Data Cleaning Pre-Migration

1. replace \" with " in all tables with NAME columns
2. Create query to determine individuals with only 1 name (no middle or last name. Can query imported OpenHDS data searching for individuals with firstName != null middleName = null, last name = null)
3. create query to determine birthdate impossible
4. SocialGroup has head with extId UNK in DSS Data. Not accepted by OpenHDS. Some of these individuals DO have PERMID’s in the INDIVIDUAL table so should be updated. There are records with Q as headId, NA and UNK. Data Managers need to fix this. Those that cannot be fixed should be listed out and given arbitrary headId’s to facilitate the migration. However when the next enumeration round comes, questions that HAVE to be re-asked are “What is your firstName, middleName and surname? As well as “Who is the head of house here?”This will then ensure we have accurate naming as well as accurate headOfHouse information.

## INDIVIDUAL

ON DSS Data alter table thus (after data cleaning has been done):

1. added a field “approximate” type INT default = 1 that maps to dobAspect
2. add field instanceId type = varchar(50). Then update instanceId =””(empty string)
3. add field FIELDWORKER varchar(6) default = ‘FWAD1’
4. add field processed\_by\_mirth, default = 0
5. add field id type INT, autoIncrement and primary key

ZERO PADDING:

update individual set extId= concat(left(extId,**3**),'00',right(extId,**7**)) where extId!='UNK'

## SOCIAL GRP

ON DSS Data alter table thus (after data cleaning has been done):

1. add field id type INT, autoIncrement and primary key
2. add field FIELDWORKER varchar(6) default = ‘FWAD1’
3. add field processed\_by\_mirth,TYPE = CHAR,default = 0
4. add field instanceId type = varchar(50). Then update instanceId =””(empty string)

update socialgroup set extId= concat(left(extId,**3**),'00',right(extId,**6**))

## LOCATION

1. add field id type INT, autoIncrement and primary key
2. add field FIELDWORKER varchar(6) default = ‘FWAD1’
3. add field processed\_by\_mirth, default = 0
4. add field instanceId type = varchar(50). Then update instanceId =””(empty string)
5. ad field latitude type = INT default 0
6. add field longitude type=INT default 0
7. add field accuracy type = INT default 0
8. add field subvillageId VARCHAR (3)

ALTER TABLE hdss.`inmigration`

ADD id **INT** AUTO\_INCREMENT NOT NULL AFTER STATUS\_DAT,

ADD FIELDWORKER **VARCHAR**(**6**),

ADD processed\_by\_mirth CHAR ASCII DEFAULT '0',

ADD instanceId **VARCHAR**(**50**),

ADD PRIMARY KEY (id);

update inmigration SET FIELDWORKER='FWAD1';

ZEROPADDING:  
update location set extId= concat(left(extId,**3**),'00',right(extId,**4**))

# scripts for location subvillageId:

UPDATE hdss.location l,openhds\_old.location o,openhds\_old.locationhierarchy h

SET o.locationLevel\_uuid=h.uuid where l.LOCATIONID=o.extId and o.locationLevel\_uuid is null

and h.extId=l.subvillageId

update location l set l.subvillageId = 'MLB'

WHERE JINA\_LA\_KI = 'MLABANI\_B' OR JINA\_LA\_KI = 'MLABANI \'B\'' OR JINA\_LA\_KI = 'MLABANI B' OR JINA\_LA\_KI = 'MLABANI \"B\"'

OR JINA\_LA\_KI = 'MLABANI B' OR JINA\_LA\_KI = 'MLABANI \'B\'' OR JINA\_LA\_KI = 'MLABANI \'B\"'

OR JINA\_LA\_KI = 'MLABANI \"B\'' OR JINA\_LA\_KI = 'MLABANIB' OR JINA\_LA\_KI = 'MLABANI\'B\'' OR JINA\_LA\_KI = 'MLABANI \'B\"'

OR JINA\_LA\_KI = 'MLABANI \'B\'\'' OR JINA\_LA\_KI LIKE 'MLA B' OR JINA\_LA\_KI LIKE 'MLA B' OR JINA\_LA\_KI LIKE 'MLA "B"';

update location l set l.subvillageId = 'MLA'

WHERE JINA\_LA\_KI = 'MLABANI\_A' OR JINA\_LA\_KI = 'MLABANI \'A\'' OR JINA\_LA\_KI = 'MLABANI A' OR JINA\_LA\_KI = 'MLABANI \"A\"'

