up to 64 characters long,									
General Lookup A field name can be up to 64 characters long,	•		Field	Properties					_
	General Lookup								

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.

jile <u>E</u> dit <u>V</u> iew <u>I</u>	nsert Query Tools	<u>W</u> indow <u>H</u> elp			
🖩 •  🖶 🎒	🖪 🖤 🕺 🖬 🖬	l 💅 🗠 🕂 - 🛛	🖁 🔁 Σ 🛛 All	💽 🖻 🏠 🛅 🖄	· 2 .
Query1 :	Append Query	Eelec	t Query sta <u>b</u> Query		
* HMS HMD HMT HMT		⊧ <b>₽</b> ∎ <u>U</u> pda	nd Query		<u>۲</u>
Field:	HMStationID	HMRHCode	HMDate	HMTime	
Field: Table:	HMStationID HMRelativeHumidity	HMRHCode HMRelativeHumidity	HMDate HMRelativeHumidity	HMTime HMRelativeHumidity	HMRelativeHumidity
Field: Table: Sort: Append To: Criteria:	= . =				
Field: Table: Sort: Append To:	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity
Field: Table: Sort: Append To: Criteria:	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity	HMRelativeHumidity

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.

Append	? ×
Append To Table <u>N</u> ame: <u>MMRelativeHumidity_Copy</u> <u>Current Database</u> <u>Another Database</u> : <u>Ele Name</u> :	OK Cancel

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

Microsoft	Access X
Δ	You are about to append 1089445 row(s).
	Once you click Yes, you can't use the Undo command to reverse the changes. Are you sure you want to append the selected rows?
	<u>Y</u> es <u>N</u> o

In case of duplicate records, the following message appears:

	Microsoft Access can't append all the records in the append query.
1	Microsoft Access set 0 field(s) to Null due to a type conversion failure, and it didn't add 91636 record(s) to the table due to key violations, 0 record(s) due to lock violations, and 0 record(s) due to validation rule violations. Do you want to run the action query anyway? To ignore the error(s) and run the query, click Yes. For an explanation of the causes of the violations, click Help.
	<u>Yes</u> <u>N</u> o <u>H</u> elp

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.

IMStationID	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
		* of 997809	40.00 AN	

### 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.

* HM1 HM1 HM1	Rainfall_PK StationID Date Fime Rainfall			
Field:		<ul> <li>HMDate</li> </ul>	HMRainfall	 
	HMRainfall_PK	HMRainfall_PK	HMRainfall_PK	
	1			
Sort:				
Sort: Show:				
			Is Null	

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.

Microsoft	Access		×
⚠	Do you want to save chan	iges to the desi	ign of query 'SelectEmptyRecords'?
	Yes	No	Cancel

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

SelectEmptyRecords : Select Query - 🗆 × HMStationID HMRainfall HMDate 70136 ۲ 17-May-83 70136 18-May-83 70136 02-Aug-83 70136 03-Aug-83 70136 04-Aug-83 70136 05-Aug-83 70136 06-Aug-83 70136 07-Aug-83 Record: 14 📢 1 • • • • • of 12901

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.

Show Table	<u>?</u> ×
Tables Queries Both giheta InventoryDataGaps SelectEmptyRecords	<u>A</u> dd <u>C</u> lose

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 i

: 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.

* HMS HMC	ectEmpty StationID Date Rainfall Fime				
					<u> </u>
r:-ld.					-
	HMStationID	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	EMPTY: Nz([HMRainfall],"9999")	
Table:	SelectEmptyRecord:				
Table: Total:	SelectEmptyRecord: Group By	Group By	Group By	Count	
Table: Total: Crosstab:	SelectEmptyRecord:				
Table: Total: Crosstab: Sort:	SelectEmptyRecord: Group By	Group By	Group By	Count	
Table: Total: Crosstab:	SelectEmptyRecord: Group By	Group By	Group By	Count	

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	<b>1</b> 1	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	<b>1</b> 1	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.

HM * HMS HMC HMT	Select Query Rainfall StationID Date Time Rainfall			×
Field:	HMStationID	YEAR: Format([HMDate],"yyyy")	HMRainfall	
Table:	HMRainfall	<u> </u>	HMRainfall	
Total:	Group By	Group By	Count	
Sort:				
Show:				
Criteria:	20010			
or:	•			

HMStationID	YEAR	CountOfHMRai
20010	1952	335
20010	1953	365
20010	1954	365
20010	1955	365
20010	1956	366
20010	1957	365
20010	1958	365
20010	1959	243

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.

