# ICT Database workflow

A database is used to process the source data needed by the ICT model.

This db goes through a series of steps to select, filter, calculate and combine data from the Health Survey of England (HSE, 2012) and the National Travel Survey (NTS, 2012). In the end a number of files needed for ICT are created and be processed straight away by R scripts.

Of these the main files produced are:

* **[Baseline.csv]**: file with the core variables needed
* **[indivHSE-NTS\_2012.csv]**: matches between HSE individuals and NTS individuals. It's an   
  m ----> n relationship table.
* **[People\_w\_NoTrips2012.csv]**: file with 2 measures of physical activity (leisure mmets and all minus walking & cycling) , plus the 5 variables used for matching (age, sex, car access, Nssec, Ethnicity).

The following document describes the data sources used and the main steps, performed mostly through queries, carried away in the db.

## Data sources

The names of the db main tables match those used in the NTS sources. These are:

* DAY (17 variables): day of the week the trips were made.
* INDIVIDUALS (698 variables): individual details. Most relevant columns are age-sex-ethnicity-car access
* HOUSEHOLD (384 vars): household details. Used mainly to filter the study region (England)
* TRIPS (81 variables): trip variables. Used for all the core variables in the scenarios creation (distances, time, mode of travel, stages, weight equivalents, …)
* STAGES: used to calculate trips involving any sort of Physical Activity (walking / cycling).

## Workflow: main data

The db performs the following operations:

1. Cyclable trips are filtered from NTS datasets:   
    TRIPS – INDIVIDUALS- DAY
2. Walked trips are filtered from NTS datasets:   
    TRIPS – STAGES- INDIVIDUALS- DAY   
   This operation takes into account stages walked & trips with series of calls.
3. 1 and 2 are combined via a union query that calculates the walking/cycling parameters.
4. Walking parameters are added by trip to obtain their distance, times and weighted equivalents.
5. METh and MMETh are added to TRIPS dataset.
6. INDIVIDUAL, TRIPS and HOUSEHOLD dataset are filtered and combined to produce the baseline.csv file. This query:

* Filters for the right age (>=18 y.o) and year (2012).
* Filters for the right region (only England considered).
* Adds derived column Age for broader age bands (16.59, or 60plus).
* Adds derived column 'Cycled' (=1 if [MainMode\_B04ID]=2).
* Adds column 'pcyc' and sets to 0 (pcyc=prob. of cycling).
* Adds column 'nowcycle' and sets to 0 (nowcycle = 1 if trip switched to cycled in a given scenario).
* Merges the columns METh & MMETh

1. INDIVIDUAL& TRIPS are merged and results *grouped* by individuals who have trips. Variables of interest per individual area added (Age, Sex, Ethnicity……).
2. Health Survey of England (HSE) is *matched* to NTS file INDIVIDUALS to produce a file of matching people HSE<>NTS and obtain their MMETS (2 columns). This is done by matching 4 variables: age/sex/ethnicity & socioeconomic status. To allow a complete matching, the initial categories in each dataset are recoded into broader categories (1,2,3) and then matched against the equivalent.
3. People without trips are selected by comparing the INDIVIDUALS file to the TRIPS file and adding the relevant variables.
4. People without trips are linked to their matching individuals (step 8) to get their mmets measures.

## DB queries

The SQL code for the db queries is included for descriptive purposes. The name of the query is may serve as orientation, but always check the query structure first.

CBM-1Trips\_Cyclable:

SELECT trips2012.SurveyYear, trips2012.TripID, trips2012.IndividualID, day2012.TravDay, trips2012.W5, trips2012.W5xHH, trips2012.SeriesCall\_B01ID, trips2012.ShortWalkTrip\_B01ID, trips2012.NumStages, trips2012.MainMode\_B03ID, trips2012.TripPurpFrom\_B01ID, trips2012.TripTotalTime, trips2012.TripTravTime, trips2012.TripOrigGOR\_B02ID, trips2012.TripDestGOR\_B02ID, trips2012.TripDisIncSW, trips2012.TripDisExSW, trips2012.JJXSC, trips2012.JOTXSC, trips2012.JTTXSC, trips2012.JD, 0 AS Cycl\_Impossible, ind2012\_200.Age\_B01ID, ind2012\_200.Sex\_B01ID, ind2012\_200.NSSec\_B03ID, ind2012\_200.CarAccess\_B01ID

