EUROPEAN TOPIC CENTRE ON AIR EMISSIONS

The CORINAIR software tools

UNDER CONTRACT TO THE EUROPEAN ENVIRONMENT AGENCY

CORINAIR DATABASE

Guidelines how to transfer emission data
from the national inventory system
into CORINAIR database

Prepared for the European Environment Agency

by European Topic Centre Air Emissions

Jozef Skákala, Jan Svetlik, Miro Linek

Tinus Pulles

CONTENTS:

PURPOSE **3**

RELATIONAL DATABASE MODEL **3**

CREATING DATABASE FOR THE FIRS TIME **3**

BASIC STRUCTURE OF THE DATABASE **4**

BUILDING INTERFACE BETWEEN CORINAIR AND NATIONAL DATABASE **5**

COMPLETING YOUR ROOT DATA TABLES **5**

PROCEDURE HOW TO LOAD AREA SOURCES DATA INTO CORINAIR DATABASE **6**

PROCEDURE HOW TO LOAD POINT SOURCES DATA INTO CORINAIR DATABASE **8**

ANNEX 1 DATABASE DEFINITION IN SQL **10**

Background

Within the framework of the CORINAIR programme the European Environment Agency (EEA) and its European Topic Centre on Air Emissions (ETC-AE) have developed a set of software tools to support national experts in compiling annual emission inventories. These tools allow for a transparent and standardised, hence comparable, data collecting and emissions reporting procedure within the framework of different international conventions and protocols. The basis of the CORINAIR tools is a relational database model covering the data and their relations required. The simplified background of the model is defined by three-dimension cube of entities: ***NUTS*** *(****N****omenclature of* ***S****tatistical* ***T****erritorial* ***U****nits) per* ***SNAP*** *(****S****elected* ***N****omenclature for sources of* ***A****ir* ***P****ollution) per* ***Pollutants*** *(set of pollutants to be reported). However, the overall database model appears to be much more complex involving 35 tables. To understand this model seems to be very important for the purpose of compiling a national air emission inventory.*

Purpose

The CORINAIR database was designed for purpose of storing and managing emission inventory data by CollectER/ ReportER software tools. Database model represents a consistent structure of data (entity tables and their relations), which reflects the real world of emission data to be collected. The standard software tools (CollectER, ReportER, ImportER...) prevent the user to destroy the consistency of this database, e.g. to add some values of emission or activities without any correct relation to the other important data in the database like nomenclatures of SNAPs, pollutants, fuels and NUTS codes. However, considering the ambition of some experts to access the database directly (e.g. by MS ACCESS) for the purpose of loading national data from different sources like MS Excel sheets, the understanding of the relational database model is required.

Relational database model

Relational model is a method of organizing data into two-dimensional tables made up of rows and columns. The model is based on the mathematical theory of relations, a part of set theory. The Relational Database is based on the relational modeling theory, it present a collection of information organized in tables. Each table models a class of objects of interest to the organization (for example, Large Point Source, LPS Parts, Emission). Each column in a table models an attribute of the object modeled by the table (for example, LPSName, Address,). Each row in a table represents one entity in the class of objects modeled by the table (for example, the LPS Name “Oildata Industries” or the Address “Rotterdam Botlek”). Queries can use data from one table to find related data in other tables. Specific type of columns in the table tagged as “key” provides the relations to other tables. Key columns in the table distinguish the primary keys having unique constraints or foreign keys i.e. defined as primary in other table.

Creating database for the first time

The procedure of compiling the emission inventory by the CollectER tool is based on the previous year data. Consequently the significant attention should be applied to the procedure of creating the first inventory in the CORINAIR db format. The new national database can be created using three different ways:

a. **Opening “Blank97.mdb”** database for the first time, selecting actual country name from combo box and actual year of the inventory and creating national database. In this case the user uses CollectER software tool to define area activities and point sources and fills this table manually from beginning using “Root Data” tables.

b. **Using ImportER tool** to import data contained in the previous CORINAIR 94 type database (or better ask SPIRIT to do it because of some experiences). This way enables to load majority of national data from previous CORINAIR database into CollectER database format (with SNAP97). However it requires in some cases some additional manual work, which should be done by national expert.

c. Direct transfer of **data from specific National Emission Inventory System** (NEIS) – which is probably most complicated and depends highly on the structure of national databases. It is not possible to create universal software tool, which will perform this transfer from every national database. However some guidelines could be introduced, which will help the national expert to transfer particular data or write own transfer procedure.

More details about methods a) and b) could be find in the “CollectER user guide”. The aim of this paper is to define guidelines, which will help the national expert to transfer data directly from his national emission database into CollectER database. The benefit of this will be, the data stored in the unified CollectER database format are enough transparent and comparable on the international level. Furthermore the unique ReportER software tool can be used to prepare international Reports.

Basic structure of the database

Three basic groups of tables could be identified in the new CORINAIR database format. Furthermore exists one specific table containing system information (database version, working-actual year and base or previous year, if any).

Root data group of tables contains nomenclatures, generally used and predefined by ETC/AE

The root data tables are divided in three subgroups:

1. Base data subgroup – common used data
   - ✓ Pollutants (83 basic pollutants applied in the most important reports)
   - ✓ Territorial units (national territorial split of 4 levels following Eurostat NUTS definition)
   - ✓ SNAP (Selected Nomenclature for Air Pollution activities of 3 levels – sector, subsector and activity, managed by ETC/AE, actual version from 1997)
   - ✓ Splits (table enabling more detailed splitting of activity, if required)
   - ✓ Fuels (table of predefined fuels and their parameters)
2. Auxiliary data – suppose to be edited by user
   - ✓ Conversions between units
   - ✓ Units of different variables used in system
   - ✓ Comments – free text
   - ✓ Default activities and Emission factors – suppose to be edited according to the national condition
3. Surrogate data – tables enabling specific (surrogate) calculation of emission
   - ✓ Surrogate data types
   - ✓ Surrogate data
   - ✓ Allocation formulae

Area sources group of tables are used for calculation emissions produced by area sources. Could be divided into two subgroups

a. Area sources definition tables
   - ✓ act_area – contains definitions of area activities (which are, combined with NUTS codes area sources)
   - ✓ emfa_asz – contains general emission factors

b. Area activity data tables – contain activity rates and local emission factors
   - ✓ as_rates – activity rates
   - ✓ emfa_as – local emission factors (defined for particular NUTS)
   - ✓ as_ratold – activity rates from previous year

Point sources group of tables are used for calculation of emissions produced by point sources. In this case the division into definition and data tables is not so clear as in the case of area sources. This is because of storing PS activity definition data together with activity rate values in the same table lps_rates.

