Introduction to OpenVigil2

author: Hans-Joachim Klein openvigil@pharmakologie.uni-kiel.de

2022-08-18

In this summary we describe the structure and the functionality of OpenVigil2 from a more technical point of view. We discuss how the various changes of the data types in LAERS/FAERS files are handled and consider extensions in functionality added to OpenVigil2 recently. It is assumed that the reader is familiar with the FDA nomenclature used in connection with adverse event reporting and with the basic concepts of relational database systems.

The database schema

FAERS[1] data of interest for our evaluations are stored by OpenVigil2 (OV2, for short) in a database using the relational database system PostgreSQL. The schema of the database has been designed to reflect the main concepts of FAERS data in different types of tables (relations) and to link data in these tables to data in tables which are filled with information coming from other sources such as the DrugBank, the drugs@fda Data Files, or the MedDRA index. Problems with the use of these data arise from changes of the structure of data in successive versions as well as the violation of upward compatibility because of changes in the nomenclature (e.g.: see Asparagines as an example for DrugBank data below).

With respect to FAERS data we are facing similar problems with changes of data structures over the years. As a consequence, different routines for the extraction of data from FAERS files have been implemented and extensions of the database schema as well as changes of types of attributes have been made. These changes are done in a way which allows to use all data from the beginning of their publication in 2004 until present without changes of the evaluation routines.

Remark: For better readability we denote names of tables in upper case letters and attributes of tables in italics with the first letter capitalized. Names of fields in FAERS files are written using lower case letters in italics. Furthermore, we speak of attributes and attribute values when we refer to entries in tables.

For clarity we have organized the SQL database tables into four subschemas: DRUGBANK, DRUGS_AT_FDA, FDA, and PUBLIC. The first three subschemas are used for the organization of the loading and preprocessing of data from the DrugBank, Drugs@FDA, and the different FAERS files. The subschema PUBLIC contains the tables with all data needed by our evaluation algorithms.

In the subschema DRUGBANK, there are two tables: DRUG_CHARACTERISTIC and MIXTURE_INVALID. The first table contains indicators for salt forms of drugs such as

[1] LAERS data are included

acetate, carbonate, or maleate. These indicators are used by OV2 to detect notations of drugs given in salt form. If components of a mixture have different salt forms, this mixture is not accepted by OV2. The information about this rejection is stored in the second table for further analysis by the user.

Tables in the subschema DRUGS_AT_FDA are PRODUCT, MATCHING_ACTIVEINGREDSPLIT, and MATCHING_CHARACTERISTICSPLIT.
In PRODUCT all names of products together with their active ingredients provided in the Drugs@FDA file PRODUCTS are collected. The other tables are used to store intermediate results which are obtained by the analysis and eventually by the splitting of entries in the fields *name* and *activeingred* of PRODUCTS.

The subschema FDA is used for storing data coming from the FAERS files together with results of their analysis. Tables DEMO, DRUG, INDI, OUTC, REAC, RPSR, and THER are used to record the data contained in the corresponding FAERS files. A simple copying is not sufficient since the data format of FAERS files has changed several times and we want to have a fixed structure in our database for all quarterly data. Furthermore, the content of fields in the FAERS files is often not in a form which is compatible with the types of our database schema. Other tables help to organize the necessary processing of the original data such that they can be transferred to tables of the main subschema PUBLIC. Some tables provide information why reports have not been accepted. In some cases this allows to 'tune' the processing of original data. Details are discussed below.

The subschema PUBLIC includes all tables with 'production data', i.e. data accepted by the evaluation of FAERS data according to our strategies for cleansing original data.

Let us first consider the types of tables for recording DrugBank data of interest for our system: DRUG, SYNONYMS, PHARMAPRODUCT, PRODUCT, ATC_CODE, and SUBSTRUCTURES.
PK shall denote the primary key of a table. If no primary key is given, the primary key consists of all attributes or the table.

The attributes of DRUG are *Drugname* (PK), *Drugbank_id*, *Ddd* (Defined daily dose), and *Lastupdate*. Values in these attributes are taken from the DrugBank data without change.

SYNONYMS uses the attributes *Name* (PK) and *Drugname* to assign synonyms as provided by DrugBank data to the corresponding drug name in the DRUG table.

PHARMAPRODUCT has the attributes *Brandname*, *Producer_name*, *Form*, *Salt*, *Biologic*, *Type*, *Enantiomer_rs*, *Enantiomer_dl*, and *Enantiomer_plusminus* with the following meaning:
Brandname: brand name of the medication
Producer_name: manufacturer of the pharmaceutical product
Salt: salt form (e.g.Haloperidol intensol has salt form lactate)
Form, *Biologic*, *Type*, *Enantiomer_rs*, *Enantiomer_dl*, *Enantiomer_plusminus*: empty, for future use.

PRODUCT has the attributes *Brandname* and *Drugname* for connecting pharmaceutical products in PHARMAPRODUCT with the corresponding drugs in DRUG.

In table ATC_CODE with attributes *Code* (PK) and *Drugname* the ATC code of drugs can be found if provided by the DrugBank.

Tables SUBSTRUCTURES with attribute *Name* and COMPONENT with attributes *Name* and *Drugname* store information about substructures of drugs.

All data in these tables result from the inspection of data offered by the DrugBank and Drugs@FDA as well as from additions which can be made by users of OV2.

