A CORRECTION OF GPS READINGS FOR UGANDA'S NATIONAL HOUSEHOLD SURVEY 1999/2000

Ulrike Wood-Sichra
Consultant

A Contribution to the Strategic Criteria for Rural Investments in Productivity (SCRIP) Program of the USAID Uganda Mission

The International Food Policy Research Institute
2033 K Street, N.W. Washington, D.C. 20006

February 2002

Strategic Criteria for Rural Investments in Productivity (SCRIP) is a USAID-funded program in Uganda implemented by the International Food Policy Research Institute (IFPRI) in collaboration with Makerere University Faculty of Agriculture and Institute for Environment and Natural Resources. The key objective is to provide spatially-explicit strategic assessments of sustainable rural livelihood and land use options for Uganda, taking account of geographical and household factors such as asset endowments, human capacity, institutions, infrastructure, technology, markets & trade, and natural resources (ecosystem goods and services). It is the hope that this information will help improve the quality of policies and investment programs for the sustainable development of rural areas in Uganda. SCRIP builds in part on the IFPRI project *Policies for Improved Land Management in Uganda (1999-2002)*. SCRIP started in March 2001 and is scheduled to run until 2006.

The origin of SCRIP lies in a challenge that the USAID Uganda Mission set itself in designing a new strategic objective (SO) targeted at increasing rural incomes. The *Expanded Sustainable Economic Opportunities for Rural Sector Growth* strategic objective will be implemented over the period 2002-2007. This new SO is a combination of previously separate strategies and country programs on enhancing agricultural productivity, market and trade development, and improved environmental management.

Contact in Kampala
Simon Bolwig and Ephraim Nkonya
IFPRI, 18 K.A.R. Drive, Lower Kololo
P.O. Box 28565, Kampala
Phone: 041-234-613 or 077-591-508
Email: E.Nkonya@cgiar.org
S.Bolwig@cgiar.org

Contact in Washington, D.C.
Stanley Wood, Project Leader
IFPRI, 2033 K Street, NW,
Washington, D.C. 20006-1002, USA
Phone: 1-202-862-5600
Email: S.Wood@cgiar.org

Contents

1. Introduction
2. Comparison of Original and Corrected Data Set
3. Preparation of Data for Comparison and Correction
4. Correction of GPS Readings
5. Correction of Household ID and/or Community ID
6. Transfer of NS and EW Spreadsheets to MSAccess Tables
7. Check on HHs Distance within Community

1. Introduction

Uganda's National Household Survey 1999/2000 has been made available to the SPARG project in digital format. All the data collected is split into various MSAccess databases, each consisting of one or more tables.

The survey itself is subdivided into three main components[1]:

- Community questionnaire: 1086 records, one for each community
- Socio-economic questionnaire: 10696 records for about 10 households per community
- Crop questionnaire: 8452 records for about 8 households per community

Both Community and Socio-economic questionnaires have two GPS readings to identify longitude and latitude of the community and the household.
On plotting community and household locations with a GIS (ArcView) it appeared that a number of communities and households had wrong GPS readings, since they were located in strange places (waterbodies or outside Uganda's boundaries).

As the project intends to use the Household Survey data for cropping pattern characteristics at different levels of administrative units (region, district, county, sub-county, parish), it is important to identify the correct geographical location of communities and households. Thus an attempt was made to correct the GPS readings when possible, following a semi-automated process in combination with knowledge of possible error sources and some optical pattern recognition.

A few assumptions underlie this process:

- coordinates of a community and their "corresponding" households should be similar, at least in the degree value (except when straddling the end of one and the beginning of the next degree), and possibly in the min value,
- first and second reading should be very similar (with the exception mentioned above)
- it may be that some households are spread out in a community, but not more than 3-4 minutes (approx. 6-10 Km),

[1] a fourth component: *Listing Questionnaire* is mentioned in the documentation but was not made available.

- readings may be wrong due to keying errors (1 and not 7, 5 and not 8, missing 0, etc)
- coding of household IDs in the community or socio-economic questionnaire may be wrong (1 and not 2, 2 and not 1), in which case the LC1 name should be used to back-up the modification.

All comparisons, new coordinates, new IDs are done in new tables, stored in CropData.mdb.

2. Comparison of Original and Corrected Data Set

item	**original data set** *	**after correction***
# communities	1086	1086
# HH in socio-economic survey	10696	10696
# communities **without** HH IDs	28	0
# HH **without** community ID in socio-economic survey	374 (with NS and/or EW problems)	0
# HH with possible **error** in GPS reading **	about 2050 (for NS and/or EW)	703 (for NS and/or EW), 103 not correctable
# communities with possible **error** in GPS reading ***	80 for EW and 110 for NS	69 (for NS and/or EW), 11 not correctable
# comm with <= 5 HHs	50	3
# comm with <=9 HHs	245	233
# comm with 10 HHs	906	960
# comm with HHs >= 8 min apart	433 for EW, 236 for NS	29 for EW, 13 for NS
# comm with HHs >= 6 min apart	681 for EW, 255 for NS	37 for EW, 15 for NS
# comm with HHs >= 4 min apart	859 for EW, 273 for NS	49 for EW, 31 for NS
# HHs corrected		1211 (for NS and/or EW)
# HHs possibly OK		8679 (for NS and/or EW)
# communities corrected		223 (for NS and/or EW)
# communities possibly OK		807 (for NS and/or EW)

note: 1 min = approx 2 km, 8 min = approx 16 km.

(*) numbers may not add up to total numbers of communities or HHs (depending on how they were calculated)
(**) flags "?" or "x" in fields ModSoEW and/or ModSoNS
(***) flags "?" or "x" in fields ModCoEW and/or ModCoNS

3. Preparation of Data for Comparison and Correction

A sub-set of data was extracted from tables in three databases of the National Household Survey in order to compare the GPS readings of households and their related communities.

The extraction was done in several steps, using MSAccess query procedures.

The relevant three databases are:

directory	database	table	size of HH-ID
comm	sec1.mdb	sec1-households	12 characters
crop	cropdata.mdb	sec1-household	13 characters
socio	sociodata.mdb	sec1-household	13 characters

Step 1 – create tables with new HHId10 to link the three tables

HH-ID in the three tables has the following composition:

comm region(3 char), area(1 char), OldNewEA(1), "0", EA(4), LC1(2)
crop region(3), area(1), OldNewEA(1), "0", EA(4), OldNewHH(1), HHNr(2)
socio stratum(3), area(1), OldNewEA(1), "0", EA(4), OldNewHH(1), HHNr(2)

The leftmost 10 characters of HH-ID in each data table will be used as a link for the tables.

With three separate queries (A, B, C) create three new tables:

- Table-CommID10
- Table-CropID10
- Table-SocioID10
-

with all required fields for the corrections.

Step 2 – combine GPS readings for all comm records and corresponding socio records.