PS tables:

- ✓ lps_def – PS definition (at NUTS3, i.e. lowest NUTS level)
- ✓ lps_parts – definition of parts. Every part posses one emission producing activity defined by SNAP code
- ✓ lps_stack – definition of PS stacks. In current CollectER version for information purposes only.
- ✓ act_stack – emission distribution through stacks of actual PS. In current CollectER version for information purposes only.
- ✓ lps_act_emis – table containing direct(measured) emission produced by parts of the PS
- ✓ lps_rates - contains PS activity definitions (SNAP(fuel ,split)) and appropriate activity rates
- ✓ emf_lps – contains emission factors for PS activities.

Building interface between CORINAIR and national database

Starting for the first time, please use CollectER software and Blank97 database to produce your dummy working database structure. However, if any previous year CollectER database, use this to produce the initial database format.

We recommend to consider carefully, which data are of amount, to be transferred from national format to the CORINAIR database in profitable way. It is always more convenient to edit manually particular data by the CollectER tool, than to study the DB structure and prepare some ad hoc procedure.

However, there are some possibilities to use the COPY/PASTE windows standards to transfer data from national database very effectively. Some editable fields in the CollectER software, as activity rates, could be filled in by usage the paste data function.

Handling the CORINAIR database by the MS ACCESS tool, requires to meet some principle:

- ✓ Prefer editing of the particular data with the CollectER tool before you start to use MS Access.
- ✓ All root tables must be completed, before you start to handle some derivative table. Please follow the dependency of derivative tables on the scheme below (figure 1 and 2), e.g. the “act_area” table on the figure 1 could be filled with data only if the root tables “snaps”, “splits” and “fuels” are completed.
- ✓ In the like manner the master tables must be completed, before the slave table will be filled in, e.g. the “act_area” and “nuts” tables must be in accordance with your inventory before you will start to fill the table “as_rates” (figure 1).
- ✓ They are no air emissions saved into the CORINAR database (excepting the LPS parts direct emissions table). All of air emissions reported by the CollectER/ ReportER tool are calculated dynamically. On the score of the actual values of the emission factor and the activity rate will be involved and the usage of the disk memory will be reduced considerably.
- ✓ If you are not familiar with the detailed database description (Annex 1) or with the recommendations mentioned above, please don’t try to handle CORINAIR database with the MS ACCESS tool.

Completing your root data tables

Step 1. Define your national territorial split definition into tables NUTS0, 1, 2, 3 according to following example:

level	length of id	example id	name of unit
NUTS0	**2 char**	**me**	**Middle Earth**
NUTS1	**3 char**	**me1**	**Coastal**
NUTS2	**4 char**	**me11**	**Eriador**
NUTS3	**5 char**	**me111**	**Arnor**

Higher-level code must be included in lower level code. Replace your local codes with NUTS codes or create temporary transformation table, which will be used during data transfer.

Step 2. Create temporary transformation table between SNAP codes at level 2 and your local codes, which specify emission producing activities. If possible replace your local codes with SNAP codes.

Step 3. Create temporary transformation table between your local pollutant codes and CORINAIR pollutant codes.

Step 4. Create temporary transformation table between your local fuel codes and CORINAIR fuel codes.

Procedure how to load area sources data into CORINAIR database

The basic procedure consists of defining the local activities in the “act_area” table. All existing combinations of “SNAP, Fuel, split and pollutant” should be defined. Consequently on the lowest available level of the territorial unit (NUTS) the rate of a relevant activity should be defined (table “as_rates”). Likewise, if any NUTS specific local emission factor exists, should be defined in the table “emfa_as”.

Finally the activity rates should be defined on all territorial levels (NUTS0, 1, 2, 3). This could be reached by filling the data implicitly for all NUTS levels in the table “as_rates” or by using the CollectER function “calculate up” or “calculate down” to recalculate the values from a particular NUTS level.

Step 5. Create transformation of general emission factors. Emission factors are defined for *area activities(SNAP*(fuel, split)) and *pollutants* .
   a. Replace your local emission producing activities codes with SNAP codes in your emission factor table (exploit the temporary table prepared above).
   b. Replace your local pollutant codes with CORINAIR pollutant codes in your emission factor table (exploit the temporary table prepared before).
   c. Replace your local fuel codes with CORINAIR fuel codes in your emission factor table (exploit the temporary table prepared before).
   d. Replace your local splits with CORINAIR splits.

Step 6. Create transformation of local emission factors (if any). Emission factors are defined for *area activities(SNAP*(fuel, split)) and *pollutants* .
   a. Replace your local emission producing activities codes with SNAP codes in your local emission factor table.
   b. Replace your local codes of territorial units with NUTS codes in your local emission factor table.
   c. Replace your local pollutant codes with CORINAIR pollutant codes in your local emission factor table.
   d. Replace your local fuel codes with CORINAIR fuel codes in your local emission factor table.
   e. Replace your local splits with CORINAIR splits.

