

EMEP Database v2.1

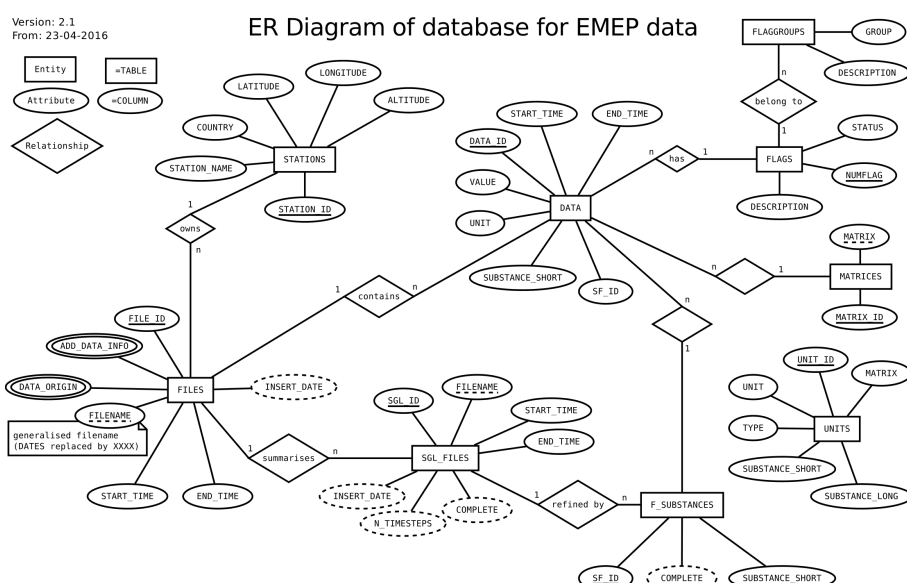
1. Content

- 1. Content
- 2. Entity Relationship Model
- 3. Downloads
 - 3.1. code, scripts, data
 - 3.2. ER Diagram
 - 3.3. mixed
- 4. Creating and Filling the Database
 - 4.1. SQL code for table creation
 - 4.2. SQL code to fill pre-filled tables
 - 4.3. SQL code for updating pre-filled tables
- 5. Database Usage
 - 5.1. R functions to fill reference tables (stations, units, flags, flaggroups)
 - 5.2. R functions to read-in EMEP text files
 - 5.3. SQL code for data export
- 6. Available Data
 - 6.1. Data from 2012 and 2013
- 7. Location of the Database

2. Entity Relationship Model

Explanation to ER diagrams

- **Squares** represent **tables**.
- **Ellipses** directly connected to squares are **columns** of these tables.
 - **Underlined text** represents the **primary key**. The primary key **uniquely** identifies a **row** in a table.
 - **Double ellipses** represent more than one column.
 - **Dashed ellipses** exist for debugging purposes.
- **Rhombi** indicate that a mapping between two tables exists and how many data sets of the one table correspond to how many of the other.
(E.g.: **One or more files** (n at the Rhombus connection **STATIONS** and **FILES**) contain the measurements performed at **one stations** (1 at the Rhombus connection **STATIONS** and **FILES**). But measurements in **one file** may not be performed at **two or more stations**.)
- **Dashed line** (no official ER symbol) indicate by which column to tables are connected to each other.



3. Downloads

red = DEPRECATED

3.1. code, scripts, data

[create_tables_v02.sql](#) (statements to create the tables)

[prefilled_tables_datafiles.zip](#) (data for tables which should be prefilled (flags, flaggroups, etc.); Ideally the script below downloads some of these tables on its own by downloading the data.), see [prefilled tables](#) for more details on the usage

[prefilled_tables_scripts.zip](#) (scripts to prepare and read in the data), see [prefilled tables](#) for more details on the usage

[emep_parser_v09_2014-04-07.zip](#) (reads text files exported from the official EMEP database into R variables), see R scripts for EMEP f or details on the usage

[emep_database_v12_2014-04-07.zip](#) (writes data into our EMEP database; requirement: [emep_parser_v08](#)), see R scripts for EMEP for details on the usage

[emep_testfile_2014-04-07.zip](#) (test file to read into the database with [emep_database](#) via the [emep_parser](#)), see R scripts for EMEP for details on the usage

[emep_utilities_UNVERSIONED_2014-04-07.zip](#) (some useful scripts and code parts)