With two separate MSAccess queries (C: for NS coordinates and D: for EW coordinates) link both new tables Table-CommID10 and Table-SocioID10 and extract the required information to new tables:

- NewGPS-NS-Ver1 and
- NewGPS-EW-Ver1

Double GPS readings and LC1 (village) names and codes are taken from two tables:

Table-CommID10	NSreading	NDDDMMsss
	EWreading	DDMMsss
	(sss are seconds expressed in 1/1000 of minutes)	
	LC1Name	possibly village name
	Lc1Code	possibly village code
Table-SocioID10	NS	n (1 for North, 2 for South)
	degNS	n
	minNS	nn

secNS	nnn (1/1000 of minutes)
degEW	nn
minEW	nn
secEW	nnn (1/1000 of minutes)
LcName	possibly village name
LcCode	possibly village code

The GPS readings of the Table-CommID10 were split into NS, deg, min and sec to make them comparable to the socio table readings.

Village names and codes were taken for backup checks.

At the same time the following differences were flagged:

Between reading 1 of comm. and socio:	000	one or both readings do not exist
	n	difference in NS identifier
	d	difference in degrees
	m	difference in minutes
	dm	difference for both degrees and minutes
	---	no differences detected

Between reading 2 of comm. and socio:	like above
Between reading 1 and 2 of comm:	like above, without 000 option
Between reading 1 and 2 of socio:	like above, without 000 option

No comparison is made between first and second readings of seconds.

The flags are kept in the following fields ("ew" in names for EW coordinates, "ns" in names for NS coordinates):

- CSew1 comparison between community and household coordinates, 1st reading EW
- CSew2 comparison between community and household coordinates, 2nd reading EW
- Cew1-ew2 comparison between 1st and 2nd community reading EW
- ew1-ew2 comparison between 1st and 2nd household reading EW

New fields are also created in each table to flag the correction process for community and household coordinates:

- ModSo modification flag for household coordinates, contents can be:
  - yes modified
  - ? may need modification, something still strange
  - x coordinates (and point) should be discarded
  - ok seems strange, but is OK
  - blank nothing was modified
- NewSoNS new NS identifier for household coordinate (1st or 2nd), taken from

    - rep representative household
    - ci corresponding community reading (i =1st or 2nd)
    - si corresponding household reading (i=1st or 2nd)
- NewSoDeg new degrees number for household coordinate (1st or 2nd), taken from
    - rep representative household
    - ci corresponding community reading (i =1st or 2nd)
    - si corresponding household reading (i=1st or 2nd)
- NewSoMin new minutes number for household coordinate (1st or 2nd), taken from
    - rep representative household
    - ci corresponding community reading (i =1st or 2nd)
    - si corresponding household reading (i=1st or 2nd)
- ModCo same as ModSo for community coordinates
- NewCoNS same as NewCoNS for community coordinates
- NewCoDeg same as NewCoDeg for community coordinates
- NewCoMin same as NewCoMin for community coordinates

Step 3 – export resulting tables to MSExcel

Open both tables (New-GPS-NS-Ver1 and New-GPS-EW-Ver1), block the entire table and export it to MSExcel (Tools/Office Links/Analyze it with MSExcel): New-GPS-NS-Ver1 and New-GPS-EW-Ver1.

Step 4 – identify households which do not have a corresponding community

Reopen queries C and D and modify as follows:

- change the relationship to "All records of Table-SocioID10",
- change source of field HHId10 from Table-CommID10 to Table-SocioID10,
- change query to "select";

and export the resulting table to MSExcel. In Excel color code the HHId10 column with a different color (to identify households which have no community).

Append these new records to the MSExcel tables created in Step 3.

4. Correction of GPS Readings

It is possible to sort the spreadsheets according to differences in communities or households and correct where necessary, but a better picture is had by sorting the spreadsheets by:
- HHId10 and
- S (household ID)

to see the groupings of households in communities.

Corrections are now executed depending on the flags:

for household GPS readings:

if 1st or 2nd reading is missing	replace with the other
if 1st and 2nd reading is missing	replace both with community GPS readings
if NS, deg or min are wrong	correct in accordance with mayority (min 70%)

for community GPS readings:

if 1st or 2nd reading is missing	replace with the other
if 1st and 2nd reading is missing	replace both with representative household GPS reading (min 70% similar)
if NS, deg or min are wrong	correct in accordance with mayority of household GPS reading (min 70%)

If both household and community GPS readings are flagged and a representative household cannot be identified (70% minimum), exercise judgement, possibly flag the ModSo and ModCo fields with "?" or "x".
Since community GPS readings appear blockwise, they can also be copied and pasted blockwise. Attention must be paid to leading "0" when new coordinates are assembled and calculated.

5. Correction of Household ID and/or Community ID

There are some community records which have no associated households, and accordingly some households which have no corresponding community. This hints at a wrong coding or keying of either household ID or community ID. The errors are all found in the 5th digit: OldNewEA.

With the help of colour coded fields in the spreadsheet and

- "guessing" the possibly wrong digit (1 instead of 2, 2 instead of 1),
- seeing communities with only 4 or 5 households,
- seeing 10 or more households without community,
- checking LC1 names

it is possible to identify and correct IDs in households and/or communities.

Eleven new columns are introduced in the spreadsheets (replace text "ew" or "EW" in column names with "ns" or "NS" for spreadsheet with NS coordinates):

Five to be filled in while correcting:

- Snew new household ID
- Cnew new community ID
- HHId10New new 10 character ID (for both households and communities), is the same as HHId10 if no recoding took place

- RepSo representative number for household
- RepCo representative number for community

and 6 which are calculated with formulas:

- NewC-S indicates if new ID for comm (C), household (S) or both (CS, does not occur)
- 1EWrNew assembly of new 1st reading of community EW coordinates to give a 7 (or 8 for NS) digit number
- 2EWrNew assembly of new 2nd reading of community EW coordinates to give a 7 (or 8 for NS) digit number
- Cewr1decNew calculated new 1st reading of community EW coordinates according to formula
- Cewr2decNew calculated new 2nd reading of community EW coordinates according to formula
- Sewr1decNew calculated new 1st reading of household EW coordinates according to formula
- Sewr2decNew calculated new 2nd reading of household EW coordinates according to formula

ATTENTION: It may be necessary to recreate columns HHId10, HHId10New, S, Snew, C, Cnew if edits are made, since sortings will create wrong results (mixture of text and number fields, despite formatting of complete column). The easiest solution is to insert new temporary columns, fill them with "=left(cellnr,n)", where n = 10 for HHId10 and HHId10New, 12 for C and Cnew, 13 for S and Snew.

This process was carried out first in the spreadsheet NewGPS-NS-Ver1, and then the modifications were also introduced into NewGPS-EW-Ver1. They must be identical.

Resulting corrections have been introduced in the new "Corrections" Table, but they do not always translate correctly into the Crop and Socio Surveys. More work is required here.

SUGGESTION: In a next exercise it is better to combine NS and EW readings into one table (and spreadsheet) in order to simplify the ID modifications.