HM	Rainfall				
*					
HMS	StationID				
HMC	Date				
	Time				
HMF	Rainfall				
	real in all				
					Þ
		VEAD: Format/[HMD.sta] "vuuu")	MONTH: Format/[HMDate] "mm")		• •
Field:	HMStationID	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall HMRainfall	
 Field:		YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall HMRainfall	
Field: Table: Sort: Show:	HMStationID HMRainfall	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")		
Field: Table: Sort:	HMStationID HMRainfall			HMRainfall	

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.

HM	Rainfall				
*					
HM:	5tationID				
	Date				
HM:	Time				
НМ	Rainfall				
	Rainfall				
	Rainfall				
	J	VEAR : Format/[HMDate] "vvvv")	MONTH: Format/[HMDate] "mm")	HMRainfall	
	HMStationID	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall HMRainfall	
Field:	HMStationID HMRainfall	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm") Group By		
Field: Table: Total: Sort:	HMStationID HMRainfall Group By	Group By	Group By	HMRainfall Count	
Field: Table: Total:	HMStationID HMRainfall			HMRainfall	

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

Microsoft Access				
Eile Edit View Insert Query Too	ols <u>W</u> indow <u>H</u> elp			
🖩 🖌 🖶 🖧 🖤 🗶 Pa	n 🗈 🚿 🗠 💼 🖣 📜 😘	Σ ΑΙ 💽 🖆 🏠 🛅 🛠	ā - 🛛 -	
🛱 Query1 : Select Query	Select Query	y		
HMR ainfall * HMStationID HMDate HMTime HMRainfall	⊡! Make-Table Query #! Append Query ★! Delete Query			۲ ۲
Field: HMStationID Table: HMRainfall	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall HMRainfall	
Total: Group By Sort:	Group By	Group By	Count	
Show: 🔽 Criteria: 20010				
Ready				

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

Microsof	t Access - [Q <mark>ue</mark> i	ry1 : Crosstab Query]			
Eile Edi	t <u>V</u> iew <u>I</u> nsert (	Query <u>T</u> ools <u>W</u> indow <u>H</u> elp			_ 8 ×
🖩 🖌 🔒	a 🛛 🗸	3, 🖻 🛍 🚿 🗠 🔳 📲	t 🔓 Σ 💽 🔹 😭 🏠	💼 ⁄a • 🛛 🗸	
* HMS HMC HMT					▲  ►
		1		1 .	
	HMStationID	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall	<b>^</b>
	HMRainfall		Group By	HMRainfall Count	
	Croup By	LCROUD BU			
Crosstab:	Group By	Group By Column Heading		Value	
Crosstab: Sort:	Row Heading	Group By Column Heading	Row Heading		
Crosstab: Sort: Criteria:	Row Heading Column Heading				
Crosstab: Sort: Criteria:	Row Heading				
Crosstab: Sort: Criteria:	Row Heading Column Heading Value				
Crosstab: Sort: Criteria:	Row Heading Column Heading Value				
Crosstab: Sort: Criteria:	Row Heading Column Heading Value				
Crosstab: Sort: Criteria:	Row Heading Column Heading Value (not shown)				×

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

Sort:	*	ne				
Total:     Group By     Group By     Group By     Count       Crosstab:     Column Heading     Row Heading     Value       Sort:     Column Heading     Column Heading       Criteria:     20010     Column Heading						<u> </u>
Crosstab:     Column Heading     Row Heading     Value       Sort:			YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")	HMRainfall	
Sort:	Table: H	HMRainfall	YEAR: Format([HMDate],"yyyy")	MONTH: Format([HMDate],"mm")		
Criteria: 20010	Table: H Total: (	HMRainfall	Group By	Group By	HMRainfall Count	
	Table: H Total: ( Trosstab:	HMRainfall	Group By	Group By	HMRainfall Count	
or:	Table: H Total: ( irosstab: Sort:	HMRainfall Group By	Group By	Group By	HMRainfall Count	

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

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	

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.

Nicrosof	t Access - [Q <mark>uery1</mark> :	: Crosstab Query]			
Eile Ed	it ⊻iew <u>I</u> nsert <u>Q</u> uer	y <u>T</u> ools <u>W</u> indow <u>H</u> elp			_ 8 ×
🖩 🖌 🔚			⁰≣ ∑ 💽 🖬 🏠 🗄	🖬 \land 🛛 🗸	
*	e				۲ ۲
Field:	StationID	YEAR: Format([Date],"yyyy")	Month: Format([Date],"mm")	Rainfall	
	AuxRwanda-RainAll		rionen ronnac([baco]) min y	AuxRwanda-RainAll	
Total:	Group By	Group By	Group By	Count	
	Row Heading	Row Heading	Column Heading	Value	
Sort:					
Criteria:					
or:	•				
Ready					

Running this query results in the following output.

	3 🖪 💙 🕺	🖻 🖻 🖉	10		X↓ 🍫	<b>e</b> 2	<b>#</b>	▶* ₿	× 🗗	<b>∕</b> a •	2.		
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				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

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.

