European Commission

JRC TECHNICAL REPORTS

The Joint Research Centre Power Plant Database (JRC-PPDB)

A European Power Plant Database for energy modelling

Konstantinos KANELLOPOULOS
Ignacio HIDALGO GONZÁLEZ
Hrvoje MEDARAC
Andreas ZUCKER

2017

Joint Research Centre

EUR 28549 EN

Contact information
Name: Konstantinos Kanellopoulos
Address: European Commission, Joint Research Centre, P.O. Box 2, 1755 ZG Petten, Netherlands
Email: Konstantinos.kanellopoulos@ec.europa.eu
Tel.: +31224565053

JRC Science Hub
https://ec.europa.eu/jrc

JRC106289

EUR 28549 EN

PDF ISBN 978-92-79-67687-1 ISSN 1831-9424 doi:10.2760/329310

Luxembourg: Publications Office of the European Union, 2017

How to cite this report: K. Kanellopoulos, I. Hidalgo, H. Medarac, A. Zucker, *The Joint Research Centre Power Plant Database (JRC-PPDB) - A European Power Plant Database for energy modelling*, EUR 28549 EN, doi:10.2760/329310

Contents

Foreword

The abundance of computing power and storage at low cost has provided modelling groups with an opportunity to expand their activities to cover increasingly larger systems at higher detail. In order to maintain its relevance and capture local and regional level effects, this expansion must be supported by accurate input.

The internet is a very crowded place readily providing information on virtually anything – including energy infrastructure elements such as power plants. Such information is provided by institutions, for example the European Network of Transmission System Operators for Electricity (ENTSO-E) [1], open source data platforms, (such as Open Power System Data [2]), the academia (such as the "European power plant infrastructure database from Chalmers" [3] or the data for the North–West European Electricity Market model from University College Cork [4]) and databases developed and maintained by commercial firms (such as Platts or Argus [5]). The fact that each source formats and presents the information in a very different way may confine a modelling team to relying on one source, which in any case will never be completely comprehensive, accurate, and up to date.

The ability to readily access and cross-reference power plant information available from a multitude of sources is therefore of interest to groups of people engaged in energy modelling. The same holds for filling the gaps with reliable class-derived characteristics, where non-specific plant data are available.

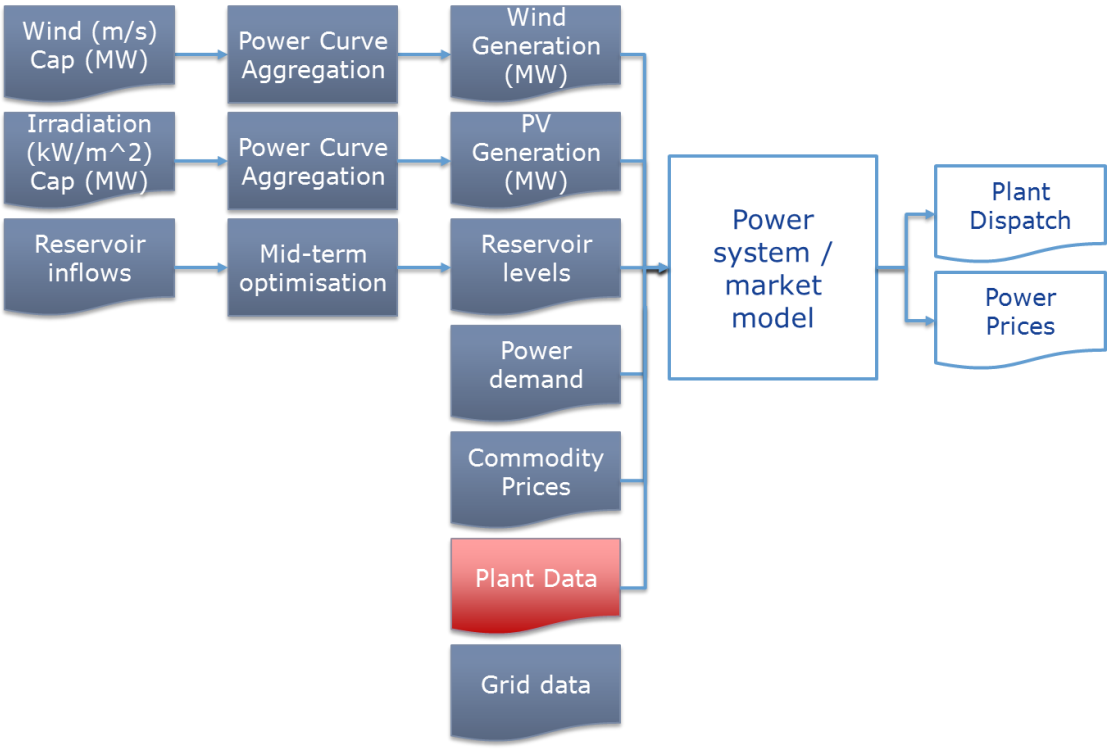
The current document reports on an attempt to cater for the above aiming to serve the modelling activities within the European Commission.