Resulting corrections have been introduced in the new "Corrections" Table, but they do not always translate correctly into the Crop and Socio Surveys. More work is required here.

6. Transfer of NS and EW Spreadsheets to MSAccess Tables

In the CropData.mdb database import 2 spreadsheets to create new tables:

spreadsheet	**MSAccess table**
New-GPS-NS-Ver1	GPS-NS-corrections
New-GPS-EW-Ver1	GPS-EW-corrections

Specifications for the import are:

- first records are field names,
- import into new table
- no key field,

Check the resulting tables and ensure they can be linked to the original tables "sec1-household" in CropData.mdb, SocioData.mdb and Sec1.mdb of directory Comm.

NOTE: In my version of spreadsheets and tables I did not manage to correctly convert the spreadsheet NEW-GPS-EW-Ver1 into an MSAccess table using the above specifications. An intermediate solution was used:

- import the spreadsheet into the table EW-NEW with specifications:
    - no record with field names
    - import into new table (compulsory)
    - no key field
- delete column with field names in table EW-NEW,
- delete all records in existing table GPS-EW-corrections, and
- with query Q-convert-EWcorrections create the table GPS-EW-corrections.

In a final step combine both spreadsheets with query Q-Assemble-All-Corrections and create the final table GPS-All-corrections with all relevant data from GPS-NS-corrections and GPS-EW-corrections. At the same time include original HH coordinate readings in decimal form from query Q-Old-HHCoordinates.

With these new tables it is possible to relate "old" household or community IDs with "new" household or community IDs and to retrieve all required data.

7. Check on HHs Distance within Community

As a last step, compare blockwise the coordinate readings of all households in each community to identify the range of household coordinates.

A set of MSAccess queries computes minima, maxima and range for the decimal coordinates of both EW and NS readings of the households in each community. Resulting tables can be sorted by range (query Q-Decimal-Differences3-EW and Q-Decimal-Differences3-NS, which both have underlying queries Q-Decimal-Differences1 and Q-Decimal-Differences2).

As a general rule it was assumed that households are within a radius of 6-8 coordinate minutes, i.e. 12 – 18 Km (min-max >= 0.06 to 0.09). But on inspections of lower threshholds some HHs were corrected when the minute value of only one (of mostly 10) was outside the value of the others. Households which lay too far apart in a community were flagged as dubious (?).

The corrections are best done by leaving open the comparison table, opening the corresponding table (EW or NS), finding the community and checking on the source of the discrepancy and maybe correcting it.

After having done modifications, reassemble both tables EW and NS into ALL again.

ANNEX 1

MSAccess Table Structures

Table: GPS-All-Corrections
Final table after including all modifications and calculations, with original data to allow comparisons, and merge with old decimal coordinates of households (query Q-Old-HHcoordinates).

Fields:

Name	**Contents** (the order may differ in the Access Table itself)	**Type**	**Size** *
HHId10	original 10 digit ID	Text	255
HHId10New	new 10 digit ID	Text	255
C	original community ID	Text	255
Cnew	new community ID	Text	255
S	original HH ID	Text	255
Snew	new HH ID	Text	255
RepSo	representative HH ID	Text	255
NewC-S	flag comparison old / new community and HH IDs	Text	255
RepCo	representative community ID	Text	255
ModCoEw	flag for modification of community EW readings	Text	255
NewCoDegEw	flag for source of new community EW degrees	Text	255
NewCoMinEw	flag for source of new community EW minutes	Text	255
1EWr	1st original community EW reading	Double	8
2EWr	2nd original community EW reading	Double	8
1EWrNew	1st new community EW reading	Text	255
2EWrNew	2nd new community EW reading	Text	255
1degEW	1st reading community EW degrees (old or new)	Text	255
1minEW	1st reading community EW minutes (old or new)	Text	255
1secEW	1st reading community EW seconds (old or new)	Text	255
2degEW	2nd reading community EW degrees (old or new)	Text	255
2minEW	2nd reading community EWminutes (old or new)	Text	255
2secEW	2nd reading community EWseconds (old or new)	Text	255
Cew1decNew	1st new community EW reading in decimal notation	Double	8
Cew2decNew	2nd new community EW reading in decimal notation	Double	8
Sew1DecOld	1st old HH EW reading in decimal notation	Double	8
Sew2DecOld	2nd old HH EW reading in decimal notation	Double	8
ModSoEW	flag for modification of HH EW readings	Text	255
NewSoDegEw	flag for source of new HH EW degrees	Text	255
NewSoMinEw	flag for sourc of new HH EW minutes	Text	255
gpse1	1st reading HH degrees EW (old or new)	Double	8
gpse2	1st reading HH minutes EW (old or new)	Double	8
gpse3	1st reading HH seconds EW (old or new)	Double	8

Name	**Contents** (the order may differ in the Access Table itself)	**Type**	**Size** *
gpse4	2nd reading HH degrees EW (old or new)	Double	8
gpse5	2nd reading HH minutes EW (old or new)	Double	8
gpse6	2nd reading HH seconds EW(old or new)	Double	8
Sew1decNew	1st new HH EW reading in decimal notation	Double	8
Sew2decNew	2nd new HH EW reading in decimal notation	Double	8
Lc1NameEW	village name (from community) from EW part	Text	255
Lc1CodeEW	village code (from community) from EW part	Double	8
LcNameEW	village name (from HH) from EW part	Text	255
LcCodeEW	village code (from HH) from EW part	Double	8
XXX	*empty separator field for EW and NS*	Text	255
ModCoNS	flag for modification of community NS readings	Text	255
NewCoNsNS	flag for source of new HH Ns orientation	Text	255
NewCoDegNS	flag for source of new HH NS degrees	Text	255
NewCoMinNS	flag for sourc of new HH NS minutes	Text	255
1NSr	1st original community NS reading	Double	8
2NSr	2nd original community NS reading	Double	8
1NSrNew	1st new community NS reading	Text	255
2NSrNEw	2nd new community NS reading	Text	255
1nsNS	1st reading community NS orientation (old or new)	Text	255
1degNS	1st reading community NS degrees (old or new)	Text	255
1minNS	1st reading community NS minutes (old or new)	Text	255
1secNS	1st reading community NS seconds (old or new)	Text	255
2nsNS	2nd reading community NS orientation (old or new)	Text	255
2degNS	2nd reading community NS degrees (old or new)	Text	255
2minNS	2nd reading community NS minutes (old or new)	Text	255
2secNS	2nd reading community NS seconds (old or new)	Text	255
Cnsr1decNew	1st new community NS reading in decimal notation	Double	8
Cnsr2decNew	2nd new community NS reading in decimal notation	Double	8
Snsr1DecOld	1st old HH NS reading in decimal notation	Double	8
Snsr2DecOld	2nd old HH NS reading in decimal notation	Double	8
ModSoNS	flag for modification of HH NS readings	Text	255
NewSoNSNs	flag for source of new HH NS orientation	Text	255
NewSoDegNs	flag for source of new HH NS degrees	Text	255
NewSoMinNs	flag for sourc of new HH NS minutes	Text	255
ns1	1st reading HH NS orientation (old or new)	Text	255
gpsns1	1st reading HH degrees NS (old or new)	Text	255
gpsns2	1st reading HH minutes NS (old or new)	Text	255
gpsns3	1st reading HH seconds NS (old or new)	Text	255
ns2	2nd reading HH NS orientation (old or new)	Text	255
gpsns4	2nd reading HH degrees NS (old or new)	Text	255
gpsns5	2nd reading HH minutes NS (old or new)	Text	255
gpsns6	2nd reading HH seconds NS (old or new)	Text	255

Name	**Contents** (the order may differ in the Access Table itself)	**Type**	**Size** *
Snsr1decNew	1st new HH NS reading in decimal notation	Double	8
Snsr2decNew	2nd new HH NS reading in decimal notation	Double	8

(*) large text fields are created automatically by MSAccess during the queries.