OR JINA\_LA\_KI = 'MLABANI A' OR JINA\_LA\_KI = 'MLABANI \'A\'' OR JINA\_LA\_KI = 'MLABANI :A:' OR JINA\_LA\_KI = 'MLABANI \'A\"'

OR JINA\_LA\_KI = 'MLABANI \"A\'' OR JINA\_LA\_KI = 'MLABANIA' OR JINA\_LA\_KI = 'MLABANI\'A\'' OR JINA\_LA\_KI = 'MLABANI \'A\"'

OR MWENYEKIT\_ LIKE '%MATOLA%';

update location l set l.subvillageId = 'MIN'

WHERE JINA\_LA\_KI = 'MINARANI';

update location l set l.subvillageId = 'LIP'

WHERE JINA\_LA\_KI = 'LIPANGALALA';

update location l set l.subvillageId = 'IFA'

WHERE JINA\_LA\_KI LIKE '%IFAKARA%' OR JINA\_LA\_KI LIKE 'IFA' OR JINA\_LA\_KI LIKE 'IFAKAR';

update location l set l.subvillageId = 'KIU'

WHERE JINA\_LA\_KI LIKE '%KIUNGANI%';

update location l set l.subvillageId = 'MAN'

WHERE JINA\_LA\_KI LIKE '%MANGWALE%';

update location l set l.subvillageId = 'NDU'

WHERE JINA\_LA\_KI LIKE '%NDUNA%';

update location l set l.subvillageId = 'KIY'

WHERE JINA\_LA\_KI LIKE '%KIYONJO%' OR JINA\_LA\_KI LIKE '%KIONJO%';

update location l set l.subvillageId = 'KAA'

WHERE JINA\_LA\_KI LIKE '%KATINDIUKA A%' OR JINA\_LA\_KI LIKE 'KATINDIUKA "A"' OR JINA\_LA\_KI LIKE 'KATINDIUKA \'A\''

OR JINA\_LA\_KI LIKE 'KATINDIUKA\_A' OR JINA\_LA\_KI LIKE 'KAT A';

update location l set l.subvillageId = 'KAB'

WHERE JINA\_LA\_KI LIKE 'KATINDIUKA B' OR JINA\_LA\_KI LIKE 'KATINDIUKA \'B\'' OR JINA\_LA\_KI LIKE 'KATINDIUKA "B"'

OR JINA\_LA\_KI LIKE 'KATINDIUKA \_B' OR JINA\_LA\_KI LIKE 'KATINDIUKA B' OR JINA\_LA\_KI LIKE 'KATINDIUKA \'B\"' OR JINA\_LA\_KI

LIKE 'KATINDIUKA\_B';

update location l set l.subvillageId = 'KAC'

WHERE JINA\_LA\_KI LIKE 'KATINDIUKA C' OR JINA\_LA\_KI LIKE 'KATINDIUKA \'C\'' OR JINA\_LA\_KI LIKE 'KATINDIUKA "C"'

OR JINA\_LA\_KI LIKE 'KATINDIUKA \_C' OR JINA\_LA\_KI LIKE 'KATINDIUKA C' OR JINA\_LA\_KI LIKE 'KATINDIUKA \'C\"' OR JINA\_LA\_KI

LIKE 'KATINDIUKA\_C';

update location l set l.subvillageId = 'VIS'

WHERE JINA\_LA\_KI LIKE '%VIWANJ%' OR JINA\_LA\_KI LIKE 'V60' OR JINA\_LA\_KI LIKE 'V.60' OR JINA\_LA\_KI LIKE 'V.GO'

OR JINA\_LA\_KI LIKE 'V/SITINI' OR JINA\_LA\_KI LIKE 'VIS 60' OR JINA\_LA\_KI LIKE 'VIWNAJA SITINI';

update location l set l.subvillageId = 'MHO'

WHERE JINA\_LA\_KI LIKE 'M%HOLA' or JINA\_LA\_KI like 'MH%LA' OR JINA\_LA\_KI LIKE 'MOLA';

update location l set l.subvillageId = 'VIW'

WHERE JINA\_LA\_KI LIKE 'VIWANDANI';

update location l set l.subvillageId = 'KIY'

WHERE JINA\_LA\_KI LIKE 'KIYONGWILE';

update location l set l.subvillageId = 'MAN'