Step 7. Filling activity rates in the as_rates table:
Area source is defined at the particular level of territorial split by area activity SNAP code, fuel code (if used), split code (if used), and territorial unit NUTS code.
There are also levels of SNAP a NUTS and fuel group code *fuel_gr_id* stored for calculation purposes. This can be filled by update of the “as_rates” table:

SNAP (only levels 2 and 1 are allowed in the activity definition)

```
UPDATE as_rates SET lev_snap = 2 WHERE LEN(snap_id) = 6
UPDATE as_rates SET lev_snap = 1 WHERE LEN(snap_id) = 4
```

similar for NUTS

```
UPDATE as_rates SET lev_nuts = (LEN(nuts_id) – 2)
```

and for fuels

```
UPDATE as_rates SET fuel_gr_id = LEFT(fuel_id,3)
```

Important note: Fields, which are included in the primary key (Annex 1) must not contain null values. This could happen in the spl_id and fuel_id fields. Update such values into empty string:

```
UPDATE as_rates SET spl_id = '' WHERE spl_id IS NULL
UPDATE as_rates SET fuel_id = '' WHERE fuel_id IS NULL
```

Finally enter values and units of activity rates in as_rates table. Values of activity rates should be inserted into fiels ***asrat_val***. After this step field asrat_typd should be updated

```
UPDATE as_rates SET asrat_typd = 'I'
```

Remaining fields (asrat_conf, asrat_qual and comment) need not be filled in.

Area Sources scheme

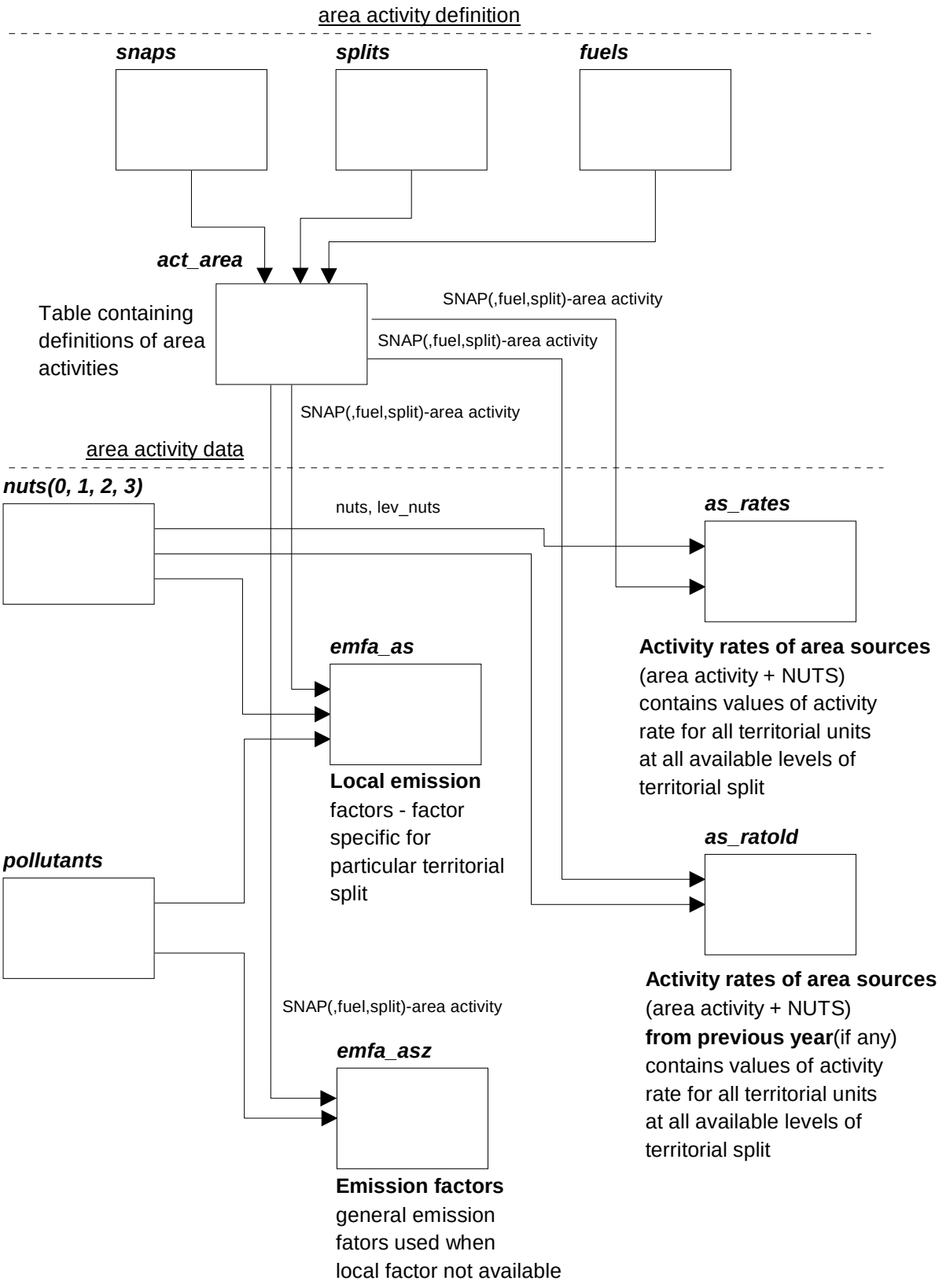

Figure 1.

Procedure how to load point sources data into CORINAIR database

There are first four steps (Step 1, till 4) common with that for area sources described in the previous section. If the transformation of NUTS, SNAP, Pollutants and Fuels is already done, you can continue to the step 5.

Step 5.Point sources definition: Every point source has unique identification code and geographical position. It is defined at lowest territorial split level 3, i.e. nuts_id field contains code from NUTS3 table. Point source has one or more parts and no or more stacks.

a. part of point sources has an identifier, which is unique within particular point source. Further, they have SNAP codes to define its emission producing activity. Several parts of the same point source may have the same SNAP code.
b. stack of the point source has an identifier, which is unique within particular point source.

Step 6. There are two types of emission, which can be entered for particular point source:

a. direct(or measured) emissions, which are stored in the table lps_act_emis
b. emissions calculated from activity rates and emission factors. Activity rates are stored in the lps_rates table, emission factors in the emfa_lps table.