FROM (trips2012 INNER JOIN day2012 ON trips2012.DayID = day2012.DayID) INNER JOIN ind2012\_200 ON trips2012.IndividualID = ind2012\_200.IndividualID

WHERE (((ind2012\_200.Age\_B01ID)>=8));

CBM-2WalkingStages:

SELECT trips2012.TripID, Stage2012.StageID, day2012.TravDay, Stage2012.IndividualID, Stage2012.StageSeq, Stage2012.StageDistance, Stage2012.StageTime, Stage2012.StageTime\_B01ID, Stage2012.StageMode\_B03ID, Stage2012.StageMode\_B04ID, Stage2012.StageMode\_B11ID, Stage2012.StageShortWalk\_B01ID, Stage2012.W5, Stage2012.W5xHH, Stage2012.SSXSC, Stage2012.STTXSC, Stage2012.SD, Stage2012.SD\_woW5, Stage2012.SSXSC\_woW5, Stage2012.STTXSC\_woW5, ind2012\_200.Age\_B01ID

FROM ((trips2012 INNER JOIN Stage2012 ON trips2012.TripID = Stage2012.TripID) INNER JOIN ind2012\_200 ON Stage2012.IndividualID = ind2012\_200.IndividualID) INNER JOIN day2012 ON Stage2012.DayID = day2012.DayID

WHERE (((Stage2012.StageMode\_B04ID)=1) AND ((ind2012\_200.Age\_B01ID)>=8))

ORDER BY trips2012.TripID;

CBM-3is1U2:

SELECT [CBM-1Trips\_Cyclable].SurveyYear, [CBM-1Trips\_Cyclable].TripID, [CBM-1Trips\_Cyclable].NumStages, [CBM-1Trips\_Cyclable].IndividualID, [CBM-1Trips\_Cyclable].TravDay, [CBM-1Trips\_Cyclable].SeriesCall\_B01ID, [CBM-1Trips\_Cyclable].ShortWalkTrip\_B01ID, [CBM-1Trips\_Cyclable].MainMode\_B03ID, [CBM-1Trips\_Cyclable].TripTotalTime, [CBM-1Trips\_Cyclable].TripDisIncSW, [CBM-1Trips\_Cyclable].TripDisExSW, [CBM-2WalkingStages].StageID, [CBM-2WalkingStages].StageMode\_B04ID, [CBM-2WalkingStages].StageDistance, [CBM-2WalkingStages].StageTime, [CBM-2WalkingStages].W5, [CBM-2WalkingStages].W5xHH, [CBM-2WalkingStages].SSXSC, [CBM-2WalkingStages].STTXSC, [CBM-2WalkingStages].SD, [CBM-2WalkingStages].SD\_woW5, [CBM-2WalkingStages].SSXSC\_woW5, [CBM-2WalkingStages].STTXSC\_woW5

FROM [CBM-1Trips\_Cyclable] LEFT JOIN [CBM-2WalkingStages] ON [CBM-1Trips\_Cyclable].TripID = [CBM-2WalkingStages].TripID

ORDER BY [CBM-1Trips\_Cyclable].SurveyYear, [CBM-1Trips\_Cyclable].NumStages DESC , [CBM-2WalkingStages].StageID;

CBM4-SumOfWalkingParams:

SELECT [CBM-3is1U2].TripID, Sum([CBM-3is1U2].StageDistance) AS SumofWStageDistance, Sum([CBM-3is1U2].StageTime) AS SumOfWStageTime, Sum([CBM-3is1U2].STTXSC) AS SumOfSTTXSC, Sum([CBM-3is1U2].SD) AS SumOfSD, Sum([CBM-3is1U2].SD\_woW5) AS WalkDistance, Sum([CBM-3is1U2].STTXSC\_woW5) AS WalkTime

FROM [CBM-3is1U2]