FAERS data are stored in tables reflecting the different files of the FAERS data collection. Report data as provided in the FAERS DEMO files are stored by OV2 in the table REPORT. The list of fields in DEMO files as well as the data type of some fields has changed several times over the years. Some fields have been removed, e.g. *death_dt* (from 2012Q3 onwards), *image* and *confid* (from 2012Q4 onwards), others have been added, e.g. *caseversion* (from 2012Q4 onwards), *auth_num*, *lit_ref*, and *age_grp* (all from 2014Q3 onwards). Renaming has also been done: from *isr* to *primaryid*, from *case* to *caseid*, and from *sex* to *gndr_cod* (in 2012Q4).

In REPORT all fields up to *age_grp*, *auth_num*, and *lit_ref* are represented as attributes partly with data types different from the FAERS data types in order to improve the efficiency of evaluation. Additional attributes are *Age_years*, *Age_days*, and *Age_hours* which are used to store values calculated from the content of the fields *age* and *age_cod* in the DEMO file. Furthermore, an attribute *Event_dt_type* has been added in order to handle the partiality of event dates in the field *event_dt* which is allowed since 2013Q3. We think it is a better idea to unify the format of dates in the same way as it has been done before 2013Q3 and to add the information which part of the date is missing in the new attribute *Event_dt_type*. If just the year is given we add '01' for month and day and denote this by 'Y' in the attribute *event_dt_type*. If year and month but not the day are given, we add '01' for the day and 'M' in the attribute event_dt_type. This approach avoids the loss of data quality (see README.DOC file in 2013Q1) without introducing partiality.

A major difference in the handling of keys for reports is reflected in the change of the fields *isr* and *case* to *primaryid* and *caseid*, respectively, and the addition of the field *caseversion*. ISR stands for Individual Safety Report number. Before 2012Q4 every line in the DEMO file has been identified by the value in the field *isr*. The values in the fields *isr* and *case* have been generated independently. An ISR is not 'speaking' i.e. it carries no information up to the unique identification of a report allowing to connect report data in different files. Additionally, a character 'I' or 'F' in the field I_F_COD provides the information whether the report is the first one of a case ('initial') or a report added for the case later on ('follow up'). Starting in 2012Q4, the identifier of a case (in *case_id*) is used to build the ISR, i.e. the identification of a report, by appending the value in the field *caseversion* to the identifier of the case. The resulting value can be found in the field *primaryid*. Hence, the simplest way to deal with the necessary change in the OV2 database is to use the content of the field *primaryid* as ISR value in the attribute *Isr* for data from 2012Q4 onwards.

In the FDA file README.PDF of April 2015 it is claimed that "By using the concept of case ids and case versions the FAERS QDE output files will always provide the latest, most current, Version of a Case available at the time the QDE is run". This characterization must not lead to the conclusion that only the highest versions of a case should be considered for evaluation of the FAERS data. Higher versions may not be considered exclusively as corrections or completions of former reporting. We find reports for a case documenting the course of a treatment (case_id 3431477 ..) including the change of medication (Salbutamol replaced by Levothyroxine sodium), the history of events (case_id 5854813, medication

unchanged), the collection of data probably originating in a study (case_id 6207022, 2007Q1 and 2007Q2: many different ages and gender code female as well as male), or the history of complains (case_id 6586742: intra-uterin contraceptive device expelled+iucd complication (isr 5670429), iucd complication (5743745), pharmaceutical product complaint (5856010)).

We decided to store all reports passing our quality check and not to delete older reports of a case when a new one comes in for the case. Besides the reasons presented above there is a problem with comparability of studies. Since the database is growing with quarterly data deleting older reports of cases could change results of former evaluations such that reproducibility is no more guaranteed. To avoid this problem, we allow to perform the evaluation (i.e. the counting of reports fulfilling the specified conditions) at the level of cases or by adding the restriction that only most recent reports are taken into account. The latter condition mimics the deletion of reports which are not the latest one of a case (highest version in the newer data).

The lines of the FAERS DRUG file are stored in the table DRUGUSAGE. There are two possible forms of connection between the tuples in DRUGUSAGE and the tuples in the table DRUG: If a brand name is detected, a look into the table PRODUCT is sufficient to find all its ingredients; if no brand name is given in a tuple *t*, we need to connect *t* with all drugs specified in *t* to the corresponding tuples in DRUG. This connection is realized by a tuple in the table D_APPL which has the attributes *Drug_seq* and *Drugname*. They are the key attributes of the tables DRUGUSAGE and DRUG, respectively.
The attributes of the table DRUGUSAGE are:

Drug_seq: primary key. Built from the content of the concatenated fields *primary_id* and *drug_seq* of the DRUG file in data from 2012Q4 onwards; for earlier quarters values are taken without change from the field *drug_seq* in the DRUG file.
A problem consists with the uniqueness of this primary key in data of recent quarters. There are values in the field ‘drug_seq’ in DRUG files of quarters before 2012Q4 which are identical with the concatenated key from DRUG data in 2012Q4 onwards.

Example: Report 4678753 from 2005Q1 has drug_seq value 1006065735
Report 100606573 from 2018Q4 has concatenated key 1006065735

We therefore extend each key generated for these newer quarters by the string ‘00’ (since these values are casted to the type bigint when transferred to the database this means a multiplication by 100). A single ‘0’ is not sufficient because there can be more than 9 entries with the same *primary_id* value in a DRUG file.

Example: Look at the report with ISR = 60892262 in 2013Q2. There are eleven entries in the corresponding DRUG file for this ISR. Among the concatenated values of the fields *primary_id* and *drug_seq* are ‘608922621’ and ‘6089226210’. Extending ‘608922621’ with ‘0’ would result in the violation of the key condition if ‘60892262’ shows up as ISR in a later quarter having 10 or more lines in the corresponding DRUG file. There are no reports with 100 or more lines in a DRUG file hence the addition of ‘00’ is sufficient to guarantee the key property.