Query: Q-Old-HHcoordinates

SELECT DISTINCT [Table-SocioID10].HHId10, [Table-SocioID10].householdId,
[gpse1]+[gpse2]/60+[gpse3]/1000/60 AS Sew1DecOld,
[gpse4]+[gpse5]/60+[gpse6]/1000/60 AS
Sew2DecOld, IIf([ns1]=1,1,-1)*([gpsns1]+[gpsns2]/60+[gpsns3]/1000/60) AS Snsr1DecOld,
IIf([ns2]=1,1,-1)*([gpsns4]+[gpsns5]/60+[gpsns6]/1000/60) AS Snsr2DecOld
FROM [Table-SocioID10]

Table: GPS-EW-corrections
Table resulting from corrections performed in Excel, and imported from there.
In this version there is a problem with direct import (with field names) from Excel into Access, therefore an intermediary table (EW-NEW) is used.

Fields:

Name	**Contents**	**Type**	**Size** *
HHId10	original 10 digit ID	Text	255
HHId10New	new 10 digit ID	Text	255
C	original community ID	Text	255
Cnew	new community ID	Text	255
S	original HH ID	Text	255
Snew	new HH ID	Text	255
CSew1	flag comparison community ID/household ID 1st reading	Text	255
CSew2	flag comparison community ID/household ID 2nd reading	Text	255
RepSo	representative HH ID	Text	255
NewC-S	flag comparison old / new community and HH IDs	Text	255
RepCo	representative community ID	Text	255
ModCo	flag for modification of community readings	Text	255
NewCoDeg	flag for source of new community degrees	Text	255
NewCoMin	flag for source of new community minutes	Text	255
Cew1-ew2	flag comparison community ID 1st and 2nd reading	Text	255
ew1-ew2	flag comparison HH ID 1st and 2nd reading	Text	255

Name	Contents	Type	Size *
1EWr	1st original community EW reading	Double	8
2EWr	2nd original community EW reading	Double	8
1EWrNew	1st new community EW reading	Text	255
2EWrNew	2nd new community EW reading	Text	255
1deg	1st reading community degrees (old or new)	Text	255
1min	1st reading community minutes (old or new)	Text	255
1sec	1st reading community seconds (old or new)	Text	255
2deg	2nd reading community degrees (old or new)	Text	255
2min	2nd reading community minutes (old or new)	Text	255
2sec	2nd reading community seconds (old or new)	Text	255
Cew1decNew	1st new community EW reading in decimal notation	Double	8
Cew2decNew	2nd new community EW reading in decimal notation	Double	8
ModSo	flag for modification of HH readings	Text	255
NewSoDeg	flag for source of new HH degrees	Text	255
NewSoMin	flag for sourc of new HH minutes	Text	255
gpse1	1st reading HH degrees (old or new)	Double	8
gpse2	1st reading HH minutes (old or new)	Double	8
gpse3	1st reading HH seconds (old or new)	Double	8
gpse4	2nd reading HH degrees (old or new)	Double	8
gpse5	2nd reading HH minutes (old or new)	Double	8
gpse6	2nd reading HH seconds (old or new)	Double	8
Sew1decNew	1st new HH EW reading in decimal notation	Double	8
Sew2decNew	2nd new HH EW reading in decimal notation	Double	8
Lc1Name	village name (from community)	Text	255
Lc1Code	village code (from community)	Double	8
LcName	village name (from HH)	Text	255
LcCode	village code (from HH)	Double	8
1NSr	original 1st reading of community NS coordinates	Double	8
2NSr	original 2nd reading from community NS coordinates	Double	8

(*) large text fields are created automatically by MSAccess during the queries.

Table: GPS-NS-corrections
Table resulting from corrections performed in Excel, and imported from there.

Fields:

Name	**Contents** (except where filled, all same as in table GPS-EW-corrections, but for NS coordinates – note also names with “ns” instead of “ew”)	Type	Size *
HHId10		Text	255
HHId10New		Text	255
C		Text	255

Name	**Contents** (except where filled, all same as in table GPS-EW-corrections, but for NS coordinates – note also names with "ns" instead of "ew")	**Type**	**Size** *
Cnew		Text	255
S		Text	255
Snew		Text	255
CSns1		Text	255
CSns2		Text	255
RepSo		Text	255
NewC-S		Text	255
RepCo		Text	255
ModCo		Text	255
NewCoNS		Text	255
NewCoDeg		Text	255
NewCoMin		Text	255
Cns1-ns2		Text	255
ns1-ns2		Text	255
1NSr		Double	8
2NSr		Double	8
1NSrNew		Text	255
2NSrNEw		Text	255
1ns	number for North(1) or South(2) orientation in 1st reading of community coordinates	Text	255
1deg		Text	255
1min		Text	255
1sec		Text	255
2ns		Text	255
2deg	number for North(1) or South(2) orientation in 2nd reading of community coordinates	Text	255
2min		Text	255
2sec		Text	255
Cnsr1decNew		Double	8
Cnsr2decNew		Double	8
ModSo		Text	255
NewSoNS	flag for source of new HH North-South orientation	Text	255
NewSoDeg		Text	255
NewSoMin		Text	255
ns1	number for North(1) or South(2) orientation in 1st reading of HH coordinates	Text	255
gpsns1		Text	255
gpsns2		Text	255
gpsns3		Text	255
ns2	number for North(1) or South(2) orientation in 2nd reading of HH coordinates	Text	255

Name	**Contents** (except where filled, all same as in table GPS-EW-corrections, but for NS coordinates – note also names with "ns" instead of "ew")	**Type**	**Size** *
gpsns4		Text	255
gpsns5		Text	255
gpsns6		Text	255
Snsr1decNew		Double	8
Snsr2decNew		Double	8
East1stReading	original 1st reading of community EW coordinates	Double	8
East2ndReading	original 2nd reading of community EW coordinates	Double	8
Lc1Name		Text	255
Lc1Code		Double	8
LcName		Text	255
LcCode		Double	8

(*) large text fields are created automatically by MSAccess during the queries.