WHERE JINA\_LA\_KI = 'MANGWELE' OR JINA\_LA\_KI LIKE 'MNGWALE';

update location l set l.subvillageId = 'LIH'

WHERE JINA\_LA\_KI LIKE 'LIHAMI';

update location l set l.subvillageId = 'MKU'

WHERE JINA\_LA\_KI LIKE 'MKUYA';

SOCIAL GROUP:

Update social groups not transferred so that Mirth reprocesses them:

update socialgroup set processed\_by\_mirth=false where SOCIALGPID NOT IN (Select extId from openhds.socialgroup);

update socialgroup s,individual e set HEADID=e.INDIVIDID where e.NAME=s.NAME;

FOR DUPLICATES:

select count(\*),extId from openhds.location group by extId order by **1** desc

RESIDENCY

GOTCHAS: Make sure that date formats are DATE types and not VARCHAR on imported residency table

After migrate

update residency set endType='NA' where endType='';

update residency set endType='DTH' where individual\_uuid in (select individual\_uuid from death )

and endType='NA';

update residency r,death d set endDate = d.deathDate where r.endType='DTH' and r.endDate is null and r.individual\_uuid=d.individual\_uuid;

update residency set endType='OMG' where endType = 'NA' and startType='ENU' and individual\_uuid in (select individual\_uuid from outmigration);

update residency r,outmigration d set endDate = d.recordedDate where r.endType='OMG' and r.endDate is null

and r.individual\_uuid=d.individual\_uuid;

MEMBERSHIP

* Remove \ (backslashes) in the source data

update membership set relationship = **1** where RLTN\_HEAD in ('HD','ONE');

update membership set relationship = **2** where RLTN\_HEAD in ('WIF');

update membership set relationship = **3** where RLTN\_HEAD in ('CHD');

update membership set relationship = **4** where RLTN\_HEAD in ('BRO','SIS');

update membership set relationship = **5** where RLTN\_HEAD in ('PAR');

update membership set relationship = **8** where RLTN\_HEAD in ('OT','GDP','OTH');

update membership set relationship = **9** where RLTN\_HEAD in ('NK','UNK');

To find members who do not exist

select \* from membership where INDIVIDID not in (select INDIVIDID from individual);

* To find socialGroups that do not exist

select \* from membership where SOCIALGPID not in (select SOCIALGPID from socialgroup);

OUTMIGRATION

* Remove \ (backslashes) in the source data
* To run after the outmigration channel

update openhds.outmigration o, openhds.residency r set o.residency\_uuid= r.uuid where

o.recordedDate= r.endDate and r.endType ='OMG' and r.individual\_uuid=o.individual\_uuid;

DEATH

update death set PLACE\_OF\_DEATH = 'HOME' where DTHPLACE = **1**;

update death set PLACE\_OF\_DEATH = 'HOSPITAL' where DTHPLACE = **2**;

update death set PLACE\_OF\_DEATH = 'OTHER' where DTHPLACE = **3**;

update death d set d.OTHDTHCAUS = null;

ALTER TABLE hdss.death

CHANGE DTHCAUSE DTHCAUSE **VARCHAR**(**11**);

update death d set d.DTHCAUSE = ‘NA’;

To see DEATH rows NOT transferred:

select \* from death t where t.INDIVIDID not in (select extId from openhds.individual i, openhds.death d where i.uuid=d.individual\_uuid)

RESIDENCY

Do at very beginning of residency migration

Update residency set EEVENTTYPE = 'OMG' where EEVENTTYPE = 'EXT';

update residency r set r.SEVENTTYPE='IMG' where r.SEVENTTYPE='ENT';

update residency r set r.SEVENTTYPE='ENU' where r.SEVENTTYPE='BIR';

Extract the residencies where startdate=enddate

select \* from residency where datediff( SDATE, EDATE) =**0**;

Extract residencies with wrong LocationID

select \* from residency where LOCATIONID not in (select LOCATIONID from location);

SELECT residency.\* FROM `residency`, individual WHERE  `individual\_uuid`=individual.uuid and `location\_uuid` is null

check residencies  
SELECT count( individual\_uuid ) num , `individual\_uuid` , `location\_uuid` , `startDate` , `startType` , `endType`,endDate  
FROM `residency`  
WHERE endDate IS NULL  
GROUP BY `individual\_uuid` , `location\_uuid` , `startDate` , `startType` , `endType`,endDate  
ORDER BY 1 DESC