Direct emissions are stored in the lps_act_emis table and connected directly to parts of PS. Direct emissions are independent from fuels consumption or split and have higher priority in the calculation process, i.e. if there is entry for particular part and pollutant, this will be included. If there are activity rate and emission factor entries in appropriate tables, these will be ignored. Only part definition, emission value and unit fields need to be filled in this table. Other fields can be used optionally.

If there is no entry in the lps_act_emis table for particular part of PS and pollutant, emission will be calculated using value of activity rate and value of emission factor.

activity rate – value of activity rate is stored in the lps_rates table. This value is connected to the activity of point source, which is similar to area source defined by

- lps_id – point source
- part_id – part of point source

and

- SNAP
- fuel (if any)
- split (if any)

Important Note: Similar to „as_rates“ table fields which are included in the primary key (Annex 1) must not contain null values. This could happen in the spl_id and fuel_id fields. Update such values into empty string:

```
UPDATE as_rates SET spl_id = '' WHERE spl_id IS NULL
UPDATE as_rates SET fuel_id = '' WHERE fuel_id IS NULL
```

To fill snap_lev and fuel_gr_id fields, following steps should be preformed
SNAP (only levels 2 and 1 are allowed in the activity definition)

```
UPDATE as_rates SET lev_snap = 2 WHERE LEN(snap_id) = 6
UPDATE as_rates SET lev_snap = 1 WHERE LEN(snap_id) = 4
UPDATE as_rates SET fuel_gr_id = LEFT(fuel_id,3)
```

Finally it is necessary to fill uni_id field. However it is not necessary to fill other fields (lpsrat_old, lps_conf, lpsrat_conf_old, lpsrat_qual, lpsrat_qual_old, uni_id_old, comment).

Emission factors are stored in the emfa_lps table. Emission factors are defined for activities connected to parts of point sources, i.e. substantial part of emission factor identification is activity identification (*lps_id, part_id, snap_id, fuel_id, spl_*id) with added identifier of pollutant. Similar to lps_rates table *lev_snap* and *fuel_gr_id* should be set.

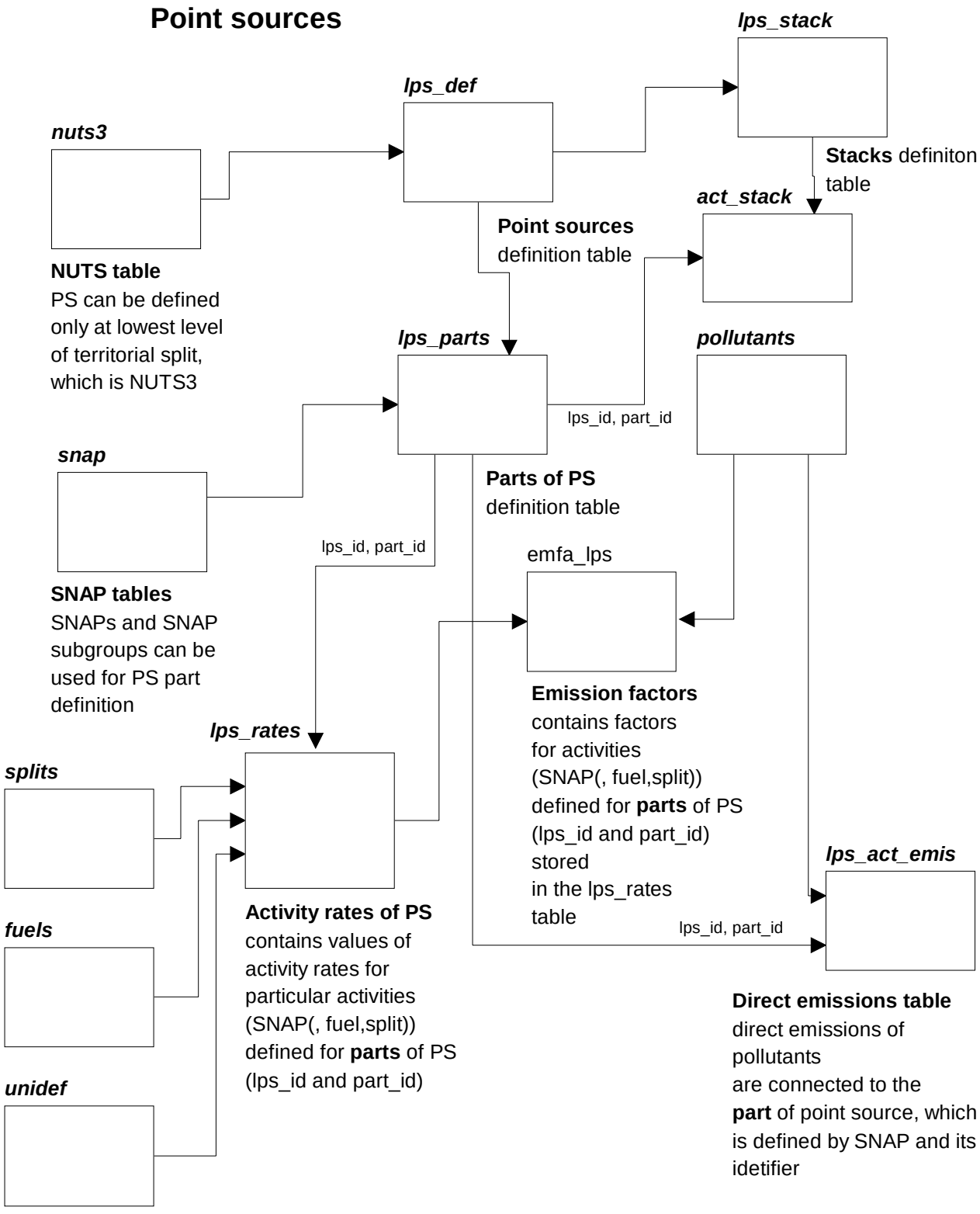
Point sources
lps_stack
lps_def
nuts3
Stacks definiton table
act_stack
Point sources definition table
NUTS table
PS can be defined only at lowest level of territorial split, which is NUTS3
lps_parts
pollutants
lps_id, part_id
snap
Parts of PS definition table
lps_id, part_id
emfa_lps
SNAP tables
SNAPs and SNAP subgroups can be used for PS part definition
Emission factors
contains factors for activities (SNAP(, fuel,split)) defined for parts of PS (lps_id and part_id) stored in the lps_rates table
lps_rates
splits
lps_act_emis
lps_id, part_id
fuels
Activity rates of PS
contains values of activity rates for particular activities (SNAP(, fuel,split)) defined for parts of PS (lps_id and part_id)
unidef
Direct emissions table
direct emissions of pollutants are connected to the part of point source, which is defined by SNAP and its idetifier

Figure 2.