Microsof	t Access - [Query1 :	Select Query]			
Eile Ed	it <u>V</u> iew <u>I</u> nsert Query	/ <u>T</u> ools <u>W</u> indow <u>H</u> elp	)		_ 8 ×
] 🖩 🖌 🔒	8 B V X I	à 🖻 💉 🗠 🗖	ι - 🚦 🗣 Σ 🗛	· · · 🖻 🗠 🗄	) ⁄a • 🛛 •
* HMH HME HMT					▲  
Field:		HMDate	HMRainfall		
Table:	RwandaAdditionalD	RwandaAdditionalD.	RwandaAdditionalD.		
Sort: Show:					
Criteria: or:			>100		
or;	•				
Ready					

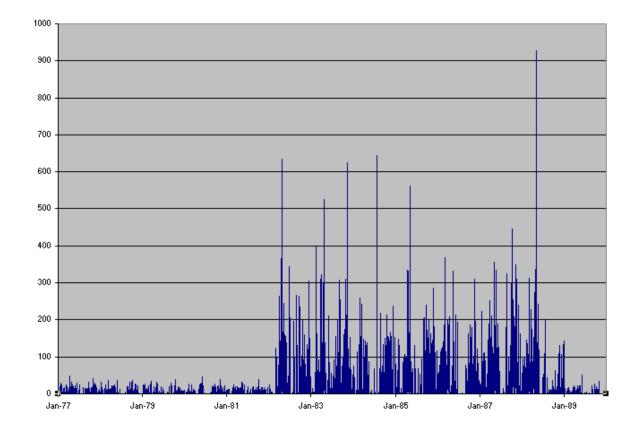
Microsoft Access	- [Query1 : Sele	ct Query]							<u>- 🗆 ×</u>
Eile Edit View	<u>I</u> nsert F <u>o</u> rmat <u>R</u> e	cords <u>T</u> ools <u>W</u> inde	ow <u>H</u> elp						_ 8 ×
🔟 -  🖬 🥔 🖸	X 🚏 👗 🖻 🖡	l 💅 🗠 🚷		V 🚡 🕅	7 🖊	▶* ₩	🗗 ⁄a •	2.	
HMHydromet_I	HMDate	HMRainfall							<b>_</b>
9129519	4/13/82	179							
9129519	4/15/82	365							_
9129519	4/17/82	231							
9129519	4/18/82	153							
9129519	4/21/82	634							
9129519	4/24/82	208							
9129519	4/26/82	112							
9129519	4/27/82	167							<b>T</b>
Record: II	1	* of 306							
Datasheet View								JUM 📃	

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.

HMRainfall_PK			
*			
HMStationID			
HMDate			
HMTime			
HMRainfall			
Field: HMStationID	HMDate	HMRainfall	
Field: HMStationID Table: HMRainfall_PK	HMDate HMRainfall_PK	HMRainfall HMRainfall_PK	
Table: <u>HMRainfall_PK</u> Sort:	HMRainfall_PK	HMRainfall_PK	
Table: HMRainfall_PK			

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;

HMStationID	HMDate	HMRainfall	
70001	01-May-83		
70001	02-May-83		
70001	03-May-83		
70001	04-May-83		
70001	05-May-83		
70001	06-May-83		
70001	07-May-83		
70001	08-May-83		
70001	09-May-83		
70001	10-May-83		
70001	11-May-83		
70001	12-May-83		
70001	13-Maγ-83		
cord: 🚺 🔳	1 🕨 🕨	• of 3957	

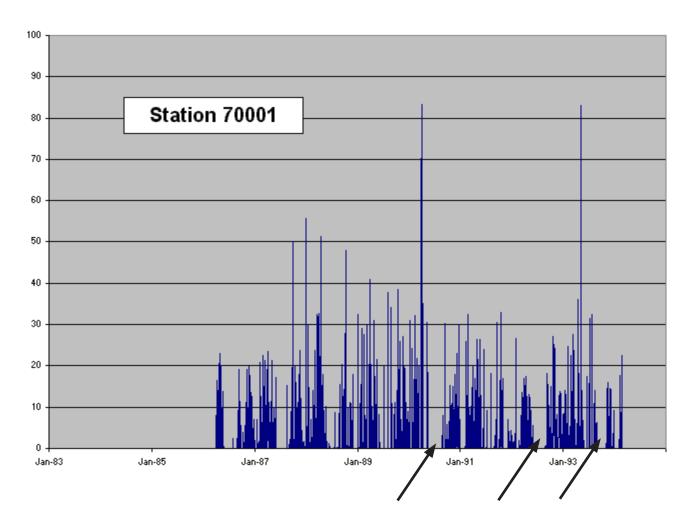
Step 3: open a new 'workbook" in MS Excel; select 'Paste' from the Edit menu; the time series for the selected station is now transferred to Excel; select all records in the 'date' and 'value' columns (by holding the 'Shift' key and navigating downwards); click the 'Chart Wizard' icon, as shown below.