SELECT count( individual\_uuid ) num , `individual\_uuid` , `socialGroup\_uuid` , `startDate` , `startType` , `endType`,endDate,bIsToA  
FROM `membership`  
WHERE endDate IS NULL  
GROUP BY `individual\_uuid` , `socialGroup\_uuid` , `startDate` , `startType` , `endType`,endDate  
ORDER BY 1 DESC, 2 asc

SELECT count( socialGroup\_uuid ) num , `socialGroup\_uuid` , `startDate` , `startType` , `endType`,endDate,bIsToA  
FROM `membership`  
WHERE endDate IS NULL and bIsToA=1  
GROUP BY `socialGroup\_uuid` , `startDate` , `startType` , `endType`,endDate,bIsToA  
ORDER BY 1 DESC, 2 asc

RELATIONSHIP

Extract from the table the individuals not in the individual table:

select \* from relationship where INDIVIDID not in (select i.INDIVIDID from individual i);

select \* from relationship where INDIVIDID2 not in (select i.INDIVIDID from individual i);

Extract list of Invalid Relationships

1. Wife is a male

select \* from relationship where INDIVIDID in (select i.INDIVIDID from individual i where i.GENDER='M');

1. Husband is Female

select \* from relationship where INDIVIDID2 in (select i.INDIVIDID from individual i where i.GENDER='F';

1. Husband below 14

select \* from relationship where INDIVIDID2 in (select i.INDIVIDID from individual i where DATEDIFF( CURDATE(), i.BIRTH\_DATE)<**5110**);

1. Wife below 12

select \* from relationship where INDIVIDID in (select i.INDIVIDID from individual i where DATEDIFF( CURDATE(), i.BIRTH\_DATE)<**4380**);

1. Start date of relationship in the future

select \* from relationship where datediff( SDATE, curdate())>0)

1. End date of relationship in the future

select \* from relationship where datediff( EDATE, curdate())>0)

BIRTH:

Do not migrate as this info is already in the individual table

MEMBERSHIP:

Individuals in member table not in INDIVIDUAL table:

select \* from membership where INDIVIDID not in (select INDIVIDID from individual);

Individuals not in a social group:

select \* from membership where SOCIALGPID not in (select SOCIALGPID from socialgroup);

update membership m, death d set m.endDate=d.deathDate, m.endType='DTH' where endDate is null and d.individual\_uuid =m.individual\_uuid;

OUTMIGRATION

UPDATE outmigration set reason=’AGRICULTURE’ WHERE OUTMGREASON=1 (X6 FOR THE OTHER REASONS)

update outmigration o set o.OUTLOCATIO='UNKNOWN' WHERE o.INTERNAL='N'

Script for creating indexes:

ALTER TABLE hdss.observation

ADD UNIQUE INDEX observe\_id\_index (OBSERVEID);

Check those not migrated:

select \* from outmigration where OBSERVEID not in

(select extId from openhds\_old.visit v,openhds\_old.outmigration o where v.uuid=o.visit\_uuid)

VISIT/OBSERVATION

update observation o set o.ROUND=**9999** where o.ROUND=**0**;

INMIGRATION

ALTER TABLE hdss.inmigration

ADD reasonForInmig **VARCHAR**(**20**) AFTER instanceId;

update inmigration d set d.reasonForInmig = 'AGRICULTURE' WHERE IMGREASON=**1**;

update inmigration d set d.reasonForInmig = 'ANIMAL HUSBANDRY' WHERE IMGREASON=**2**;

update inmigration d set d.reasonForInmig = 'MARITAL CHANGE' WHERE IMGREASON=**3**;

update inmigration d set d.reasonForInmig = 'EDUCATION/SCHOOL' WHERE IMGREASON=**4**;

update inmigration d set d.reasonForInmig = 'WORK' WHERE IMGREASON=**5**;

update inmigration d set d.reasonForInmig = 'OTHER' WHERE IMGREASON=**6**;

update inmigration i set i.REGION\_NAM='UNKNOWN' where i.REGION\_NAM is null;

Before running channel, turn off residency foreign key constraint in inmigration (??) and alter table to reflect that this column can be null.