Remark: It must be mentioned that in quarters before 2012Q4 values for the identification of drug usages have not been assigned as systematically as in recent FAERS files. Insofar problems could occur with the uniqueness of keys even when our approach is applied. The

database system would reject the insertion of a report with such a uniqueness problem. The method for guaranteeing uniqueness of keys would have to be adapted appropriately.

Brandname: name of the pharmaceutical product if provided and known to OV2.

Role_cod, *Val_vbm*, *Route*,*Dose_vbm*, *Dechal*, *Rechal, Lot_num*, *Exp_dt*, *Nda_num*: same meaning as defined for FAERS data.

Dailydosis: the field *dose_vbm* in DRUG files contains information on the dose, the frequency of use and the form of administration of the medication in the form of free text. By using appropriate regular expressions a daily dose in milligram is computed from the given text whenever possible.

Isr: the identifier of the report the DRUGUSAGE tuple belongs to.

Drugname_orig: for files before 2014Q3 the value of this attribute is the content of the field *drugname* which may contain several drug names or a brand name. From 2014Q3 onwards we have to take into account that pharmaceutical product names and product ingredients should be given in separate fields: *drugname* and *prod_ai*. The dealing with this change and problems with data filled in by users in these fields are discussed below.

Tuples in the table DRUGUSAGE can be connected to the drugs involved either by the table PRODUCT or by the table D_APPL, not by both. If a pharmaceutical product could be identified by inspecting the text given in the field *drugname* of a line in the DRUG file, a corresponding brand name is inserted in the attribute *Brandname* of the corresponding tuple in the table DRUGUSAGE. If a brand name could not be identified but the name of one or more drugs, corresponding tuples are inserted in the table D_APPL. If neither a brand name nor drug names could be identified the complete report is rejected.

Information on events, outcomes, and report sources (FAERS files REACquarter, OUTCquarter, RPSRquarter) is related directly to the corresponding report. Tables REP_EVENT, REP_OUTCOME, and REP_RPSR realize these relationships by combining the values in the key attribute of REPORT (*Isr*) and those of the tables EVENT, OUTCOME, and RPSR (*Pt*, *Outc_cod*, *Rpsr_cod*, resp.). Additional attributes in these tables are *Sysorgclass* for EVENT and *Meaning* for OUTCOME and RPSR. Values in these attributes are taken from the corresponding FAERS files.

Information on indications is related to the usage of drugs, i.e. not directly to the reports. Since there is no source with standardized terms for indications we read the values in the field *indi_pt* of the FAERS DRUG quarterly file and store them without duplication in the table INDICATION. Attributes of INDICATION are *Indi_pt* (PK) and *Anatomgrp*. There can be several indications for a single drug usage and an arbitrary number of drug usages for a single indication (a so-called m-to-n relationship). Hence we combine in the table USED the key attributes of DRUGUSAGE and INDICATION to assign indications to drug usages of a report.

The information on a therapy is related to a drug usage as well. The restriction here is that a single therapy is assigned uniquely to a drug usage. For a single drug usage, however, several therapies may have been prescribed. Hence we do not need an intermediate table for realizing the relationship between drug usages and indications but the addition of a ‘counting’ attribute (*T_id*) to the table THERAPY is sufficient to realize the 1-to-n relationship. Further attributes of THERAPY are *Start_dt*, *End_dt*, *Dur_cod*, *Dur* from FAERS file

THERAPYquarter and *Dur_years*, *Dur_days*, and *Dur_seconds* derived for analytical purposes.

For approximately 21% of the drugs stored in the DRUG file of a recent version of the OV2 database an ATC code is given. It is stored in the table ATC_CODE together with the drug name. Attributes of ATC_CODE are *Code* and *Drugname*.

In the DrugBank, substructures are often given for drugs. If so, a tuple in the table COMPONENT (attributes *Name* and *Drugname*) is inserted by OV2 with the name of the substructure and the name of the drug. In the table SUBSTRUCTURES (attribute *Name*) all names of substructures are collected. Thus it is possible to find all drugs for a given component.

In case pharmaceutical products are mixtures it may be possible that one or more ingredients have no counterpart in the tables DRUG or SYNONYMS at the time of the loading of DrugBank or Drugs@FDA data. The table INCOMPLETE with attribute *Name* contains all such drugs. The table UNCLASSIFIED with attributes *Name* and *Brandname* connects the drugs with the corresponding pharmaceutical product. Currently there are 1564 entries in INCOMPLETE not contained in DRUG.

Caveat:

What has to be taken into account when interpreting data in these tables is that information is missing for many reports (see percentages of missing values given with the FAERS files). Furthermore, OV2 uses a strict cleansing process rejecting a report if relevant information cannot be interpreted uniquely based upon the given information on drugs, synonyms, and pharmaceutical products.

A further problem is that the data type of some fields has changed over the time (see *route*, *mfr_num*, *dose_vbm*, *lot_num* in 2012Q3, revised in October 2014). In some cases we adapted the types of the corresponding attributes; in other cases where the very long content of a field is of no interest for our evaluation algorithms we cut strings which are longer than allowed by the type of the attribute.

Loading basic data

Our main source for drug data is the content of the DrugBank mentioned above (offered as ASCII and as XML file at https://www.drugbank.com). We analyze the XML file by inspecting the following elements:
drug, drugbankid, name, substructure, synonym, brands, brand, international_brands, international_brand, atccode, salt, mixtures, and mixture.