Annex 1 Database definition in SQL

```
CREATE TABLE Act_stack                          //stack to point source part definition table//
    (lps_id          TEXT(4),      FK           //large point source (LPS) identifier FK//
     lpsst_id        TEXT(2),      FK           //stack identifier//
     part_id         TEXT(3),      FK           //identifier of the part of LPS//
     perc            SHORT,                     //activity rate per cent connected to stack//
     comment         TEXT(5)                    //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON Act_stack
(
     lps_id          ASC,
     lpsst_id        ASC,
     part_id         ASC
);


CREATE TABLE Comment                            //comment definiftion table//
    (Codcom          TEXT(5),                   //comment code(identifier)//
     Textcom         TEXT(250),                 //description of the stack//
     fl_stand        TEXT(1)                    //comment property flag - standard/user defined//
);

CREATE UNIQUE INDEX PrimaryKey
  ON Comment
(
     Codcom          ASC
);


CREATE TABLE Split                              //split definition table//
    (spl_id          TEXT(3),                   //split identifier//
     spl_name        TEXT(45),                  //split name//
     spl_abbr        TEXT(15),                  //abbreviation of the split name//
     usr_def         TEXT(1)                    //split property flag - user defined (Y/N)
);

CREATE UNIQUE INDEX PrimaryKey
  ON Split
(
     spl_id          ASC
);


CREATE TABLE Strings                            //system table//
    (Name            TEXT(2),                   //property code//
     Texts           TEXT(50)                   //property value//
);

CREATE UNIQUE INDEX PrimaryKey
  ON Strings
(
     Name            ASC
);


CREATE TABLE act_area                           //definition table of area activities, which can be used in
                                                the inventory//
    (fuel_id         TEXT(4),      FK           //fuel identifier//
     spl_id          TEXT(3),      FK           //split identifier//
     snap_id         TEXT(6),      FK           //SNAP identifier//
     lev_snap        SHORT,                     //level of the SNAP idetifier (1,2,3)//
```

```
    uni_id          SHORT,          //unit identifier//
    uni_id_old      SHORT,          //identifier of unit, which was used in previous year//
    comment         TEXT(5)         //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON act_area
(
    fuel_id             ASC,
    spl_id              ASC,
    snap_id             ASC
);


CREATE TABLE actdef                 //list of activities used in the inventory//
    (spl_id         TEXT(3),    FK  //split identifier//
    snap_id         TEXT(6),    FK  //SNAP identifier//
    fuel_gr_id      TEXT(3),    FK  //fuel group identifier//
    lev_snap        SHORT,          //level of the SNAP identifier (1,2,3)//
    uni_id          SHORT,          //unit identifier//
    comment         TEXT(5)         //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON actdef
(
    spl_id              ASC,
    snap_id             ASC,
    fuel_gr_id          ASC
);


CREATE TABLE as_rates               //table of activity rates for area activities//
    (fuel_id        TEXT(4),    FK  //fuel identifier//
    nuts_id         TEXT(5),    FK  //NUTS identifier//
    spl_id          TEXT(3),    FK  //split identifier//
    snap_id         TEXT(6),    FK  //SNAP identifier//
    fuel_gr_id      TEXT(3),    FK  //fuel group identifier//
    lev_snap        SHORT,          //level of the SNAP identifier (1,2,3)//
    asrat_val       DOUBLEFLOAT,    //value of activity rate//
    asrat_typd      TEXT(1),        //activity rate value type Ins-inserted,Calc-calculated//
    asrat_conf      TEXT(1),        //confidentiality of data//
    asrat_qual      TEXT(1),        //quality of data//
    lev_nuts        SHORT,          //level of the NUTS identifier (0,1,2,3)//
    comment         TEXT(5)         //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON as_rates
(
    fuel_id             ASC,
    nuts_id             ASC,
    spl_id              ASC,
    snap_id             ASC
);


CREATE TABLE as_ratold              //table of old activity rates for area activities(e.g. from
                                    prev.year//
    (spl_id         TEXT(3),    FK  //split identifier//
    snap_id         TEXT(6),    FK  //SNAP identifier//
    fuel_id         TEXT(4),    FK  //fuel identifier//
    nuts_id         TEXT(5),    FK  //NUTS identifier//
    lev_snap        SHORT,          //level of the SNAP identifier (1,2,3)//
    lev_nuts        SHORT,          //level of the NUTS identifier (0,1,2,3)//
    asrat_val       DOUBLEFLOAT,    //value of activity rate//
```

```
    asrat_typd          TEXT(1),          //activity rate value type Ins-inserted,Calc-calculated//
    asrat_conf          TEXT(1),          //confidentiality of data//
    asrat_qual          TEXT(1)           //quality of data//
);

CREATE UNIQUE INDEX PrimaryKey
  ON as_ratold
(
    spl_id                ASC,
    snap_id               ASC,
    fuel_id               ASC,
    nuts_id               ASC
);


CREATE TABLE UConvert                     //units conversion table//
    (uni_id1            SHORT,            //first unit identifier//
    uni_id2             SHORT,            //second unit identifier//
    ratio              TEXT(25)           //ratio between units//
);

CREATE UNIQUE INDEX PrimaryKey
  ON convert
(
    uni_id1               ASC,
    uni_id2               ASC
);


CREATE TABLE emfa_as                      //local emission factors for area sources//
    (spl_id             TEXT(3),    FK    //split identifier//
    fuel_id             TEXT(4),    FK    //fuel identifier//
    nuts_id             TEXT(5),    FK    //NUTS identifier//
    pol_id              TEXT(3),    FK    //pollutant identifier//
    snap_id             TEXT(6),    FK    //SNAP identifier//
    fuel_gr_id          TEXT(3),    FK    //fuel group identifier//
    lev_snap            SHORT,            //level of the SNAP identifier (1,2,3)//
    lev_nuts            SHORT,            //level of the NUTS identifier (0,1,2,3)//