1 Introduction

The Joint Research Centre Power Plant Database for energy systems modelling (JRC-PPDB) is developed by Unit C.7 of the European Commission's Joint Research Centre in order to support the unit activities in energy systems modelling and knowledge management.

The amount of data required for energy modelling has increased significantly during the last years, both due to the geographic expansion of the modelled areas as well as the requirements to model the effects of continuously rising intermittent capacity. Figure 1 provides an indication of the range and complexity of data required to model the Power System. The database described in this report focuses only on power plant data.

Figure 1 – Data for energy modelling



Transparency is very high on the agenda, both as a means for enhancing self-regulation [6] as well as a way of enabling a more transparent political and societal discussion [7]. Several ongoing initiatives have embarked on an effort to create open data sets and models for energy modelling at a European scale. However it is likely that these efforts have still some way ahead before datasets achieve the required completeness and integrity levels.

ENTSO-E currently publishes a wealth of significant information on power generating units which are important for performing energy systems studies and power markets analysis. However the information currently available from ENTSO-E is neither sufficiently detailed, nor complete. Commercial databases on the other hand are reasonably accurate, while they provide in most cases very detailed information on power plants. However as this information is supplied by third parties it may be less accurate compared to ENTSO-E data. Cardinalities also differ when data comes from different sources (i.e. different Transmission System Operators - TSO hereafter). The above are significant barriers to directly using ENTSO-E published information in energy or power systems models.

The present work aims to address the above issues by creating a database which allows the user to access and cross reference information on power plants from multiple sources (currently ENTSO-E and two commercial databases).

Currently the JRC-PPDB contains the following fields for centrally dispatched power plants for 28 EU member states plus Norway, Switzerland and the Energy Community Western Balkan member states:

- Energy Identification Codes
- Plant and unit names
- Year of entry into service
- Location (country and coordinates)
- Power generation technology
- Fuel
- Installed capacity
- Cogeneration capability
- Efficiency
- Minimum load
- Ramp rates
- Start-up time
- Minimum up and down times
- No load costs
- Start-up costs
- Ramping costs

In addition to those details on specific power plants, the JRC-PPDB also contains information on renewable power generation capacity (wind, solar, etc.) clustered at country level.

The rest of the report is structured as follows:

- Chapter 2 provides a concise description of the three different datasets used.
- Chapter 3 describes the methodology used for automatically identifying the links between records from different databases.
- Chapter 4 describes the database structure and how the database tables were created by implementing outer joins on interim tables. The integration of class derived plant characteristics into the tables in order to create a power plant input file for the Dispa-Set [8] model is presented. The generation of a unique JRC_Code for each entry in the Database is described.
- Chapter 5 reports on the capacity checks, providing an indication of the completeness level of the JRC-PPDB.
- Finally in Chapter 6 a summary of our conclusions and ideas for further development are presented.
- A very large part of this work has been based on (Java and SQL) scripts that automatically generate and update tables. The most important SQL scripts are listed in 11 (annexes).

2 Data sets

The geographical scope of the current work extends to EU-28, Norway, Switzerland and six countries in Western Balkans, which are all members of the Energy Community. The datasets used are sourced from two commercial databases offered by Platts, ENTSO-E, and National/Regional TSOs.

2.1 Platts databases

The two power plant databases available commercially from Platts provide complete coverage of the geographical scope. There is some complementarity on the information offered, while prominent differences exist with regard to generator aggregation and the classification of technologies and fuels. Both databases contain a unique unit record id and an associated plant or location id.

2.1.1 The World Electric Power Plants database (WEPP)

Platts World Electric Power Plants database [9] is a global inventory of electric power generating units. It total it contains in excess of 63 thousand records for Europe. WEPP provides information at generator level.

2.1.2 PowerVision

Platts PowerVision [10] offers information on existing and planned power plants covering EU28 + CH, NO, IS, AL, BA, MK, ME, RS and KS. In total it contains in excess of 20 thousand records. PowerVision provides information at a higher level of aggregation, compared to WEPP, at autonomous unit level rather than at generator level.

2.2 Transmission System Operators

The TSO-derived data are sourced either from ENTSO-E's Transparency Platform and/or the respective National/Regional TSO.