The content of the DrugBank is constantly extended and modified in new versions. Sometimes, drug names are replaced and the association of DrugBank identifiers with drugs is changed.

Drugbankid (DrugBank Accession Number in the DrugBank) cannot be taken in our database as a key value for drugs; it is included in the information about drugs for having a direct link into the DrugBank but should be used carefully. DrugBank Accession Numbers have changed over time losing their uniqueness such that the content of the OV2 database wrt them may not be up to date. Another reason is that we consider also drug data from Drugs@FDA to complete the information about drugs.

Example: Asparaginese with 2016-08-17 as last update in a former DrugBank version has identifier DB00072 which is later (update 2021-01-01) the identifier of Trastuzumab. The name Asparaginese has been replaced by Asparaginase escherichia coli (DB00023) and is now a synonym of this drug name.

OV2 has to be aware of such changes. We have to preserve the former notions because they are used in older FAERS data. Drug names in the table DRUG are never changed and are used as key values. A replacement of a name in a new version of the DrugBank is realized by inserting the new name and its DrugBank_id in the DRUG table when loading the new version. The tuple in DRUG with the former drug name is not modified. The old and the new name of the drug are connected by inserting the old name as synonym of the new name in the SYNONYMS table. This occurs automatically if the old name is declared as synonym of the new name in the DrugBank (see below). If this is not the case, both drugs are considered as different.

Example continued: When inserting a report with Asparaginase as drug name in one of its DRUG file entries, the connection to the DRUG table by the D_APPL table is by this drug name, i.e. by Asparaginase. (see lines 110ff in module AersDrugnameMappingSql). Asparaginase, however, has no synonyms, hence we have no symmetry. As a consequence, the new name Asparaginase escherichia coli should be used in searches.

Brand names are inserted into the PHARMAPRODUCT table and connected via the PRODUCT table with the corresponding names of its ingredients in the DRUG table.

A special treatment is applied to drug names representing salts. In the table DRUGBANK.DRUG_CHARACTERISTIC we provide notions for salts such as Maleate or Dihydrochloride. If such a salt term is found together with the name of a known drug in an entry of the DRUG file, the name of the drug is stored together with the salt term as synonym of the drug.

In a second step, data from Drugs@FDA can be checked for names of drugs and products not yet known to OV2 because they are not provided by the DrugBank. Names of drugs are presented in the field *ActiveIngredient* and names of pharmaceutical products in *DrugName*. The values in these fields of the FAERS XML file Products are stored in the tables DRUG and PHARMACEUTICALPRODUCT, respectively, and connected by inserting them in the PRODUCT table. If during the evolution of the OV database data on drugs have been provided earlier by Drugs@FDA than by the DrugBank the value of the *Drugbank_id* as well as that of *Lastupdate* is NULL. This is a further reason why *Drugbank_id* cannot serve as key attribute.

The problem with duplicates

In
https://www.fda.gov/drugs/surveillance/questions-and-answers-fdas-adverse-event-reporting-system-faers (last access April,19th,2022)
it is stated that “There are also duplicate reports where the same report was submitted by a consumer and by the sponsor.”

This formulation is a bit problematic since ‘the same report’ and ‘duplicate reports’ have to be specified precisely in the context of the data sources. What is meant is that reports may be submitted referring to the same observation in connection with a case. The case identifier

and the case version, do they have to be the same? This would contradict the key concept. In which fields entries may differ? Surely this is the case in the fields *mfr_num*, *mfr_sndr*, and, probably, *fda_dt* of the DEMO file. Does every information related to the reports and stored in the other files (DRUG, INDI, REAC, ...) has to be identical or is incompleteness allowed but contradictions not. A straightforward solution to handle these problems does not seem to exist. We therefore decided not to solve the duplicate problem automatically but to generate messages in case of problematic cases and to offer an interface for finding a solution.

Cleansing FAERS data

In the following we describe the processing of FAERS data by OV2, i.e. how data in the FAERS files are transferred into the OV2 database and why they are eventually rejected. In some cases we use intermediate tables in the subschema FDA for the processing of data before they are inserted into the tables of the main subschema PUBLIC. Above it has already been described how data in FAERS files INDI, OUTC, REAC, RPSR, and THER can be transfered into the OV2 database. The main challenge with the transfer of data arises with the data in the FAERS files DEMO and DRUG. The handling of data in DEMO files by OV2 has already been discussed. The changes of the type of DEMO files over the years are taken into account by checking the header line of these files and by adapting the loading procedure appropriately. The type of the REPORT table in OV2 has been extended to take the changes into account but the meaning of the 'old' attributes has not been changed. This proceeding allows the loading of 'old' data after changes of the type of DEMO files.
The main application of OV2 is the computation of over- and underproportionality for drugs and products associated with one or more adverse events based upon contingency tables (see http://openvigil.sourceforge.net/doc/DPA.pdf).
Since this kind of evaluation refers to the existence of relationships between drugs or products and adverse events in single reports as well as to their non-existence we decided to apply a strong regime for the consideration of reports for evaluation: If OV2 cannot interpret uniquely an entry in a FAERS DRUG file the corresponding report is not considered at all, i.e. the data being part of this report are not accepted for the OV2 database. Take as an example a report *r* with two entries *e1* and *e2* in the DRUG file. Assume that a value in the field *drugname* (older quarters) or in the fields *drugname* and *prod_ai* (quarters from 2014Q3 onwards) in *e1* could be assigned to a product or to a drug *d1* known to OV2 but the value in the field in *e2* could not be interpreted as a known product or drug (for special interpretations see below). Adverse events in FAERS data are associated with reports, not directly with drugs or pharmaceutical products. If we insert *r* with *e1* as single entry it follows that for any drug or product known to the database up to *d1* it is assumed for evaluation that it is not reported in *r*. Furthermore, *d1* would be considered as the only drug reported in *r1* as related to the adverse events associated with *r1*. This means that unknown information is interpreted as sure information.
In case a report has a single entry in the DRUG file (about 59% in 2020Q4, for example) and no name of a drug or a pharmaceutical product could be assigned by OV2, it is obvious that this report cannot contribute to the relationship between drugs or pharmaceutical products and adverse events. It has to be ignored.