    emfa_val            DOUBLEFLOAT,      //value of the emission factor//
    emfa_qual           TEXT(1)           //quality of emission factor//
);

CREATE UNIQUE INDEX PrimaryKey
  ON emfa_as
(
    spl_id                ASC,
    fuel_id               ASC,
    nuts_id               ASC,
    pol_id                ASC,
    snap_id               ASC
);


CREATE TABLE emfa_asz                     //global emission factors for area sources//
    (fuel_id            TEXT(4),    FK    //fuel identifier//
    pol_id              TEXT(3),    FK    //pollutant identifier//
    spl_id              TEXT(3),    FK    //split identifier//
    snap_id             TEXT(6),    FK    //SNAP identifier//
    fuel_gr_id          TEXT(3),    FK    //fuel group identifier//
    lev_snap            SHORT,            //level of the SNAP identifier (1,2,3)//
    emfa_val            DOUBLEFLOAT,      //value of the emission factor//
    emfa_qual           TEXT(1),          //quality of emission factor//
    comment             TEXT(5),          //comment identifier//
    uni_id              SHORT             //unit identifier//
);
```

```
CREATE UNIQUE INDEX PrimaryKey
  ON emfa_asz
(
      fuel_id                 ASC,
      pol_id                  ASC,
      spl_id                  ASC,
      snap_id                  ASC
);


CREATE TABLE emfa_defa                          //default emission factors//
     (pol_id              TEXT(3),        FK    //pollutant identifier//
      spl_id              TEXT(3),        FK    //split identifier//
      snap_id              TEXT(6),       FK    //SNAP identifier//
      fuel_gr_id           TEXT(3),       FK    //fuel group identifier//
      lev_snap             SHORT,               //level of the SNAP identifier (1,2,3)//
      emfa_val             DOUBLEFLOAT,         //value of the emission factor//
      emfa_qual            TEXT(1),             //quality of emission factor//
      uni_id              SHORT                 //unit identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON emfa_defa
(
      pol_id                  ASC,
      spl_id                  ASC,
      snap_id                  ASC,
      fuel_gr_id               ASC
);


CREATE TABLE emfa_lps                           //emission factors for point sources//
     (fuel_id             TEXT(4),        FK    //fuel identifier//
      spl_id              TEXT(3),        FK    //split identifier//
      snap_id              TEXT(6),       FK    //SNAP identifier//
      part_id             TEXT(3),        FK    //LPS part identifier//
      lps_id              TEXT(4),        FK    //LPS identifier//
      pol_id              TEXT(3),        FK    //pollutant identifier//
      fuel_gr_id           TEXT(3),       FK    //fuel group identifier//
      lev_snap             SHORT,               //level of the SNAP identifier (1,2,3)//
      emfa_val             DOUBLEFLOAT,         //value of the emission factor//
      emfa_qual            TEXT(1),             //quality of emission factor//
      uni_id              SHORT,                //unit identifier//
      comment              TEXT(5)              //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON emfa_lps
(
      fuel_id                 ASC,
      spl_id                  ASC,
      snap_id                  ASC,
      part_id                 ASC,
      lps_id                  ASC,
      pol_id                  ASC,
      fuel_gr_id               ASC
);


CREATE TABLE fuel_grp                           //fuel groups definition table//
     (fuel_gr_id          TEXT(3),             //fuel group identifier//
      fuel_gr_name         TEXT(50),           //fuel group name//
      fuel_gr_abbr         TEXT(15),           //abbreviation of the fuel group name//
      fuel_form            TEXT(2)             //reserved for form of the fuel - solid, liquid,...//
);
```

```
CREATE UNIQUE INDEX PrimaryKey
  ON fuel_grp
(
     fuel_gr_id                ASC
);


CREATE TABLE fuels                                  //fuels definition table//
     (fuel_id              TEXT(4),                 //fuel identifier//
      fuel_gr_id            TEXT(3),       FK      //fuel group identifier//
      fuel_ref             TEXT(1),                 //refernce code Dry/Wet//
      fuel_saf             TEXT(1),                 //S% ash free//
      fuel_scont            SHORT,                  //sulphur content//
      fuel_ccont            SHORT,                  //carbon content//
      fuel_ashcont          SHORT,                  //ash content//
      fuel_lhv             SHORT,                   //fuel LHV//
      fuel_water            SHORT,                  //water content//
      comment               TEXT(5)                 //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON fuels
(
     fuel_id                   ASC
);

CREATE UNIQUE INDEX PrimaryKey
  ON fuels
(
     fuel_id                   ASC
);


CREATE TABLE lps_act_emis                           //direct emissions from point sources//
     (pol_id               TEXT(3),       FK      //pollutant identifier//
      part_id              TEXT(3),       FK      //identifier of LPS part//
      lps_id               TEXT(4),       FK      //point source identifier//
      uni_id               SHORT,                 //unit identifier//
      em_val                DOUBLEFLOAT,          //value of emission//
      em_qual               TEXT(1),              //quality of emission value//
      em_conf               TEXT(1),              //emission confidentiality//
      em_old                DOUBLEFLOAT,          //emission from previous year//
      uni_id_old            SHORT,                //unit identifier of the emission from previous year//
      em_qual_old             TEXT(1),            //quality of emission value from previous year//
      em_conf_old             TEXT(1),            //quality of emission value from previous year//
      comment               TEXT(5)               //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON lps_act_emis
(
     pol_id                    ASC,
     part_id                   ASC,
     lps_id                   ASC
);


CREATE TABLE lps_def                                //large point sources definition tables//