Table: New-GPS-EW-Ver1
Table resulting from the combination of selected community and socio-economic questionnaire fields, including calculated fields to flag errors. Only includes EW coordinates in details. (It would have been better to combine this table with the next – New-GPS-NS-Ver1).

Fields:

Name	**Contents** (same as corresponding fields in GPS-EW-corrections – the order may sometimes be different)	**Type**	**Size** *
HHId10		Text	255
C		Text	13
S		Text	13
CSew1		Text	255
CSew2		Text	255
RejCo		Text	255
NewCoDeg		Text	255
NewCoMin		Text	255
Cew1-ew2		Text	255
1NSr		Long Integer	4
1EWr		Long Integer	4
1deg		Text	255
1min		Text	255
1sec		Text	255
2NSr		Long Integer	4

Name	**Contents** (same as corresponding fields in GPS-EW-corrections – the order may sometimes be different)	**Type**	**Size** *
2EWr		Long Integer	4
2deg		Text	255
2min		Text	255
2sec		Text	255
ModSo		Text	255
NewSoDeg		Text	255
NewSoMin		Text	255
ew1-ew2		Text	255
gpse1		Integer	2
gpse2		Integer	2
gpse3		Integer	2
gpse4		Integer	2
gpse5		Integer	2
gpse6		Integer	2
Lc1Name		Text	30
Lc1Code		Integer	2
LcName		Text	30
LcCode		Integer	2

(*) large text fields are created automatically by MSAccess during the queries.

Table: New-GPS-NS-Ver1
Table resulting from the combination of selected community and socio-economic questionnaire fields, including calculated fields to flag errors. Only includes NS coordinates in details. (It would have been better to combine this table with the previous – New-GPS-EW-Ver1).

Fields:

Name	**Contents** (same as corresponding fields in GPS-NS-corrections – the order may sometimes be different)	**Type**	**Size** *
HHId10		Text	255
C		Text	13
S		Text	13
CSns1		Text	255
CSns2		Text	255
RejCo		Text	255
NewCoNS		Text	255
NewCoDeg		Text	255
NewCoMin		Text	255
Cns1-ns2		Text	255

Name	**Contents** (same as corresponding fields in GPS-NS-corrections – the order may sometimes be different)	**Type**	**Size** *
1NSr		Long Integer	4
1ns		Text	255
1deg		Text	255
1min		Text	255
1sec		Text	255
2NSr		Long Integer	4
2ns		Text	255
2deg		Text	255
2min		Text	255
2sec		Text	255
ModSo		Text	255
NewSoNS		Text	255
NewSoDeg		Text	255
NewSoMin		Text	255
ns1-ns2		Text	255
ns1		Integer	2
gpsns1		Integer	2
gpsns2		Integer	2
gpsns3		Integer	2
ns2		Integer	2
gpsns4		Integer	2
gpsns5		Integer	2
gpsns6		Integer	2
East1stReading		Long Integer	4
East2ndReading		Long Integer	4
Lc1Name		Text	30
Lc1Code		Integer	2
LcName		Text	30
LcCode		Integer	2

(*) large text fields are created automatically by MSAccess during the queries.

ANNEX 2

MSAccess Query Specifications

CREATE TABLES WITH NEW HH-ID:

A. Make-table query: CommID10 (creates Table-CommID10 with 1086 records)

SELECT [sec1-household-comm].householdId, Left([householdId],10) AS HHId10, [sec1-household-comm].Region, [sec1-household-comm].Area, [sec1-household-comm].OldNewEA, [sec1-household-comm].EA, [sec1-household-comm].LC1, [sec1-household-comm].Lc1Name, [sec1-household-comm].Lc1Code, [sec1-household-comm].[N/S1stReading], [sec1-household-comm].East1stReading, [sec1-household-comm].[N/S2ndReading], [sec1-household-comm].East2ndReading INTO [Table-CommID10]
FROM [sec1-household-comm]
ORDER BY [sec1-household-comm].householdId, [sec1-household-comm].EA;

B. Make-table query: CropID10 (creates Table-CropID10 with 8452 records)

SELECT [sec1-household].householdId, Left([householdId],10) AS HHId10, [sec1-household].Region, [sec1-household].Area, [sec1-household].OldNewEA, [sec1-household].EA, [sec1-household].OldNewHousehold, [sec1-household].HouseholdNumber, [sec1-household].LCName, [sec1-household].LCCode INTO [Table-CropID10]
FROM [sec1-household]
ORDER BY [sec1-household].householdId;

C. Make-table query: SocioID10 (creates Table-SocioID10 with 10696 records)

SELECT [sec1-household-socio].householdId, Left([householdId],10) AS HHId10, [sec1-household-socio].Stratum, [sec1-household-socio].Area, [sec1-household-socio].OldNewEA, [sec1-household-socio].EA, [sec1-household-socio].OldNewHousehold, [sec1-household-socio].HouseholdNumber, [sec1-household-socio].LcName, [sec1-household-socio].LcCode, [sec1-household-socio].ns1, [sec1-household-socio].gpsns1, [sec1-household-socio].gpsns2, [sec1-household-socio].gpsns3, [sec1-household-socio].gpse1, [sec1-household-socio].gpse2, [sec1-household-socio].gpse3, [sec1-household-socio].ns2, [sec1-household-socio].gpsns4, [sec1-household-socio].gpsns5, [sec1-household-socio].gpsns6, [sec1-household-socio].gpse4, [sec1-household-socio].gpse5, [sec1-household-socio].gpse6 INTO [Table-SocioID10]
FROM [sec1-household-socio]
ORDER BY [sec1-household-socio].householdId;