Remark

The so-called *open world assumption* should naturally underlie the processing of data in FAERS files. This means that if information on a drug or product/adverse event relationship

for some report case is missing, for example, the fact that this relationship does not exist for the report cannot be derived with certainty. For such a conclusion we would have to be sure that no information is missing in our observation data. Nevertheless, we usually take such facts as not given when evaluating the data set, i.e. we assume that a statement like 'no other drugs or products than the given ones are involved in the treatment of the patient' is valid. This means that evaluation is done under the so-called *closed world assumption* which should always be taken into account.

The special value 'placebo' (in 2020Q4 there are 632 reports with this value in the field *drugname* and 'unspecified ingredient' in the field *prod_ai*) allows to accept DRUG entries although no product or drug can be assigned. Under the assumption that each placebo is an inactive substance, an entry with this value as drug name should not cause a complete report to be not accepted. If all other entries for this report in the FAERS DRUG file are accepted, the report should be accepted as well though no relationship between the 'placebo entry' and drugs can be established. Among other values which cannot be assigned to a concrete product or drug are 'multi-vitamins', 'multi-vitamin', and 'vitamin tab'. An entry in the FAERS DRUG file with one of these values in the field *drugname* could be accepted nonetheless: Since there is no relationship to a product or drug, no wrong conclusion concerning the relationship between drugs or products and adverse events can be drawn. There are about 5300 reports in the actual database with such values in the field *drugname* of DRUG files.

How these special data constellations can be taken into account by the software is discussed below in connection with the use of the table MISSPELLINGS.

In the discussion of the analysis of FAERS DRUG data we have to distinguish between the older form up to 2014Q2 and the new form in subsequent quarters. Let us consider first the older form.

In the older form the field *drugname* has been intended to contain information on drugs or on pharmaceutical products. Providers of report data did often insert free text like 'AVE5026 OR PLACEBO', 'CS-866 OR PLACEBO (OLMESARTAN MEDOXOMIL) (TABLET) (OLMESARTAN', 'IPILIMUMAB OR PLACEBO' (all 2009Q1). Mixing of names of pharmaceutical products and drug names sometimes in parentheses as well as lists of names separated by slashes or 'and' can be found quite often (Examples: 'TRIATEC (1.25 MG, TABLET) (RAMIPRIL)',
'PLETAL (CILOSTAZOL)' (both in 2013Q1))

Sometimes a slash may be read as 'or' like in 'SORAFENIB/ PLACEBO' (2009Q1); sometimes it is used to separate names of ingredients of combi predicates like in ABACAVIR SULFATE/LAMIVUDINE (2009Q1). We also find entries like 'BLINDED AmBisome' (2013Q1) coming from a clinical trial or 'VX-950 (Telaprevir)', i.e. an external name (DrugBank nomenclature) together with a brand name (2013Q1).

There are numerous such examples. As a consequence, a simple text search for the occurrence of substrings denoting drugs or products in the data field *drugname* is not advisable. What is needed is an analysis of the content of the field. Since both names of drugs and names of pharmaceutical products occur in the field even in the same entry we decided to give preference to products over drugs. If the name of a pharmaceutical product can be derived without doubts it is preferred to the name of an ingredient of the product

derived as well from the content of the field *drugname*. This means that the brand name is filled in the attribute *Brandname* of the DRUGUSAGE tuple under consideration but no tuple is generated for the table D_APPL.

The inspection of the entries in the field *drugname* of the FAERS file DRUG is done on the table FDA.DRUG where all lines of the file DRUG are represented as tuples with values of some attributes possibly shortened. It starts by looking for parts of the attribute *Drugname* which can be assigned to product or drug names or to synonyms in the SYNONYMS table or names in the MISSPELLING table or which can be identified as salts. This is done in several steps by applying different regular expressions in SQL statements. In the first step, the table FDA.DRUG_DRUGNAME_SPLIT_INITIAL with attributes *Drug_seq* and *Drugname_part* is filled with values from tuples of the FDA.DRUG table having a string as value in the attribute *Drugname* which
- does not contain parentheses or brackets. After applying 'lower' it has to fit the regular expression '^([a-z0-9 %,+/-]+)$'.
- contains parentheses or brackets. In this case the entry is splitted interpreting the strings in parentheses and brackets as single names of drugs if they fulfill the regular expression '^([a-z0-9 %,+/-]+)$'. Splitting is done by using the SQL functions regexp_matches, trim, and unnest (see PostgreSQL).

Example: In the FAERS DRUG file of 2013Q1 we find for primaryid 68815752 in the field *drugname* the entry HALOPERIDOL (HALOPERIDOL) (HALOPERIDOL). Analyzing and splitting gives us a single entry for HALOPERIDOL in the table FDA.DRUG_DRUGNAME_SPLIT_INITIAL.