After running channel, do (because channel purposefully does not set the residency):

update openhds\_old.inmigration i, openhds\_old.residency r set i.residency\_uuid= r.uuid where

i.recordedDate= r.startDate and r.startType ='IMG' and r.individual\_uuid=i.individual\_uuid and i.residency\_uuid is null;

CHECK ALL INMIGS IMPORTED:  
create table test as select concat(v.extId,p.extId,i.recordedDate) as id from openhds\_old.visit v,openhds\_old.inmigration i,openhds\_old.individual p where v.uuid= i.visit\_uuid

and p.uuid=i.individual\_uuid;

select \* from inmigration2 where concat(OBSERVEID,INDIVIDID,**DATE**) not in (select id from test);

PREGNANCY OUTCOME

For post migration check of what has NOT been migrated, we need a unique field to check against. In this case, that is EVENTID. Therefore add column eventid to openhds.pregnancyoutcome.

After running the channel, find rows not transferred to openhds:

select OBSERVEID FROM pregnancyoutcome where OBSERVEID not in

(select extId from openhds.visit v,openhds.pregnancyoutcome po where v.uuid=po.visit\_uuid)

PREGNANCY OBSERVATION

* Create column pregnancy\_due\_date where due date is calculated from either last period (either from clinic card or from interviewer)
* Check rows not migrated

create table reconcile as

(select concat(v.extId,i.extId) as id

from rural\_openhds.pregnancyobservation q,rural\_openhds.visit v,rural\_openhds.individual i

where q.visit\_uuid=v.uuid and q.mother\_uuid=i.uuid);

select \* from pregnancy\_observation po where concat(po.OBSERVEID,po.INDIVIDID)

not in (select id from reconcile);

# HRS1-HRS2 DATA

INDIVIDUAL

1. ALTER TABLE individual

ADD id **INT** AUTO\_INCREMENT NOT NULL AFTER STATUS\_DAT,

ADD approximate INT,

ADD FIELDWORKER **VARCHAR**(**6**),

ADD processed\_by\_mirth CHAR ASCII DEFAULT '0',

ADD instanceId **VARCHAR**(**50**),

ADD PRIMARY KEY (id);

update inmigration SET FIELDWORKER='FWAD1';

LOCATION

ALTER TABLE location

ADD id **INT** AUTO\_INCREMENT NOT NULL AFTER STATUS\_DAT,

ADD FIELDWORKER **VARCHAR**(**6**),

ADD processed\_by\_mirth CHAR ASCII DEFAULT '0',

ADD instanceId **VARCHAR**(**50**),

ADD PRIMARY KEY (id);

update inmigration SET FIELDWORKER='FWAD1';

update location SET subvillageId = 'GOA' where KITONGOJI='GODAWN A';

update location SET subvillageId = 'GOB' where KITONGOJI='GODAWN B';

update location SET subvillageId = 'GOC' where KITONGOJI='GODAWN CHINI';

update location SET subvillageId = 'GOJ' where KITONGOJI='GODAWN JUU';

update location SET subvillageId = 'BMA' where KITONGOJI='BOMA MZINGA A';

update location SET subvillageId = 'ICH' where KITONGOJI='ICHIMA';

update location SET subvillageId = 'ALA' where KITONGOJI='ALABAMA';

update location SET subvillageId = 'BMB' where KITONGOJI='BOMA MZINGA B';

update location SET subvillageId = 'BMB' where KITONGOJI='BOMAMZINGA B';

update location SET subvillageId = 'ARA' where KITONGOJI='ARAMBA';

update location SET subvillageId = 'BUT' where KITONGOJI='BUTIAMA';

update location SET subvillageId = 'CHI' where KITONGOJI='CHIKAGO';

update location SET subvillageId = 'CHW' where KITONGOJI='CHIWACHIWA';

update location SET subvillageId = 'GEZ' where KITONGOJI='GEZAULOLE';

update location SET subvillageId = 'HOS' where KITONGOJI='HOSPITALI';

update location SET subvillageId = 'IDA' where KITONGOJI='IDANDU';

update location SET subvillageId = 'IDM' where KITONGOJI='IDANDU MSITUNAM';

update location SET subvillageId = 'IDM' where KITONGOJI='IDANDU MSITU WA';

update location SET subvillageId = 'IDL' where KITONGOJI='IDULIKE';

update location SET subvillageId = 'IDU' where KITONGOJI='IDUNDA';

update location SET subvillageId = 'IGM' where KITONGOJI='IGIA MBONGOLO';

update location SET subvillageId = 'IGH' where KITONGOJI='IGIA HEKANANE';

update location SET subvillageId = 'IGI' where KITONGOJI='IGIMA';

update location SET subvillageId = 'IGO' where KITONGOJI='IGOTA';