3.2. ER Diagram

[erm_database_v02-1_2016-04-23_f ull_corr.pdf](#)

[erm_database_v02-1_2016-04-23_f ull_corr.svg](#)

3.3. mixed

Daniel's notes (may help if database breaks and the wiki documentation does not provide help): [emep.tex](#), [postgres.tex](#), [r.tex](#) (date: 01/04/2014)

[available_data_2012-2013_2014-05-28.xls](#) (Data from 2012 and 2013 which is already available in EMEP)

4. Creating and Filling the Database

4.1. SQL code for table creation

All statements to create the tables are also contained in the script [create_tables_v02.sql](#) in the download section. Changes with respect to version 2.0 are marked by a comment `/* NEW_v02.1` or `/* REMOVED_v02.1`.

create table units

```
CREATE TABLE units (  
    unit_id SERIAL PRIMARY KEY,  
    substance_long varchar(64),  
    matrix varchar(64),  
    substance_short varchar(16),  
    unit varchar(16),  
    type varchar(40)  
);
```

create table matrices

```
CREATE TABLE matrices (  
    matrix_id SERIAL PRIMARY KEY,  
    matrix varchar(64)  
);
```

create table flaggroups

```
CREATE TABLE flaggroups (  
    flaggroup smallint PRIMARY KEY,  
    description varchar(256)  
);
```

create table flags

```
CREATE TABLE flags (  
    numflag smallint PRIMARY KEY,  
    status char(1),  
    description varchar(256),  
    flaggroup smallint references flaggroups(flaggroup)  
);
```

create table stations

```
CREATE TABLE stations (  
    station_id character(7) PRIMARY KEY,  
    station_name varchar(64),  
    country character(2),  
    longitude numeric(9, 6),  
    latitude numeric(9, 6),  
    altitude smallint  
);
```

create table files

```
CREATE TABLE files (  
    file_id BIGSERIAL PRIMARY KEY,  
    file_name varchar(256),  
    station_id character(7),  
    start_time integer,  
    end_time integer,  
    data_definition varchar(32),  
    set_type_code varchar(10),  
    regime varchar(10),  
    station_code character(7),  
    platform_code character(7),  
    startdate date,  
    timeref time,  
    revision_date date,  
    component varchar(32),  
    unit varchar(16),  
    matrix varchar(64),  
    period_code varchar(4),  
    resolution_code varchar(4),  
    laboratory_code varchar(64),  
    instrument_type varchar(64),  
    instrument_name varchar(64),  
    method_ref varchar(64),  
    ext_lab_code varchar(64),  
    ext_meth_ref varchar(64),  
    add_qualifier varchar(64),  
    insert_date TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP  
);
```

create table sgl_files

```
CREATE TABLE sgl_files (  
  sgl_id SERIAL PRIMARY KEY,  
  file_id integer references files(file_id),  
  file_name varchar(256),  
  startdate date,  
  start_time_yyyyddd integer,  
  start_time_hh smallint,  
  end_time_yyyyddd integer,  
  end_time_hh smallint,  
  n_timesteps integer DEFAULT -1,  
  special_comment_present boolean DEFAULT FALSE, /* NEW_v02.1 */  
  complete boolean DEFAULT FALSE,  
  insert_date TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
  inserted_by varchar(64) DEFAULT current_user /* NEW_v02.1 */  
);
```

create table f_substances

```
CREATE TABLE f_substances (  
  sf_id BIGSERIAL PRIMARY KEY,  
  sgl_id integer references sgl_files(sgl_id),  
  substance_short varchar(16),  
  matrix varchar(64), /* NEW_v02.1 */  
  method_ref varchar(64), /* NEW_v02.1 */  
  complete boolean DEFAULT FALSE  
);
```

create table data

```
CREATE TABLE data (  
  data_id BIGSERIAL PRIMARY KEY,  
  file_id integer references files(file_id),  
  sf_id integer references f_substances(sf_id),  
  station_id character(7) references stations(station_id),  
  substance_short varchar(16),  
  matrix varchar(64),  
  unit varchar(16),  
  start_time_yyyyddd integer,  
  start_time_hh smallint,  
  end_time_yyyyddd integer,  
  end_time_hh smallint,  
  value double precision,  
  /* REMOVED_v02.1: numflag smallint references flags(numflag) */  
  numflag1 smallint references flags(numflag), /* NEW_v02.1 */  
  numflag2 smallint references flags(numflag), /* NEW_v02.1 */  
  numflag3 smallint references flags(numflag) /* NEW_v02.1 */  
);
```