In the second step, tuples in the resulting table FDA.DRUG_DRUGNAME_SPLIT_INITIAL are checked for bad endings such as /29372/ which are quite often to find. These endings are deleted and the resulting string is inserted into the table FDA.DRUG_DRUGNAME_SPLIT_WITHOUT_BAD_ENDINGS with attributes *Drug_seq* and *Drugname_part*.

Example: In the DRUG file of 2013Q1 we find for *primaryid* 69909462 in the field *drugname* the entry VITAMIN D /00107901/ (ERGOCALCIFEROL). We get in this case two tuples in FDA.DRUG_DRUGNAME_SPLIT_WITHOUT_BAD_ENDINGS with vitamin d and ergocalciferol as values in the attribute *Drugname* of the table together with an identical drug sequence number.

In the third step, tuples in the table
FDA.DRUG_DRUGNAME_SPLIT_WITHOUT_BAD_ENDINGS
are checked for strings in the attribute *Drugname_part* which can be divided into smaller parts by looking for occurrences of 'and ', ' with ', ' + ', and ' w/ ' as well as ',' or '/' if no number or unit of measuring like 'ml' or 'kg' is following. '/w ' is often used for 'with '.
If smaller parts can be identified they are stored together with the value in *Drug_seq* into the table FDA.DRUG_DRUGNAME_SPLIT. Like the other tables this table has the attributes *Drug_seq* and *Drugname_part*.
The often used backslash instead of 'and ' is replaced by 'and ' when filling the FDA.DRUG table with the data of the DRUG file.

Example: Consider again the DRUG file in 2013Q1. For 87294095 in the field *primaryid* we find the entry 'HYDROCHLOROTHIAZINE\LISINOPRIL' which is replaced by 'hydrochlorothiazine and lisinopril' in the corresponding tuple of FDA.DRUG. Hence we get in

this step two tuples with the same drug sequence number in the table FDA.DRUG_DRUGNAME_SPLIT.

In the last step, entries (i.e. strings) in the attribute *Drugname_part* of FDA.DRUG_DRUGNAME_SPLIT are checked whether they represent drug names, synonyms of drug names, or names of pharmaceutical products. If so, the entry is stored in the attribute *Drugname* of the table FDA.DRUG_APPLICATION together with the drug sequence number and the type, i.e. drug, synonym, or product.
If no assignment is possible, the table MISSPELLINGS (attributes are *Name*, *Misspelling* (PK), and *Type*) is inspected for a possible replacement of the entry by a name known to the OV database as drug, synonym of a drug, or pharmaceutical product. The information about the type of the entry (product, drug, or nothing) is stored in MISSPELLINGS together with the name which represents the corrected form of the entry in case of the types product and drug. If the given type is 'nothing', the attribute *Name* contains no value. An automatic correction of misspellings without reference to a table like MISSPELLINGS is not applied because of possible ambiguities.
There is a special treatment of entries in parentheses or brackets which represent unconventional information like 'no concurrent medication', 'nos', or 'placebo'. They do not necessarily lead to the rejection of a report though no assignment of a name of a drug or product is possible. By storing such an entry with the type 'nothing' in the table FDA.DRUG_APPLICATION no link to the table DRUG is installed by a tuple in D_APPL or by a brand name.

The content of the table MISSPELLINGS is user-defined. A setting is provided with the software but can be changed arbitrarily. Changes of the table become effective with the next loading of FAERS data. In order not to get a different treatment of the quarterly data, the table should be fixed at the beginning of loading the database.

If both the name of a drug and the name of a product are found in an entry, it is checked whether the drug is an active ingredient of the product. If so the product name gets preference to the drug name, i.e. only the product name is stored in the table FDA.DRUG_APPLICATION.

Example: Consider the string 'TRIATEC (1.25 MG, TABLET) (RAMIPRIL)' from above. The product name TRIATEC is more informative than the drug name RAMIPRIL, since Ramipril is an active ingredient of Triatec.

Drug names denoting salts get a special treatment. By consulting the table DRUGNAME.DRUG_CHARACTERISTIC entries which could not yet be assigned are checked for containing an extension typical for salts like hydrochloride or sulfide (e.g. Tramadol hydrochloride). These entries are reduced to the basic name (Tramadol in the example). This basic name is analyzed like the entries in *Drugname_part* before. What has to be taken into account is that entries with more than one salt term are possible. When the basic term can be identified as drug name or as synonym of a drug name, the complete entry is added to the table SYNONYMS together with the basic term.

Above has been mentioned that since October 2014 information on products and drugs in the FAERS DRUG file is given in two fields: *drugname* and *prod_ai*. According to its specification by the FDA, the field *drugname* should contain the name of a pharmaceutical product and *prod_ai* the name of a drug which is considered as the active ingredient of the product. Though by this separation the quality of the information in FAERS DRUG files has

strongly increased, it is not sufficient to simply take the content of the field *drugname* and to look for a direct assignment of a pharmaceutical product or, in case no such product is known to OV2, to take the value in the field *prod_ai* as active ingredient and to try to find a corresponding drug name in the database. We provide some examples to demonstrate the problems which may arise.