CREATE TABLES FOR CORRECTIONS

D. Make-table query: Comm+Socio-NS (10696 records with right join: All Socio records, 10350 with left join: All Comm records)

```
SELECT [Table-CommID10].HHId10, [Table-CommID10].householdId AS C, [Table-SocioID10].householdId AS S, " "+IIf(IsNull([1ns]) Or IsNull([ns1])," o ",IIf([1ns]<>[ns1]," N "," - "))+IIf(IsNull([1deg]) Or IsNull([gpsns1])," o ",IIf([1deg]<>[gpsns1]," d "," - "))+IIf(IsNull([1min]) Or [1min]="" Or IsNull([gpsns2])," o ",IIf([1min]<>[gpsns2]," m "," - ")) AS CSns1, " "+IIf(IsNull([2ns]) Or IsNull([ns2])," o ",IIf([2ns]<>[ns2]," N "," - "))+IIf(IsNull([2deg]) Or IsNull([gpsns4])," o ",IIf([2deg]<>[gpsns4]," d "," - "))+IIf(IsNull([2min]) Or [2min]="" Or IsNull([gpsns5])," o ",IIf([2min]<>[gpsns5]," m "," - ")) AS CSns2, " " AS ModCo, " " AS NewCoNS, " " AS NewCoDeg, " " AS NewCoMin, " "+IIf([1ns]<>[2ns]," N "," - ")+IIf([1deg]<>[2deg]," d "," - ")+IIf([1min]<>[2min]," m "," - ") AS [Cns1-ns2], [Table-CommID10].[N/S1stReading] AS 1NSr, Mid([1nsr],1,1) AS 1ns, Mid([1NSr],2,3) AS 1deg, Mid([1nsr],5,2) AS 1min, Mid([1nsr],7,3) AS 1sec, [Table-CommID10].[N/S2ndReading] AS 2NSr, Mid([2nsr],1,1) AS 2ns, Mid([2NSr],2,3) AS 2deg, Mid([2NSr],5,2) AS 2min, Mid([2nsr],7,3) AS 2sec, " " AS ModSo, " " AS NewSoNS, " " AS NewSoDeg, " " AS NewSoMin, " "+IIf(IsNull([ns1])," o ",IIf(IsNull([ns2])," o ",IIf([ns1]<>[ns2]," N "," - )))+IIf(IsNull([gpsns1])," o ",IIf(IsNull([gpsns4])," o ",IIf([gpsns1]<>[gpsns4]," d "," - )))+IIf(IsNull([gpsns2])," o ",IIf(IsNull([gpsns5])," o ",IIf([gpsns2]<>[gpsns5]," m "," - ))) AS [ns1-ns2], [Table-SocioID10].ns1, [Table-SocioID10].gpsns1, [Table-SocioID10].gpsns2, [Table-SocioID10].gpsns3, [Table-SocioID10].ns2, [Table-SocioID10].gpsns4, [Table-SocioID10].gpsns5, [Table-SocioID10].gpsns6, [Table-CommID10].East1stReading, [Table-CommID10].East2ndReading, [Table-CommID10].Lc1Name, [Table-CommID10].Lc1Code, [Table-SocioID10].LcName, [Table-SocioID10].LcCode INTO [New-GPS-NS-Ver1] FROM [Table-CommID10] LEFT JOIN [Table-SocioID10] ON [Table-CommID10].HHId10 = [Table-SocioID10].HHId10;
```

E. Make-table query: Comm+Socio-EW (10696 records with right join: All Socio records, 10350 with left join: All Comm records)

```
SELECT [Table-CommID10].HHId10, [Table-CommID10].householdId AS C, [Table-SocioID10].householdId AS S, " "+IIf(IsNull([1deg]) Or IsNull([gpse1])," o ",IIf([1deg]<>[gpse1]," d "," - "))+IIf(IsNull([1min]) Or [1min]="" Or IsNull([gpse2])," o ",IIf([1min]<>[gpse2]," m "," - ")) AS CSew1, " "+IIf(IsNull([2deg]) Or IsNull([gpse4])," o ",IIf([2deg]<>[gpse4]," d "," - "))+IIf(IsNull([2min]) Or [2min]="" Or IsNull([gpse5])," o ",IIf([2min]<>[gpse5]," m "," - ")) AS CSew2, " " AS ModCo, " " AS NewCoDeg, " " AS NewCoMin, " "+IIf([1deg]<>[2deg]," d "," - ")+IIf([1min]<>[2min]," m "," - ") AS [Cew1-ew2], [Table-CommID10].[N/S1stReading] AS 1NSr, [Table-CommID10].East1stReading AS 1EWr, Mid([1EWr],1,2) AS 1deg, Mid([1EWr],3,2) AS 1min, Mid([1EWr],5,3) AS 1sec, [Table-CommID10].[N/S2ndReading] AS 2NSr, [Table-CommID10].East2ndReading AS 2EWr, Mid([2EWr],1,2) AS 2deg, Mid([2EWr],3,2) AS 2min, Mid([2EWr],5,3) AS 2sec, " "
```

AS ModSo, " " AS NewSoDeg, " " AS NewSoMin, " "+IIf([gpse1]<>[gpse4]," d "," - ")+IIf([gpse2]<>[gpse5]," m "," - ") AS [ew1-ew2], [Table-SocioID10].gpse1, [Table-SocioID10].gpse2, [Table-SocioID10].gpse3, [Table-SocioID10].gpse4, [Table-SocioID10].gpse5, [Table-SocioID10].gpse6, [Table-CommID10].Lc1Name, [Table-CommID10].Lc1Code, [Table-SocioID10].LcName, [Table-SocioID10].LcCode INTO [New-GPS-EW-Ver1]
FROM [Table-CommID10] LEFT JOIN [Table-SocioID10] ON [Table-CommID10].HHId10 = [Table-SocioID10].HHId10;

CREATE TABLE FROM SPREADSHEET FOR EW COORDINATES (solution for my problem)

G: Append-table query : Q-Convert-EWCorrections

INSERT INTO [GPS-EW-corrections] (HHId10, HHId10New, C, Cnew, S, Snew, CSew1, CSew2, RepSo, [NewC-S], RepCo, ModCo, NewCoDeg, NewCoMin, [Cew1-ew2], [ew1-ew2], 1EWr, 2EWr, 1EWrNew, 2EWrNew, 1deg, 1min, 1sec, 2deg, 2min, 2sec, Cew1decNew, Cew2decNew, ModSo, NewSoDeg, NewSoMin, gpse1, gpse2, gpse3, gpse4, gpse5, gpse6, Sew1decNew, Sew2decNew, 1NSr, 2NSr, Lc1Name, Lc1Code, LcName, LcCode)
SELECT [EW-NEW].Field1, [EW-NEW].Field2, [EW-NEW].Field3, [EW-NEW].Field4, [EW-NEW].Field5, [EW-NEW].Field6, [EW-NEW].Field7, [EW-NEW].Field8, [EW-NEW].Field9, [EW-NEW].Field10, [EW-NEW].Field11, [EW-NEW].Field12, [EW-NEW].Field13, [EW-NEW].Field14, [EW-NEW].Field15, [EW-NEW].Field16, [EW-NEW].Field17, [EW-NEW].Field18, [EW-NEW].Field19, [EW-NEW].Field20, [EW-NEW].Field21, [EW-NEW].Field22, [EW-NEW].Field23, [EW-NEW].Field24, [EW-NEW].Field25, [EW-NEW].Field26, [EW-NEW].Field27, [EW-NEW].Field28, [EW-NEW].Field29, [EW-NEW].Field30, [EW-NEW].Field31, [EW-NEW].Field32, [EW-NEW].Field33, [EW-NEW].Field34, [EW-NEW].Field35, [EW-NEW].Field36, [EW-NEW].Field37, [EW-NEW].Field38, [EW-NEW].Field39, [EW-NEW].Field40, [EW-NEW].Field41, [EW-NEW].Field42, [EW-NEW].Field43, [EW-NEW].Field44, [EW-NEW].Field45
FROM [EW-NEW];