GROUP BY [CBM-3is1U2].TripID;

CBM5-TripsBaseline:

SELECT IIf([Age\_B01ID]<16,'16.59','60plus') AS Age, [Age] & [Sex] AS agesex, ind2012\_200.Age\_B01ID, household2012.HHoldGOR\_B02ID, IIf([Sex\_B01ID]=1,'Male','Female') AS Sex, ind2012\_200.Sex\_B01ID, ind2012\_200.CarAccess\_B01ID, ind2012\_200.NSSec\_B03ID, ind2012\_200.IndIncome2002\_B02ID, ind2012\_200.EthGroupTS\_B02ID, trips2012.SurveyYear, trips2012.TripID, trips2012.DayID, trips2012.IndividualID, trips2012.HouseholdID, trips2012.PSUID, trips2012.W5, trips2012.W5xHH, trips2012.TravDay, trips2012.SeriesCall\_B01ID, trips2012.ShortWalkTrip\_B01ID, trips2012.NumStages, trips2012.MainMode\_B03ID, trips2012.MainMode\_B04ID, trips2012.MainMode\_B11ID, trips2012.TripTotalTime, trips2012.TripTravTime, trips2012.TripDisIncSW, trips2012.TripDisExSW, trips2012.JJXSC, trips2012.JOTXSC, trips2012.JTTXSC, trips2012.JD, trips2012.WalkDistance, trips2012.WalkTime, IIf([MainMode\_B04ID]=2,1,0) AS Cycled, 0 AS Pcyc, 0 AS now\_cycle, trips2012.METh, trips2012.MMETh

FROM (trips2012 INNER JOIN ind2012\_200 ON trips2012.IndividualID = ind2012\_200.IndividualID) INNER JOIN household2012 ON ind2012\_200.HouseholdID = household2012.HouseholdID

WHERE (((ind2012\_200.Age\_B01ID)>=8) AND ((household2012.HHoldGOR\_B02ID)<10))

ORDER BY household2012.HHoldGOR\_B02ID;

CBM8\_people\_W\_trips:

SELECT ind2012\_200.IndividualID, ind2012\_200.Age\_B01ID, ind2012\_200.Sex\_B01ID

FROM [CBM5-TripsBaseline] INNER JOIN ind2012\_200 ON [CBM5-TripsBaseline].IndividualID = ind2012\_200.IndividualID

WHERE (((ind2012\_200.SurveyYear)=2012))

GROUP BY ind2012\_200.IndividualID, ind2012\_200.Age\_B01ID, ind2012\_200.Sex\_B01ID

HAVING (((ind2012\_200.Age\_B01ID)>=8));

CBM8\_people\_WO\_trips:

SELECT ind2012\_200.IndividualID, ind2012\_200.Age\_B01ID, ind2012\_200.Sex\_B01ID, ind2012\_200.EthGroupTS\_B02ID, ind2012\_200.CarAccess\_B01ID, ind2012\_200.NSSec\_B03ID

FROM ind2012\_200 LEFT JOIN [CBM5-TripsBaseline] ON ind2012\_200.IndividualID = [CBM5-TripsBaseline].IndividualID

WHERE (((ind2012\_200.SurveyYear)=2012) AND (([CBM5-TripsBaseline].IndividualID) Is Null) AND ((ind2012\_200.W1)=1))

GROUP BY ind2012\_200.IndividualID, ind2012\_200.Age\_B01ID, ind2012\_200.Sex\_B01ID, ind2012\_200.EthGroupTS\_B02ID, ind2012\_200.CarAccess\_B01ID, ind2012\_200.NSSec\_B03ID

HAVING (((ind2012\_200.Age\_B01ID)>=8));

CBM8\_people\_WO\_trips\_ENGLANDonly:

SELECT ind2012\_200.IndividualID, ind2012\_200.Age\_B01ID, household2012.HHoldGOR\_B02ID, ind2012\_200.Sex\_B01ID, ind2012\_200.CarAccess\_B01ID, ind2012\_200.CarAccess\_B02ID, ind2012\_200.NSSec\_B03ID, ind2012\_200.EthGroupTS\_B02ID