Consider the following part of the DRUG file in the third quarter in 2017 which gives you an impression of the variability of entries in the fields *drugname* and *prod_ai* (for better readability empty lines have been inserted and entries in the fields *drugname* and *prod_ai* are shown in bold letters):

primaryid$caseid$drug_seq$role_cod$**drugname$prod_ai**$val_vbm$route$dose_vbmcum_dose_chrcum_dose_unit$dechal$rechallot_numexp_dtnda_numdose_amt$dose_unit$dose_form$dose_freq"

1001443212$10014432$82SS**VITAMIN D$CHOLECALCIFEROL**$1$Unknown$$$$U$U$$$$$$$

1001443212$10014432$83SSAMINOBENZOIC **ACID\CHOLINE BITARTRATE\FOLIC ACID\INOSITOL\VITAMIN B COMPLEX$VITAMINS**$1$$$$$U$U$$$$$$$

1001443212$10014432$84SS**VITAMIN B (UNSPECIFIED)$VITAMIN B**$1$Unknown$$$$U$U$$$$$$$

1001443212$10014432$85SS**CALIUM (CALCIUM CARBONATE)$CALCIUM CARBONATE**$1$$UNK$$$U$U$$$$$$$

1001443212$10014432$86SS**VITAMIN D$CHOLECALCIFEROL**$1$$$$$U$U$$$$$$$

1001443212$10014432$87SS**docosahexaenoic acid (+) eicosapentaenoic acid$DOCONEXENT\ICOSAPENT**$1$$$$$U$U$$$$$$$

1001443212$10014432$88SS**SENOKOT$SENNOSIDES**$1$$$$$U$U$$$$$$$

1001443212$10014432$89SS**SULFAMETHOXAZOLE$SULFAMETHOXAZOLE**$1$Oral$$$$U$U$$$$$$$

1001443212$10014432$90SS**sulfamethoxazole (+) trimethoprim$SULFAMETHOXAZOLE\TRIMETHOPRIM**1Unknown$$$$U$U$$$$$$$

As this example demonstrates, we cannot assume that entries in the attribute *Drugname* are always names of products and that entries in the attribute *Prod_ai* are always names of active ingredients of the products. As a consequence we apply the same analysis to the values in these attributes as for the attribute *Drugname* in the reports before 2014Q4.
To keep changes in the software as small as possible, we proceed as follows: When loading data from the FAERS DRUG file into the table FDA.DRUG we store the values in the field *drugname* under the attribute *Prod_ai* and vice versa. At the end of the loading we check the values in the attribute *Prod_ai* whether they represent a pharmaceutical product known to OV2. If this is the case, we replace the content of the tuple under consideration in the attribute *Drugname* by the value of the tuple in the attribute *Prod_ai*.
By this proceeding we can go on as before where a single field (*drugname*) has been used for storing information on drugs involved in a treatment. A problem may be that the content of an entry in the field *prod_ai* of the DRUG file does not represent an active ingredient of the

product given in the field *drugname*. In this case the product ‘reigns’ over the content in the field *prod_ai*.

Preparation for enhanced evaluation

We discussed already the advice given with FAERS data that only the last reports for a case should be considered for evaluation. In order to allow this restriction for the evaluation by OV2 we offer the variant ‘ISR (most recent)’ additionally to the variants ‘case (entire case)’ and ‘ISR (unique reports)’. A new table MRISR has been added to the database to handle this variant efficiently. Its attributes are *Isr* and *Case_id*. The table has to be filled after every loading of new report data. Filling is done by inspecting the table REPORT and choosing all tuples which have the highest value in the attribute *Isr* for all reports of the case they belong to. The values of these tuples in the attributes *Isr* and *Case_id* form the tuples of MRISR. The following SQL commands can be used:

```
create table public.mrisr
(isr bigint not null,
case_id bigint,
constraint mrisr_pk primary key (isr)
);
```

After every loading of FAERS data:

```
delete from mrisr;

insert into mrisr select max(isr), case_id from report group by
case_id;
```

Remark: In the OV2 database with reports up to the second quarter of 2022, for 625 cases the maximum of its ISR values belongs to a report which has not the latest fda_dt value. The report with the latest fda_dt value of such a case, however, is often a duplicate of the report with the maximal ISR value. Hence the wrong assignment to the table MRISR in these cases seems to be acceptable.

***Integration of the MedDRA* index**

According to the Readme.doc files of FAERS (and LAERS) data, adverse events in the REAC files should be coded as low level terms (LLTs) of the MedDRA index. Up to a very small number of entries this is the case.

The OV2 interface for search allows to choose between the options LLT, LLT_class, PT, HLT, SMQ_N, and SMQ_B. During typing in the input field possible extensions of the given substring in the tables EVENT, PT, HLT, or SMQ, respectively, are shown. The user can mark the term she/he is interested in. The following tables are relevant for the presentation of possible terms and the evaluation:
LLT: REP_EVENT
LLT_class: REP_EVENT, PT and REP_PT
PT: PT and REP_PT
HLT: HLT and REP_HLT
SMQ_B: SMQ and REP_SMQ_B
SMQ_N: SMQ and REP_SMQ_N

With option LLT the basic table REP_EVENT representing the connections between reports and adverse events as given in the FAERS files REACxyQz is used for finding qualifying reports. As mentioned above, not every term in the *Pt* attribute of the table is a low level term in the MedDRA index (e.g. 'troponin c decreased'). In a database version with data up to 2022Q2 there are 2330 entries in the table REP_EVENT (which has more than 35 million entries) with adverse events not being such a low level term.
LLT_class allows to look for all reports with adverse events which belong to the same PT as the LLT in the input field. In this case it is checked whether the term(s) provided by the user are LLTs of the index. This variant may be helpful if a search at the PT level shall be done but the PT for a given LLT is not at hand.
With option PT all LLTs related to the given term in the index are taken as adverse events looked for in the query.
Option HLT can be chosen when all adverse events related to the specified high level MedDRA term shall be considered for evaluation.
With option SMQ_B (SMQ_N) an SMQ term has to be specified. OV2 looks for all adverse events which are related to the given SMQ term (with term_scope = 2) according to their connections in the index.