🔀 Mi	crosoft Exc	el - Book2				- 0 ×
·······		w <u>I</u> nsert F <u>o</u> rmat	<u>T</u> ools <u>D</u> ata	Window A	elp	_ 8 ×
0	2 🖬 🔒	a 🕻 🖏	K) + 2	ε 🛍 100	% 🔹	* <u>·</u> · *
	B1	• = HMI	Date			
	A	В	С	D	E	F 👗
3924	70001	25-Jan-94				
3925	70001	26-Jan-94				
3926	70001	27-Jan-94				
3927	70001	28-Jan-94				
3928	70001	29-Jan-94				
3929	70001	30-Jan-94				
3930	70001	31-Jan-94				
3931	70001	01-Feb-94	0.00			
3932	70001	02-Feb-94	0.00			
3933	70001	03-Feb-94	0.00			
3934	70001	04-Feb-94	2.10			
3935	70001	05-Feb-94	0.00			
3936	70001	06-Feb-94	10.40			
3937	70001	07-Feb-94	12.20			
3938	70001	08-Feb-94	0.00			
3939	70001	09-Feb-94	0.00			
3940	70001	10-Feb-94	1.20			
3941	70001	11-Feb-94	17.70			<b>_</b>
	▶ 🕨 \Shee	t1 / Sheet2 / Shee	et3 /			
Read	у 🗌 🖂	Sum=128273095	i.6	NUM	1	

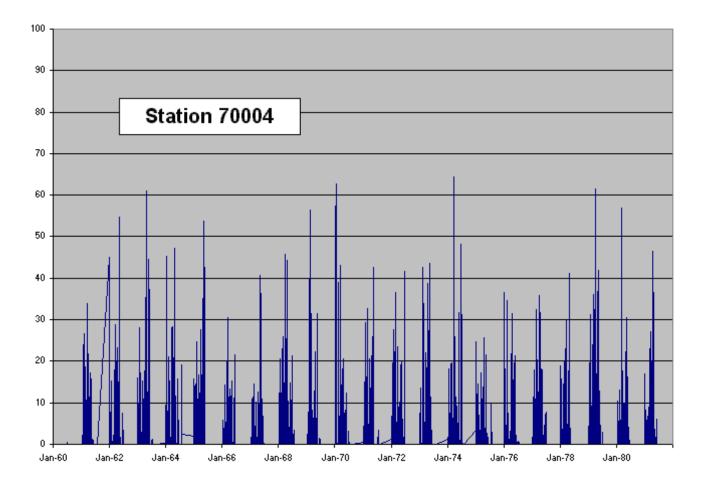
Step 4: select "Line"; follow the Chart Wizard instructions.

Standard Types Chart type: Chart type: Column Colum	Custom Types Custom Types Chart sub-type:	
Radar Surface Bubble Stock	Cancel < Back Next >	

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.



### 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.

HMRainfall * HMStationID HMDate HMTime HMRainfall			
Field: HMStationID	YEAR: Format/[HMDate]."vvvv"	MONTH: Format([HMDate]."mm")	HMRainfall
Field: HMStationID Fable: HMRainfall ADD BH	YEAR: Format([HMDate],"yyyy",	MONTH: Format([HMDate],"mm")	HMRainfall HMRainfall ADD BH
Table: HMRainfall_ADD_BH Total: Group By		MONTH: Format([HMDate],"mm") Group By	
Table: HMRainfall_ADD_BH Total: Group By Sort:	Group By	Group By	HMRainfall_ADD_BH Sum Ascending
fable: HMRainfall_ADD_B⊢ Total: Group By Sort: Show: ☑	Group By	Group By	HMRainfall_ADD_BH Sum
Table: HMRainfall_ADD_BH Total: Group By Sort:	Group By	Group By	HMRainfall_ADD_BH Sum Ascending

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.

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

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.

📰 Query3 : Select (	Juery	_ 🗆 ×
HMStationID	YEAR	SumOfHMRain
70015	1986	
70136	1986	176.7
70147	1986	273.6
70143	1986	343.1
▶ 70054	1986	386.2
70055	1986	386.2
70001	1986	487.8
70046	1986	713.06
70045	1986	900.50000000
70009	1986	953.4
70038		1076.8
70037	1986	1077.3
70138		1081
70103		1156
70061		1169.4
70043		1183.1
70034		1201.7
70052	1986	1224.4
70053		1392.5
70047		1484.72
70048		1541.2
70181		1762.8
70044	1986	2574.6
Record: 14	5 🕨 🕨	* of 23

#### 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.