update location SET subvillageId = 'IGK' where KITONGOJI='IGUMBIRO KATI';

update location SET subvillageId = 'IGU' where KITONGOJI='IGUNDA';

update location SET subvillageId = 'IJI' where KITONGOJI='IJIA';

update location SET subvillageId = 'IJC' where KITONGOJI='IJIA CHUGUNI';

update location SET subvillageId = 'IJH' where KITONGOJI='IJIA HEKANANE';

update location SET subvillageId = 'IJS' where KITONGOJI='IJIA SHAMBA LA';

update location SET subvillageId = 'IJC' where KITONGOJI='IJIACHUGUUNI';

update location SET subvillageId = 'IKE' where KITONGOJI='IKELA' or KITONGOJI='IKERA' ;

update location SET subvillageId = 'KIK' where KITONGOJI='KIKOLEKO' or KITONGOJI='IKOLEKO';

update location SET subvillageId = 'IKU' where KITONGOJI='IKULE';

update location SET subvillageId = 'IKA' where KITONGOJI='IKUNGUA';

update location SET subvillageId = 'ILA' where KITONGOJI='ILANGALI';

update location SET subvillageId = 'ILO' where KITONGOJI='ILOLE';

update location SET subvillageId = 'IMW' where KITONGOJI='IMWAGA';

update location SET subvillageId = 'IPE' where KITONGOJI='IPERA';

update location SET subvillageId = 'IPN' where KITONGOJI='IPERANGOMAMBILI';

update location SET subvillageId = 'IRA' where KITONGOJI='IRAGUA KATI';

update location SET subvillageId = 'ISA' where KITONGOJI='ISAGO';

update location SET subvillageId = 'IKW' where KITONGOJI='ISAGO KWAMASHAR';

update location SET subvillageId = 'ITO' where KITONGOJI='ITOBANILO';

update location SET subvillageId = 'ITA' where KITONGOJI='ITONGOA A';

update location SET subvillageId = 'ITB' where KITONGOJI='ITONGOA B';

update location SET subvillageId = 'ITU' where KITONGOJI='ITOWU';

update location SET subvillageId = 'IYU' where KITONGOJI='IYUNDU';

update location SET subvillageId = 'KAL' where KITONGOJI='KALIAGOGO';

update location SET subvillageId = 'KAI' where KITONGOJI='KALIJINGULA';

update location SET subvillageId = 'KAN' where KITONGOJI='KANYAGA TWENDE';

update location SET subvillageId = 'KIB' where KITONGOJI='KIBUGUMO-MIOMBO';

update location SET subvillageId = 'KIU' where KITONGOJI='KIBURUBUTU';

update location SET subvillageId = 'KCH' where KITONGOJI='KICHANGANI KATI';

update location SET subvillageId = 'KID' where KITONGOJI='KIDETE';

update location SET subvillageId = 'KIG' where KITONGOJI='KIDUGALO';

update location SET subvillageId = 'KIG' where KITONGOJI='KIDUGALO MKONYA';

update location SET subvillageId = 'KIG' where KITONGOJI='KIDUGALOKIPALAN';

update location SET subvillageId = 'ILU' where KITONGOJI='ILUNGUSHA';

update location SET subvillageId = 'KIK' where KITONGOJI='KIKULUKUTU KARI';

update location SET subvillageId = 'KIY' where KITONGOJI='KIKUYU';

update location SET subvillageId = 'KIN' where KITONGOJI='KINGURUNGURU';

update location SET subvillageId = 'KNU' where KITONGOJI='KININGU';

update location SET subvillageId = 'KIP' where KITONGOJI='KIPALANGANGA';

update location SET subvillageId = 'KGU' where KITONGOJI='KIPINGU';

update location SET subvillageId = 'KIM' where KITONGOJI='KISAKI MBALI';

update location SET subvillageId = 'KIS' where KITONGOJI='KISEGESE';

update location SET subvillageId = 'KGA' where KITONGOJI='KISIGARA';

update location SET subvillageId = 'KSW' where KITONGOJI='KISIWANI';

update location SET subvillageId = 'KWA' where KITONGOJI='KISWAMBA';

update location SET subvillageId = 'KIT' where KITONGOJI='KITETE';

update location SET subvillageId = 'KUN' where KITONGOJI='KUNGWA';

update location SET subvillageId = 'LIB' where KITONGOJI='LIBARATULA';

update location SET subvillageId = 'LIG' where KITONGOJI='LIGALAUKA';