To consider the relationships in the index with every evaluation of the input values would be very time consuming. We therefore decided to build so-called materialized views, i.e. tables filled with the tuples obtained as result of a query evaluation. Up to the option 'LLT' they are used instead of the table REP_EVENT. This means that before using OV2 after the loading of new data, a script has to be run which fills the tables REP_PT, REP_HLT, REP_SMQ_B, and REP_SMQ_N with tuples representing the relationships between reports and MedDRA terms for each level of the index allowed in queries. In REP_PT, for example, ISRs of all reports are included which are related to an adverse event related to the given PT.

Under http://openvigil.sourceforge.net/doc/Fillreptables.sql you find appropriate SQL statements for the filling of the four REP-tables. Since the structure of the MedDRA index is not a tree but may be considered as a directed graph without (directed) loops, a special form of recursion has to be applied for the generation of REP_SMQ_B and REP_SMG_N. For REP_HLT and REP_PT simpler statements can be used.

The user interface

There are 5 tabs for the different functionalities of OV2:

- *Administration*: Allows to load data into the OV2 database from different sources (FDA, Drugbank, drugs@fda).

- *Search*: Offers flexible parametrization of queries as well as different options for the representation of results.

- *SQL*: Arbitrary SQL select statements can be formulated using tables from the PUBLIC subschema.

- *ShowReport*: Allows to inspect the raw data of single reports.

- *Browse*: Inverse browsing is possible on different attributes.

Up to *Search* it is straightforward how the different functionalities can be used.

The interface *Search* is a bit more involved and offers several possibilities to parameterize the query input as well as the form of the result. It allows to specify names of drugs or products as well as ATC codes as terms of Boolean expressions. The same flexibility is allowed for adverse events which can be chosen from different levels of the MedDRA index.

Furthermore, it is possible to add for each occurrence of a name of a drug or product, ATC code or for a substructure in the search interface a condition for the role it has to play in the drug usage part of a report in order to be considered as qualifying for evaluation. Possible roles are *primary suspect*, *secondary suspect*, *concomitant*, or *interacting*. If no condition shall be applied the default *Do not filter* has to be chosen. The specification of roles allows to formulate conditions like 'Lisinopril as primary suspect or Lisinopril as secondary suspect' or 'Dronedarone as primary suspect and Metoprolol not as concomitant'.

There are three forms for the providing of input data:
- Full specification, i.e. data on drugs etc. as well as on adverse events are given.
All reports or cases are qualifying for evaluation which fulfil the given conditions for the usage of drugs etc. and for the reported adverse events.

- No conditions for adverse events are specified:
All reports or cases fulfilling the given condition for the usage of drugs etc. are collected and grouped according to the adverse events related to these reports or cases. For each adverse event found so far the evaluation method chosen in the interface is executed.

- No condition for drugs etc. is specified.
All reports or cases fulfilling the given condition for the adverse events are collected and grouped according to the value found in the choice field of the Drug part in the interface (Drug, Pharmaproduct, ATC-Code, and Substructure). For each group the evaluation method chosen in the interface is executed.

Under 'Advanced search' additional filters can be specified which are applied to the search condition for reports. They are self-explanatory. What should be mentioned is that by specifying the earliest and the latest date of submission the set of reports to consider can be restricted to an arbitrary period of time.

For the counting of records contributing to the result of a search, three options are offered: Case (entire cases), ISR (most recent), and ISR (unique reports).
With the first option counting is done at the level of complete cases. If a single report of a case qualifies for the result, the case counts for the result. If more than one report qualifies this does not change the result.
The 'most recent' option restricts the set of reports taken for the computation of the result of a search to those reports which have the highest ISR value among all reports of the case they belong to. This proceeding allows to mimic the FDA proposal to take only the newest report of a case for evaluation (see remark above).
With the third option all reports may contributing to the counting.

Three evaluation methods can be chosen: Raw_data, Frequency, and Frequentist_methods.

- Raw_data: Report data are shown for all qualifying reports. Which of the report data are presented can be chosen under 'Output items'

- Frequency: A contingency table is computed in case of full specification of input data and presented in table form. If no adverse event is given but only a condition for the Drug option

and a level of the MedDRA index, a listing is produced with a line for every MedDRA term at the chosen level associated with a qualifying report together with the number of reports and cases in the set of qualifying reports and cases, respectively, having this MedDRA term as well among their given reactions.
If no condition for the Drug option is given up to the mandatory value in the choice field, a listing is produced with a line for every name of a drug or product which occurs in a qualifying report, together with the number of reports and the number of cases with such occurrences. In each line ‘drug’ or ‘product’ as value in the column ‘type’ tells whether its entry refers to an ingredient of a product or a product itself.

- Frequentist_methods: The same options as for ‘Frequency’ are available. The output listings, however, are extended by offering all values which are computed in connection with the generation of a contingency table (DE, dE,...,ROR, PRR,....).

If ‘Raw data’ is chosen for the evaluation method, output items are shown which can be chosen for the listings mentioned above.

Three output formats are offered: HTML, CSV, and Excel_CSV. Thus subsequent processing of results can easily be done.

Together with a result OV2 shows the SQL queries which have been generated and executed to generate the result. A user can take these queries, modify them arbitrarily by adding or removing conditions, for example, or by changing their target list. In the *SQL* tab, modified queries as well as arbitrary SQL select statements can be executed. This tab, however, and the tab *Administration* are not available for public use in our Web version for obvious reasons. It can only be used in a private installation of OV2.