FROM (ind2012\_200 INNER JOIN household2012 ON ind2012\_200.HouseholdID = household2012.HouseholdID) LEFT JOIN [CBM5-TripsBaseline] ON ind2012\_200.IndividualID = [CBM5-TripsBaseline].IndividualID

WHERE (((ind2012\_200.Age\_B01ID)>8) AND ((household2012.HHoldGOR\_B02ID)<10) AND ((household2012.SurveyYear)=2012) AND (([CBM5-TripsBaseline].IndividualID) Is Null))

ORDER BY ind2012\_200.IndividualID;

CBM8\_people\_WO\_trips\_Get\_mMets-v6:

SELECT CBM8\_people\_WO\_trips\_ENGLANDonly.IndividualID, CBM8\_people\_WO\_trips\_ENGLANDonly.Age\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.Sex\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.CarAccess\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.NSSec\_B03ID, CBM8\_people\_WO\_trips\_ENGLANDonly.EthGroupTS\_B02ID, Avg([CBM9\_match-Baselinecbm2-HSE7k\_v6].mMETs\_leisure) AS mMETs\_leisure, Avg([CBM9\_match-Baselinecbm2-HSE7k\_v6].[mMETs\_all\_minus\_W&C]) AS [mMETs\_all\_minus\_W&C]

FROM CBM8\_people\_WO\_trips\_ENGLANDonly INNER JOIN [CBM9\_match-Baselinecbm2-HSE7k\_v6] ON CBM8\_people\_WO\_trips\_ENGLANDonly.IndividualID = [CBM9\_match-Baselinecbm2-HSE7k\_v6].IndividualID

GROUP BY CBM8\_people\_WO\_trips\_ENGLANDonly.IndividualID, CBM8\_people\_WO\_trips\_ENGLANDonly.Age\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.Sex\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.CarAccess\_B01ID, CBM8\_people\_WO\_trips\_ENGLANDonly.NSSec\_B03ID, CBM8\_people\_WO\_trips\_ENGLANDonly.EthGroupTS\_B02ID;

CBM9\_match-Baselinecbm2-HSE7k\_v6:

SELECT ind2012\_England.IndividualID, HSE7k\_v6.Age, HSE7k\_v6.Sex, ind2012\_England.NSSec\_B03ID, lkp\_Ethnicity\_CBM.EthGroupTS\_B02ID, ind2012\_England.CarAccess\_B01ID, HSE7k\_v6.mMETs\_leisure, HSE7k\_v6.[mMETs\_all\_minus\_W&C]

FROM HSE7k\_v6 INNER JOIN (lkp\_Age\_CBM1 INNER JOIN ((lkp\_Ethnicity\_HSE INNER JOIN (([lkp\_Nssec8-CBM1] INNER JOIN ([lkp\_Nssec8-CBM2] INNER JOIN ind2012\_England ON [lkp\_Nssec8-CBM2].NSSec\_B03ID = ind2012\_England.NSSec\_B03ID) ON [lkp\_Nssec8-CBM1].numMatch = [lkp\_Nssec8-CBM2].numMatch) INNER JOIN lkp\_Ethnicity\_CBM ON ind2012\_England.EthGroupTS\_B02ID = lkp\_Ethnicity\_CBM.EthGroupTS\_B02ID) ON lkp\_Ethnicity\_HSE.numMatch = lkp\_Ethnicity\_CBM.numMatch) INNER JOIN lkp\_Age\_CBM2 ON ind2012\_England.Age\_B01ID = lkp\_Age\_CBM2.Age\_B01ID) ON lkp\_Age\_CBM1.Age\_level = lkp\_Age\_CBM2.Age\_Level) ON (HSE7k\_v6.Origin = lkp\_Ethnicity\_HSE.Origin) AND (HSE7k\_v6.nssec8 = [lkp\_Nssec8-CBM1].NSSec8) AND (HSE7k\_v6.Age = lkp\_Age\_CBM1.Age) AND (HSE7k\_v6.Sex = ind2012\_England.Sex\_B01ID)

ORDER BY ind2012\_England.IndividualID;