update location SET subvillageId = 'LIT' where KITONGOJI='LITULANYATI';

update location SET subvillageId = 'LON' where KITONGOJI='LONDON';

update location SET subvillageId = 'LDO' where KITONGOJI='LONDO';

update location SET subvillageId = 'LUK' where KITONGOJI='LUKOLONGO';

update location SET subvillageId = 'LKA' where KITONGOJI='LUKULA';

update location SET subvillageId = 'LUP' where KITONGOJI='LUPIRO KATI';

update location SET subvillageId = 'MAD' where KITONGOJI='MADIBIRA';

update location SET subvillageId = 'MAE' where KITONGOJI='MAENDELEO';

update location SET subvillageId = 'MAF' where KITONGOJI='MAFINJI';

update location SET subvillageId = 'MAG' where KITONGOJI='MAGEREZA';

update location SET subvillageId = 'MAH' where KITONGOJI='MAHONGOLE';

update location SET subvillageId = 'MAN' where KITONGOJI='MANGALA';

update location SET subvillageId = 'MAJ' where KITONGOJI='MANJOLE';

update location SET subvillageId = 'MAT' where KITONGOJI='MATINGI';

update location SET subvillageId = 'MBA' where KITONGOJI='MBANGAIPUTI';

update location SET subvillageId = 'MBS' where KITONGOJI='MBASA';

update location SET subvillageId = 'MBG' where KITONGOJI='MBASA GANJA';

update location SET subvillageId = 'MBM' where KITONGOJI='MBASA MDIBULE';

update location SET subvillageId = 'MMI' where KITONGOJI='MBASA MIPINGUNI';

update location SET subvillageId = 'MMP' where KITONGOJI='MBASA MPANDE';

update location SET subvillageId = 'MBE' where KITONGOJI='MBEMBE';

update location SET subvillageId = 'MKL' where KITONGOJI='MBEMBE KWA LISI';

update location SET subvillageId = 'MBC' where KITONGOJI='MBENJA CENTRE';

update location SET subvillageId = 'MBI' where KITONGOJI='MBINGU';

update location SET subvillageId = 'MBO' where KITONGOJI='MBOGOLO';

update location SET subvillageId = 'MBU' where KITONGOJI='MBUYUNI';

update location SET subvillageId = 'MCH' where KITONGOJI='MCHOMBE';

update location SET subvillageId = 'MDA' where KITONGOJI='MDABA';

update location SET subvillageId = 'MDI' where KITONGOJI='MDIBULE';

update location SET subvillageId = 'MGU' where KITONGOJI='MGUDENI';

update location SET subvillageId = 'MHI' where KITONGOJI='MHIKILANIKI';

update location SET subvillageId = 'MIA' where KITONGOJI='MIALE';

update location SET subvillageId = 'MYC' where KITONGOJI='MIALE YA CHINI';

update location SET subvillageId = 'MCA' where KITONGOJI='MICHEKENI A';

update location SET subvillageId = 'MCB' where KITONGOJI='MICHEKENI B';

update location SET subvillageId = 'MIE' where KITONGOJI='MIEMBENI';

update location SET subvillageId = 'MIK' where KITONGOJI='MIKINDANI';

update location SET subvillageId = 'MKI' where KITONGOJI='MIKOROSHINI';

update location SET subvillageId = 'MIL' where KITONGOJI='MILOLA';

update location SET subvillageId = 'MIJ' where KITONGOJI='MILOLA JUU';

update location SET subvillageId = 'MLK' where KITONGOJI='MILOLA KATI';

update location SET subvillageId = 'MIN' where KITONGOJI='MINAZINI';

update location SET subvillageId = 'MIW' where KITONGOJI='MIWANGANI';

update location SET subvillageId = 'MWA' where KITONGOJI='MIWANGANI A';

update location SET subvillageId = 'MWB' where KITONGOJI='MIWANGANI B';

update location SET subvillageId = 'MIY' where KITONGOJI='MIYALE';

update location SET subvillageId = 'MYI' where KITONGOJI='MIYOMBONI';

update location SET subvillageId = 'MJI' where KITONGOJI='MJIMWEMA';

update location SET subvillageId = 'MKA' where KITONGOJI='MKANGAWALO';

update location SET subvillageId = 'MKU' where KITONGOJI='MKULA';

update location SET subvillageId = 'MKS' where KITONGOJI='MKUSI';