CREATE NEW TABLE OF ALL CORRECTIONS

H. Create-table query: Q-Assemble-All-Corrections

SELECT DISTINCT [GPS-EW-corrections].HHId10, [GPS-EW-corrections].HHId10New, [GPS-EW-corrections].C, [GPS-EW-corrections].Cnew, [GPS-EW-corrections].S, [Q-Old-HHcoordinates].Sew1DecOld, [Q-Old-HHcoordinates].Sew2DecOld, [GPS-EW-corrections].Snew, [GPS-EW-corrections].RepSo, [GPS-EW-corrections].[NewC-S], [GPS-EW-corrections].RepCo, [GPS-EW-corrections].ModCo AS ModCoEw, [GPS-EW-corrections].NewCoDeg AS NewCoDegEw, [GPS-EW-corrections].NewCoMin AS

```
NewCoMinEw, [GPS-EW-corrections].[1EWr], [GPS-EW-corrections].[2EWr], [GPS-EW-
corrections].[1EWrNew], [GPS-EW-corrections].[2EWrNew], [GPS-EW-corrections].[1deg]
AS 1degEW, [GPS-EW-corrections].[1min] AS 1minEW, [GPS-EW-corrections].[1sec] AS
1secEW, [GPS-EW-corrections].[2deg] AS 2degEW, [GPS-EW-corrections].[2min] AS
2minEW, [GPS-EW-corrections].[2sec] AS 2secEW, [GPS-EW-corrections].Cew1decNew,
[GPS-EW-corrections].Cew2decNew, [GPS-EW-corrections].ModSo AS ModSoEW, [GPS-
EW-corrections].NewSoDeg AS NewSoDegEw, [GPS-EW-corrections].NewSoMin AS
NewSoMinEw, [GPS-EW-corrections].gpse1, [GPS-EW-corrections].gpse2, [GPS-EW-
corrections].gpse3, [GPS-EW-corrections].gpse4, [GPS-EW-corrections].gpse5, [GPS-EW-
corrections].gpse6, [GPS-EW-corrections].Sew1decNew, [GPS-EW-
corrections].Sew2decNew, [GPS-EW-corrections].Lc1Name AS Lc1NameEW, [GPS-EW-
corrections].Lc1Code AS Lc1CodeEW, [GPS-EW-corrections].LcName AS LcNameEW,
[GPS-EW-corrections].LcCode AS LcCodeEW, " " AS XXX, [GPS-NS-corrections].ModCo
AS ModCoNS, [GPS-NS-corrections].NewCoNS AS NewCoNSNS, [GPS-NS-
corrections].NewCoDeg AS NewCoDegNS, [GPS-NS-corrections].NewCoMin AS
NewCoMinNS, [GPS-NS-corrections].[1NSr], [GPS-NS-corrections].[2NSr], [GPS-NS-
corrections].[1NSrNew], [GPS-NS-corrections].[2NSrNEw], [GPS-NS-corrections].[1ns] AS
1nsNS, [GPS-NS-corrections].[1deg] AS 1degNS, [GPS-NS-corrections].[1min] AS
1minNS, [GPS-NS-corrections].[1sec] AS 1secNS, [GPS-NS-corrections].[2ns] AS 2nsNS,
[GPS-NS-corrections].[2deg] AS 2degNS, [GPS-NS-corrections].[2min] AS 2minNS, [GPS-
NS-corrections].[2sec] AS 2secNS, [GPS-NS-corrections].Cnsr1decNew, [GPS-NS-
corrections].Cnsr2decNew, [Q-Old-HHcoordinates].Snsr1DecOld, [Q-Old-
HHcoordinates].Snsr2DecOld, [GPS-NS-corrections].ModSo AS ModSoNS, [GPS-NS-
corrections].NewSoNS AS NewSoNSNs, [GPS-NS-corrections].NewSoDeg AS
NewSoDegNs, [GPS-NS-corrections].NewSoMin AS NewSoMinNs, [GPS-NS-
corrections].ns1, [GPS-NS-corrections].gpsns1, [GPS-NS-corrections].gpsns2, [GPS-NS-
corrections].gpsns3, [GPS-NS-corrections].ns2, [GPS-NS-corrections].gpsns4, [GPS-NS-
corrections].gpsns5, [GPS-NS-corrections].gpsns6, [GPS-NS-corrections].Snsr1decNew,
[GPS-NS-corrections].Snsr2decNew INTO [GPS-All-Corrections]
FROM ([GPS-EW-corrections] INNER JOIN [GPS-NS-corrections] ON [GPS-EW-
corrections].Snew = [GPS-NS-corrections].Snew) INNER JOIN [Q-Old-HHcoordinates] ON
[GPS-EW-corrections].S = [Q-Old-HHcoordinates].householdId;
```

QUERY TO CALCULATE OLD HH COORDINATES FROM ORIGINAL DATA

I. Select-query: Q-Old-HHCoordinates

```
SELECT DISTINCT [Table-SocioID10].HHId10, [Table-SocioID10].householdId,
[gpse1]+[gpse2]/60+[gpse3]/1000/60 AS Sew1DecOld,
[gpse4]+[gpse5]/60+[gpse6]/1000/60 AS Sew2DecOld, IIf([ns1]=1,1,-
1)*([gpsns1]+[gpsns2]/60+[gpsns3]/1000/60) AS Snsr1DecOld, IIf([ns2]=1,1,-
1)*([gpsns4]+[gpsns5]/60+[gpsns6]/1000/60) AS Snsr2DecOld
FROM [Table-SocioID10];
```

QUERIES TO COMPUTE MIN, MAX and RANGES OF HH COORDINATES

J. Select-query: Q-Decimal-Differences1

SELECT DISTINCT [GPS-EW-corrections].HHId10New, Count([GPS-EW-corrections].HHId10New) AS CountOfHHId10New, Min([GPS-EW-corrections].Cew1decNew) AS MinOfCew1decNew, Max([GPS-EW-corrections].Cew1decNew) AS MaxOfCew1decNew, Min([GPS-EW-corrections].Cew2decNew) AS MinOfCew2decNew, Max([GPS-EW-corrections].Cew2decNew) AS MaxOfCew2decNew, Min([GPS-EW-corrections].Sew1decNew) AS MinOfSew1decNew, Max([GPS-EW-corrections].Sew1decNew) AS MaxOfSew1decNew, Min([GPS-EW-corrections].Sew2decNew) AS MinOfSew2decNew, Max([GPS-EW-corrections].Sew2decNew) AS MaxOfSew2decNew, Min([GPS-NS-corrections].Cnsr1decNew) AS MinOfCnsr1decNew, Max([GPS-NS-corrections].Cnsr1decNew) AS MaxOfCnsr1decNew, Min([GPS-NS-corrections].Cnsr2decNew) AS MinOfCnsr2decNew, Max([GPS-NS-corrections].Cnsr2decNew) AS MaxOfCnsr2decNew, Min([GPS-NS-corrections].Snsr1decNew) AS MinOfSnsr1decNew, Max([GPS-NS-corrections].Snsr1decNew) AS MaxOfSnsr1decNew, Min([GPS-NS-corrections].Snsr2decNew) AS MinOfSnsr2decNew, Max([GPS-NS-corrections].Snsr2decNew) AS MaxOfSnsr2decNew
FROM [GPS-EW-corrections] RIGHT JOIN [GPS-NS-corrections] ON [GPS-EW-corrections].Snew = [GPS-NS-corrections].Snew
GROUP BY [GPS-EW-corrections].HHId10New;