     (lps_id               TEXT(4),               //point source identifier//
      lps_ctc              LONG,                  //LPS corrected thermal capacity//
      nuts3_id              TEXT(5),       FK      //NUTS 3 identifier//
      lps_name               TEXT(50),            //point source name//
      lps_lg               DOUBLEFLOAT,           //longitude//
      lps_lg_c              TEXT(1),              //longitude cardinal point W/E//
      lps_lt               DOUBLEFLOAT,           //latitude//
      lps_lt_c              TEXT(1),              //latitude cardinal point N/S//
```

```
    ipcc_code          TEXT(5)                  //IPCC code//
    nose_p             TEXT(6)                  //reserved for NOSE-P code//
    nace_code          TEXT(5)                  // reserved for NACE code//
    comment            TEXT(5)                  //comment//
);

CREATE UNIQUE INDEX PrimaryKey
  ON lps_def
(
    lps_id                ASC
);


CREATE TABLE lps_parts                          //definition table of the LPS parts//
    (part_id           TEXT(3),                 //part identifier//
    lps_id             TEXT(4),        FK       //point source identifier//
    snap_id            TEXT(6),        FK       //SNAP identifier//
    lev_snap           SHORT,                   //level of the SNAP identifier (1,2,3)//
    lcp_flag           INTEGER                  //LCP flag 0-no LCP source,1- new source, 2-old source)
    comment            TEXT(5)                  //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON lps_parts
(
    part_id               ASC,
    lps_id                ASC
);


CREATE TABLE lps_rates                          //point sources activity rates table//
    (snap_id           TEXT(6),        FK       //SNAP identifier//
    spl_id             TEXT(3),        FK       //split identifier//
    part_id            TEXT(3),        FK       //part identifier//
    lps_id             TEXT(4),        FK       //point source identifier//
    fuel_id            TEXT(4),        FK       //fuel identifier//
    fuel_gr_id         TEXT(3),        FK       //fuel group identifier//
    lev_snap           SHORT,                   //level of the SNAP identifier (1,2,3)//
    lpsrat_old         DOUBLEFLOAT,             //value of activity rate from previous year//
    lpsrat_conf_old    TEXT(1),                 //confidentiality of activity rate from previous year//
    lpsrat_qual_old    TEXT(1),                 //quality of activity rate from previous year//
    lpsrat_val         DOUBLEFLOAT,             //value of activity rate//
    lpsrat_conf        TEXT(1),                 //confidentiality of activity rate//
    lpsrat_qual        TEXT(1),                 //quality of activity rate//
    uni_id             SHORT,                   //unit identifier//
    uni_id_old         SHORT,                   //unit identifier from previous year//
    comment            TEXT(5)                  //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON lps_rates
(
    snap_id               ASC,
    spl_id                ASC,
    part_id               ASC,
    lps_id                ASC,
    fuel_id               ASC,
    fuel_gr_id            ASC
);


CREATE TABLE lps_stack                          //stack definition table//
    (lps_id            TEXT(4),        FK       //point source identifier//
    lpsst_id           TEXT(2),                 //stack identifier//
    lpsst_he           DOUBLEFLOAT,             //stack height//
    lpsst_out          DOUBLEFLOAT,             //stack outlet area (km2)//
```

```
    lpsst_tmp          DOUBLEFLOAT,          //temperature of gases//
    lpsst_flr          DOUBLEFLOAT,          //flow rate Nm/h dry//
    comment            TEXT(5)               //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
 ON lps_stack
(
    lps_id                ASC,
    lpsst_id              ASC
);


CREATE TABLE media                            //media definition table//
    (med_id            TEXT(2),               //media identifier//
    med_name           TEXT(50)               //media name//
);



CREATE UNIQUE INDEX PrimaryKey
 ON media
(
    med_id                ASC
);


CREATE TABLE nuts0                            //NUTS 0 definition tables//
    (nuts0_id          TEXT(2),               //NUTS 0 identifier//
    nuts0_name         TEXT(50),              //NUTS 0 name//
    local_code         TEXT(15)               //NUTS 0 local code//
);


CREATE UNIQUE INDEX PrimaryKey
 ON nuts0
(
    nuts0_id              ASC
);


CREATE TABLE nuts1                            //NUTS 1 definition tables//
    (nuts1_id          TEXT(3),               //NUTS 1 identifier//
    nuts0_id           TEXT(2),       FK      //NUTS 0 identifier(parent)//
    nuts1_name         TEXT(50),              //NUTS 1 name//
    local_code         TEXT(15)               //NUTS 1 local code//
);

CREATE UNIQUE INDEX PrimaryKey
 ON nuts1
(
    nuts1_id              ASC
);


CREATE TABLE nuts2                            //NUTS 2 definition tables//
    (nuts2_id          TEXT(4),               //NUTS 2 identifier//
    nuts1_id           TEXT(3),       FK      //NUTS 1 identifier(parent)//
    nuts2_name         TEXT(50),              //NUTS 2 name//
    local_code         TEXT(15)               //NUTS 2 local code//
);

CREATE UNIQUE INDEX PrimaryKey
 ON nuts2
(
    nuts2_id              ASC
```

```
);


CREATE TABLE nuts3                              //NUTS 3 definition tables//
     (nuts3_id              TEXT(5),            //NUTS 3 identifier//
      nuts2_id              TEXT(4),      FK    //NUTS 2 identifier(parent)//
      nuts3_name             TEXT(50),          //NUTS 3 name//
      local_code            TEXT(15)            //NUTS 3 local code//
);

CREATE UNIQUE INDEX PrimaryKey
  ON nuts3
(
      nuts3_id                  ASC
);


CREATE TABLE pol_grp                            //definition of pollutant groups//
     (pol_grp_id            TEXT(2),            //pollutant group identifier//
      pol_grp_name           TEXT(45)           //pollutant group name//
);


CREATE UNIQUE INDEX PrimaryKey
  ON pol_grp
(
      pol_grp_id                ASC
);


CREATE TABLE pollutants                         //definition of pollutants//