4.2. SQL code to fill pre-filled tables

Some tables have to be filled after the creation of the database and before EMEP data is imported.

```
/* copy content into units */
\copy units (substance_long, matrix, substance_short, unit, type)
FROM '~/emep_units_4db';

/* copy content into flaggroups; NEEDS TO BE FILLED BEFORE 'flags' */
\copy flaggroups (flaggroup, description) FROM
 '~/emep_flaggroups_4db';

/* copy content into flags; 'flaggroups' NEEDS TO BE FILLED BEFORE */
\copy flags (numflag, status, description, flaggroup) FROM
 '~/emep_flags_4db';

/* copy content into stations */
\copy stations (station_id, station_name, country, longitude,
latitude, altitude) FROM '~/emep_stations_4db';

/* copy content into matrices */
\copy matrices (matrix) FROM '~/emep_matrices_4db';
```

The files which are needed above can be found at the page [EMEP database: prefilled table data](#).

4.3. SQL code for updating pre-filled tables

New flags may be defined from EMEP or flag descriptions (or status) may change as well as new stations may be defined. What do we do?

Situation: We have an outdated table called 'flags' and an updated table in file 'emep_flags_4db'.

Aim: We want to alter existing entries of 'flags' according to 'emep_flags_4db' and add new entries.

Problem: We cannot just remove the content of 'flags' and reread 'emep_flags_4db' because other tables are connected to the column 'numflag'.

These two pages provided some help: <http://www.techonthenet.com/sql/update.php> and <http://www.postgresql.org/docs/current/interactive/sql-update.html>.

update table flags

```
/* create temporary table */
CREATE TABLE tmp (
    numflag smallint PRIMARY KEY,
    status char(1),
    description varchar(256),
    flaggroup smallint references flaggroups(flaggroup)
);

/* read 'emep_flags_4db' into an empty temporary table */
\copy tmp FROM ~/emep_flags_4db

/* update column 'status' of existing entries */
UPDATE flags SET status = (SELECT status FROM tmp WHERE tmp.numflag =
flags.numflag);

/* update column 'description' of existing entries */
UPDATE flags SET description = (SELECT description FROM tmp WHERE
tmp.numflag = flags.numflag);

/* insert those entries into 'flags' which are in 'tmp' but not in
'flags' */
INSERT INTO flags (SELECT * FROM tmp WHERE tmp.numflag NOT IN (SELECT
numflag FROM flags));

/* remove temporary table; remove the 'spaces' before use - the wiki
does not allow this command to be saved :- ) */
D R O P  TABLE tmp;
```

Some useful further code:

further useful commands

```
SELECT * FROM tmp WHERE tmp.numflag NOT IN (SELECT numflag FROM
flags); /* find entries missing in 'flags' */
```

5. Database Usage

5.1. R functions to fill reference tables (stations, units, flags, flaggroups)

See [EMEP database: prefilled table data](#) or see data and scripts (prefilled_tables_*) in the download section.

5.2. R functions to read-in EMEP text files

See [EBAS files: R scripts for processing](#) or see scripts (emep_parser, emep_database and emep_testfile) in the download section.

5.3. SQL code for data export

some examples in [EMEP Database: PostgreSQL commands](#) (bottom, one R-script for download)

6. Available Data

country	station	in database, =< 2010	2011	2012	2013	comment
DE	all	YES	YES?	SOME	NO	
NL	all	YES	NO	NO	NO	rework online_IC measurements before read in
DK	all	YES	NO	NO	NO	
BE	all	YES	NO	NO	NO	
NO	all	YES	SOME	SOME	NO	problems with meteorological parameters and their flags
PL	all	YES	NO	NO	NO	
IS	all	YES	NO	NO	NO	
FR	all	YES	YES?	NO	NO	
IE	all	YES	NO	NO	NO	
SE	all	YES	SOME	SOME	NO	
UK	all	YES	YES?	SOME	NO	rework online_IC measurements before read in ; problems with mercury measurements

6.1. Data from 2012 and 2013

See which data set from 2012 and 2013 are already available in the download section ([available_data_2012-2013_2014-05-28.xls](#)).

7. Location of the Database

DB Server: rzsv2330

DB Name: OBSERVATIONS

Administrator: Armin Aulinger

see ... TODO