update location SET subvillageId = 'MKW' where KITONGOJI='MKWAJUNI';

update location SET subvillageId = 'MLI' where KITONGOJI='MLIMANI';

update location SET subvillageId = 'MNG' where KITONGOJI='MNGETA';

update location SET subvillageId = 'MNY' where KITONGOJI='MNYUAMACHI';

update location SET subvillageId = 'MPM' where KITONGOJI='MPANDE MBASA';

update location SET subvillageId = 'MSA' where KITONGOJI='MSALAKA';

update location SET subvillageId = 'MSE' where KITONGOJI='MSESELE';

update location SET subvillageId = 'MSH' where KITONGOJI='MSHIKAMANO';

update location SET subvillageId = 'MSO' where KITONGOJI='MSOKERO';

update location SET subvillageId = 'MSU' where KITONGOJI='MSUMBIJI';

update location SET subvillageId = 'MJA' where KITONGOJI='MSUMBIJI A';

update location SET subvillageId = 'MJB' where KITONGOJI='MSUMBIJI B';

update location SET subvillageId = 'MSC' where KITONGOJI='MSUMBIJI C';

update location SET subvillageId = 'MBC' where KITONGOJI='MSUMBIJI CHINI';

update location SET subvillageId = 'MSJ' where KITONGOJI='MSUMBIJI JUU';

update location SET subvillageId = 'MTA' where KITONGOJI='MTAMBULE';

update location SET subvillageId = 'MTY' where KITONGOJI='MTYANGIMBOLE';

update location SET subvillageId = 'NAK' where KITONGOJI='NAKAGULU';

update location SET subvillageId = 'NAK' where KITONGOJI='NAKAGURU';

update location SET subvillageId = 'NAM' where KITONGOJI='NAMARUMBO';

update location SET subvillageId = 'NYA' where KITONGOJI='NAMBAYA';

update location SET subvillageId = 'NMA' where KITONGOJI='NAMHANGA';

update location SET subvillageId = 'NMA' where KITONGOJI='NAMHANGA LUBELE';

update location SET subvillageId = 'NMA' where KITONGOJI='NAMHANGA STAND;

update location SET subvillageId = 'NMA' where KITONGOJI='NAMHANGALONDONI';

update location SET subvillageId = 'NAA' where KITONGOJI='NAMWAWALA A';

update location SET subvillageId = 'NAB' where KITONGOJI='NAMWAWALA B';

update location SET subvillageId = 'NDO' where KITONGOJI='NDOLO';

update location SET subvillageId = 'NDU' where KITONGOJI='NDUNDU';

update location SET subvillageId = 'NGA' where KITONGOJI='NGAI';

update location SET subvillageId = 'NJA' where KITONGOJI='NGAJENGWA';

update location SET subvillageId = 'NGO' where KITONGOJI='NGONDOLA';

update location SET subvillageId = 'NGM' where KITONGOJI='NGAJIMA';

update location SET subvillageId = 'NVA' where KITONGOJI='NGAVALYA';

update location SET subvillageId = 'NGO' where KITONGOJI='NGOMAMBILI';

update location SET subvillageId = 'NJG' where KITONGOJI='NJAGI';

update location SET subvillageId = 'NYA' where KITONGOJI='NYAKADETE';

update location SET subvillageId = 'NYU' where KITONGOJI='NYAKAGULU';

update location SET subvillageId = 'NYU' where KITONGOJI='NYAKAGURU';

update location SET subvillageId = 'POP' where KITONGOJI='POPOTO';

update location SET subvillageId = 'RAM' where KITONGOJI='RAMBA';

update location SET subvillageId = 'SAN' where KITONGOJI='SANGULI';

update location SET subvillageId = 'SNJ' where KITONGOJI='SANJE';

update location SET subvillageId = 'SLS' where KITONGOJI='SHAMBA LA SABA';

update location SET subvillageId = 'SYN' where KITONGOJI='STENDI YA NYANY';

update location SET subvillageId = 'TOA' where KITONGOJI='TOA';

update location SET subvillageId = 'UAR' where KITONGOJI='UARABUNI';

update location SET subvillageId = 'UWN' where KITONGOJI='UWANJA WA NDEGE';

update location SET subvillageId = 'VID' where KITONGOJI='VIDENGE';

update location SET subvillageId = 'VIG' where KITONGOJI='VIGAENI';

update location SET subvillageId = 'ZAH' where KITONGOJI='ZAHANATI';