     (pol_id                TEXT(3),            //pollutant identifier//
      pol_grp_id             TEXT(2),      FK    //pollutant group identifier//
      med_id                TEXT(2),            //media identifier//
      pol_name               TEXT(50),          //pollutant name//
      pol_abbr              TEXT(15),           //abbreviation of the pollutant name//
      pol_uni_ca             TEXT(3),           //unit for calculation(identifier)//
      pol_uni_re             TEXT(3),           //unit for report(identifier)//
      pol_Aeq               DOUBLEFLOAT,        //Acid equivalent//
      pol_Ceq               DOUBLEFLOAT         //Carbon equivalent//
);


CREATE UNIQUE INDEX PrimaryKey
  ON pollutants
(
      pol_id                    ASC
);


CREATE TABLE snap                               //SNAP definition table//
     (snap_id               TEXT(6),            //SNAP identifier//
      snapsg_id              TEXT(4),      FK    //SNAP subgroup(subsector) identifier//
      snap_name              TEXT(60),          //SNAP name//
      uni_id                SHORT               //unit identifier//
);


CREATE UNIQUE INDEX snap_id
  ON snap
(
      snap_id                   ASC
);


CREATE TABLE snap_grp                           //SNAP groups(sectors) identifier table//
```

```
    (snap_gr_id          TEXT(2),                  //SNAP group identifier//
     snap_gr_name        TEXT(45)                  //SNAP group name//
);


CREATE UNIQUE INDEX snap_gr_id
  ON snap_grp
(
     snap_gr_id              ASC
);


CREATE TABLE snap_spl                              //SNAP - split definition table//
     (spl_id             TEXT(3),        FK        //split identifier//
      snap_id            TEXT(6)         FK        //SNAP identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON snap_spl
(
     spl_id                  ASC,
     snap_id                 ASC
);


CREATE TABLE snapsbgr                              //SNAP subgroups(subsectors) definition table//
     (snapsg_id          TEXT(4),                  //SNAP subgroup identifier//
      snap_gr_id         TEXT(2),        FK        //SNAP group identifier//
      snapsg_name        TEXT(45)                  //SNAP subgroup name//
);


CREATE UNIQUE INDEX snapsg_id
  ON snapsbgr
(
     snapsg_id               ASC
);


CREATE TABLE surrdata                              //surrogate data values//
     (nuts_id            TEXT(5),        FK        //NUTS identifier//
      surr_id            SHORT,          FK        //identifier of the surrogate data type//
      surr_val           DOUBLEFLOAT,              //value of surrogate data//
      comment            TEXT(5)                   //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON surrdata
(
     nuts_id                 ASC,
     surr_id                 ASC
);


CREATE TABLE surrdef                               //definition of surrogate data types//
     (surr_id            SHORT,                    //identifier of the surrogate data type//
      surr_name          TEXT(45),                 //surrogate data name//
      surr_abbr          TEXT(15),                 //abbreviation of the surrogate data name//
      uni_id             SHORT,                    //unit identifier//
      comment            TEXT(5),                  //comment identifier//
      lev_nuts           SHORT                     //level of the NUTS identifier (0,1,2,3)//
);

CREATE UNIQUE INDEX PrimaryKey
  ON surrdef
(
```

```
    surr_id                 ASC
);

CREATE TABLE surrform                                   //table of surrogate formulae//
    (fuel_id            TEXT(4),        FK              //fuel identifier//
     spl_id             TEXT(3),        FK              //split identifier//
     snap_id            TEXT(6),        FK              //SNAP identifier//
     lev_snap           SHORT,                          //level of the SNAP identifier (1,2,3)//
     surr_id1           SHORT,                          //identifier of first surr.data type used//
     surr_id2           SHORT,                          //identifier of second surr.data type used//
     surr_id3           SHORT,                          //identifier of third surr.data type used//
     surfco_val1        DOUBLEFLOAT,                    //value of first coefficient//
     surfco_val2        DOUBLEFLOAT,                    //value of second coefficient//
     surfco_val3        DOUBLEFLOAT,                    //value of third coefficient//
     comment            TEXT(5)                         //comment identifier//
);

CREATE UNIQUE INDEX PrimaryKey
  ON surrform
(
    fuel_id                 ASC,
    spl_id                  ASC,
    snap_id                 ASC
);

CREATE TABLE unidef                                     //table of units definition//
    (uni_id             SHORT,                          //unit identifier//
     uni_name           TEXT(45),                       //unit name//
     uni_abbr           TEXT(15),                       //abbreviation of the unit name//
     usr_def            TEXT(1),                        //user defined flag (Y/N)//
     uni_typ            TEXT(1)                         //reserved for unit type//
);

CREATE UNIQUE INDEX PrimaryKey
  ON unidef
(
    uni_id                  ASC
);
```

CollectER database format version 4

New database format is version 4 format. It was created by changing actual database by:

- adding new codes into point source(facility) definition table
  - ipcc_code - reserved for IPCC code
  - nose_p - reserved for NOSE-P code
  - nace_code - reserved for NACE code
- adding new flag to lps_part table
  - lcp_flag - 0 – no LCP source, 1 – new LCP source, 2 – old LCP source

Name of the table “Convert” is changed to “Uconvert”. This was necessary due to use of new ADO Microsoft database interface, which replace older DAO interface, which will not be developed by Microsoft in the future.

New pollutants were added according to requirements of the NFR. List of added pollutants:
aldrin, chlordane, chlordecone, DDT, dieldrin, endrin, heptachlor, mirex, toxaphene, hexabromobiphenyl, polychlorinated biphenyls (PCBs), short-chained chlorinated parraffins, pentachlorophenol