K. Select-query: Q-Decimal-Differences2

SELECT [Q-Decimal-Differences1].HHId10New, [Q-Decimal-Differences1].CountOfHHId10New, [Q-Decimal-Differences1].MinOfSew1decNew AS f1, [Q-Decimal-Differences1].MaxOfSew1decNew AS f2, [f2]-[f1] AS HH1ewdiff, [Q-Decimal-Differences1].MinOfSew2decNew AS f3, [Q-Decimal-Differences1].MaxOfSew2decNew AS f4, [f4]-[f3] AS HH2ewdiff, [Q-Decimal-Differences1].MinOfSnsr1decNew AS f5, [Q-Decimal-Differences1].MaxOfSnsr1decNew AS f6, [f6]-[f5] AS HH1nsdiff, [Q-Decimal-Differences1].MinOfSnsr2decNew AS f7, [Q-Decimal-Differences1].MaxOfSnsr2decNew AS f8, [f8]-[f7] AS HH2nsdiff
FROM [Q-Decimal-Differences1];

L. Create-table-query: Q-Decimal-Differences3-EW

SELECT [Q-Decimal-Differences2].CountOfHHId10New AS NrHHs, [Q-Decimal-Differences2].HH1ewdiff, [Q-Decimal-Differences2].HH2ewdiff, [GPS-EW-corrections].HHId10, [GPS-EW-corrections].HHId10New, [GPS-EW-corrections].C, [GPS-EW-corrections].Cnew, [GPS-EW-corrections].S, [GPS-EW-corrections].Snew, [GPS-EW-corrections].CSew1, [GPS-EW-corrections].CSew2, [GPS-EW-corrections].RepSo, [GPS-

EW-corrections].[NewC-S], [GPS-EW-corrections].RepCo, [GPS-EW-corrections].ModCo, [GPS-EW-corrections].NewCoDeg, [GPS-EW-corrections].NewCoMin, [GPS-EW-corrections].[Cew1-ew2], [GPS-EW-corrections].[ew1-ew2], [GPS-EW-corrections].[1EWr], [GPS-EW-corrections].[2EWr], [GPS-EW-corrections].[1EWrNew], [GPS-EW-corrections].[2EWrNew], [GPS-EW-corrections].[1deg], [GPS-EW-corrections].[1min], [GPS-EW-corrections].[1sec], [GPS-EW-corrections].[2deg], [GPS-EW-corrections].[2min], [GPS-EW-corrections].[2sec], [GPS-EW-corrections].Cew1decNew, [GPS-EW-corrections].Cew2decNew, [GPS-EW-corrections].ModSo, [GPS-EW-corrections].NewSoDeg, [GPS-EW-corrections].NewSoMin, [GPS-EW-corrections].gpse1, [GPS-EW-corrections].gpse2, [GPS-EW-corrections].gpse3, [GPS-EW-corrections].gpse4, [GPS-EW-corrections].gpse5, [GPS-EW-corrections].gpse6, [GPS-EW-corrections].Sew1decNew, [GPS-EW-corrections].Sew2decNew, [GPS-EW-corrections].[1NSr], [GPS-EW-corrections].[2NSr], [GPS-EW-corrections].Lc1Name, [GPS-EW-corrections].Lc1Code, [GPS-EW-corrections].LcName, [GPS-EW-corrections].LcCode INTO [EW-diff]
FROM [Q-Decimal-Differences2] INNER JOIN [GPS-EW-corrections] ON [Q-Decimal-Differences2].HHId10New = [GPS-EW-corrections].HHId10New
ORDER BY [Q-Decimal-Differences2].HH1ewdiff DESC , [Q-Decimal-Differences2].HH2ewdiff DESC , [GPS-EW-corrections].Snew DESC;

M. Create-table-query: Q-Decimal-Differences3-NS

SELECT [Q-Decimal-Differences2].CountOfHHId10New AS NrHHs, [Q-Decimal-Differences2].HH1nsdiff, [Q-Decimal-Differences2].HH2nsdiff, [GPS-NS-corrections].HHId10, [GPS-NS-corrections].HHId10New, [GPS-NS-corrections].C, [GPS-NS-corrections].Cnew, [GPS-NS-corrections].S, [GPS-NS-corrections].Snew, [GPS-NS-corrections].CSns1, [GPS-NS-corrections].CSns2, [GPS-NS-corrections].RepSo, [GPS-NS-corrections].[NewC-S], [GPS-NS-corrections].RepCo, [GPS-NS-corrections].ModCo, [GPS-NS-corrections].NewCoNS, [GPS-NS-corrections].NewCoDeg, [GPS-NS-corrections].NewCoMin, [GPS-NS-corrections].[Cns1-ns2], [GPS-NS-corrections].[ns1-ns2], [GPS-NS-corrections].[1NSr], [GPS-NS-corrections].[2NSr], [GPS-NS-corrections].[1NSrNew], [GPS-NS-corrections].[2NSrNEw], [GPS-NS-corrections].[1ns], [GPS-NS-corrections].[1deg], [GPS-NS-corrections].[1min], [GPS-NS-corrections].[1sec], [GPS-NS-corrections].[2ns], [GPS-NS-corrections].[2deg], [GPS-NS-corrections].[2min], [GPS-NS-corrections].[2sec], [GPS-NS-corrections].Cnsr1decNew, [GPS-NS-corrections].Cnsr2decNew, [GPS-NS-corrections].ModSo, [GPS-NS-corrections].NewSoNS, [GPS-NS-corrections].NewSoDeg, [GPS-NS-corrections].NewSoMin, [GPS-NS-corrections].ns1, [GPS-NS-corrections].gpsns1, [GPS-NS-corrections].gpsns2, [GPS-NS-corrections].gpsns3, [GPS-NS-corrections].ns2, [GPS-NS-corrections].gpsns4, [GPS-NS-corrections].gpsns5, [GPS-NS-corrections].gpsns6, [GPS-NS-corrections].Snsr1decNew, [GPS-NS-corrections].Snsr2decNew, [GPS-NS-corrections].East1stReading, [GPS-NS-corrections].East2ndReading, [GPS-NS-corrections].Lc1Name, [GPS-NS-corrections].Lc1Code, [GPS-NS-corrections].LcName, [GPS-NS-corrections].LcCode INTO [NS-diff]

```
FROM [Q-Decimal-Differences2] INNER JOIN [GPS-NS-corrections] ON [Q-Decimal-
Differences2].HHId10New = [GPS-NS-corrections].HHId10New
ORDER BY [Q-Decimal-Differences2].HH1nsdiff DESC , [Q-Decimal-
Differences2].HH2nsdiff DESC , [GPS-NS-corrections].Snew DESC;
```