2.2.1 ENTSO-E transparency Platform

One of the missions of the European Network of Transmission System Operators for Electricity (ENTSO-E) established by Regulation 714/2009 [11] is to ensure greater transparency regarding the entire electricity transmission network in the EU.

Previous transparency requirements were considered not enough to ensure a correct operation of the internal electricity market. Therefore access to more and better information was deemed necessary for all market participants, including "more precise information on electricity generation, supply and demand including forecasts, network and interconnection capacity, flows and maintenance, balancing and reserve capacity", in order to secure a genuine, well-functioning, open and efficient internal market in electricity. Equal access to information on the physical status and efficiency of the system is necessary to enable all market participants to assess the overall demand and supply situation and identify the reasons for movements in the wholesale price.

TSOs (transmission system operators, the members of ENTSO-E) were mandated by Regulation 714/2009 to provide information to the public, including "relevant data on aggregated forecast and actual demand, on availability and actual use of generation and load assets, on availability and use of the networks and interconnections, and on balancing power and reserve capacity. For availability and actual use of small generation and load units, aggregated estimate data may be used… The market participants concerned shall provide the transmission system operators with the relevant data".

In order to provide an overall view of relevant information across the Union, Regulation 714/2009 was amended by Regulation 543/2013 [12], which mandates that TSOs should facilitate the collection, verification and processing of data, and that ENTSO-E should make the data available to the public through a central information transparency platform. Regulation 543/2013 also lays down the minimum common set of data relating to generation, transportation and consumption of electricity to be made available to market participants and the public free of charge for at least five years.

The platform [1] was established by ENTSO-E at the beginning of 2015 and since then it publishes all data which TSOs are required to submit to the ENTSO-E in accordance to Regulation 543/2013.

2.2.2 National/Regional TSO data

National/Regional TSO data were used to complement Transparency Platform data, wherever available.

3 Database record matching

The first task of this work involved linking the ENTSO-E unit identifiers (EIC Codes) with the Platts database identifiers (Unit IDs). This task was carried out automatically, and then checked manually, in order to identify similar items.

3.1 The Energy Identification Coding (EIC) scheme

The Energy Identification Coding (EIC) scheme, developed by ENTSO-E provides a unique identification of the market participants and other entities active within the Energy Internal European Market (IEM).

According to this scheme a unique code is allocated to the following object types:

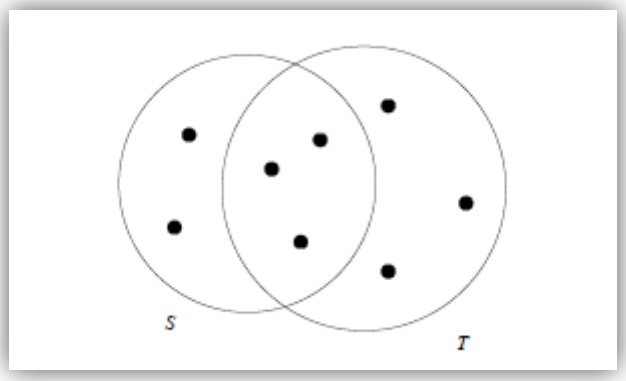
- Areas – object type "Y", Areas for inter System Operator data interchange
- Measuring Points – object type "Z", Energy Metering points
- Resource objects – object type "W", such as Production plants, consumption units, etc.
- Tie-lines – object type "T", International tie lines between areas
- Location – object type "V", Physical or logical place where a market participant or IT system is located
- Substations – object type "A".

Further information on the EIC scheme can be found at ENTSO-E's website [13]. The present work focuses on Resource objects (object types "W"), and in particular "Production Units" and "Generation Units".

3.2 Identifying and linking different records based on similarity

The metric used to define the similarity was the Jaccard similarity applied to a set of k-shingles. The Jaccard similarity of sets S and T is $|S \cap T|/|S \cup T|$, that is, the ratio of the size of the intersection of S and T to the size of their union. Figure 2 illustrates the notion of the Jaccard similarity.

Figure 2 - Two sets with Jaccard similarity 3/8 **[14]**



The notion of K-shingles is explained in [14]. A k-shingle is any k-length substring contained in the string we want to compare. Each record is associated with a set of k-shingles derived from the name string. The records are compared on the basis of the Jaccard similarity of the respective sets of k-shingles.

3.2.1 String pre-processing

The name conventions used for power plants by ENTSO-E vary widely between different areas/countries/TSOs. The same, though to a lesser extent, is true for the commercial databases. The following pre-processing was applied to strings before calculating the Jaccard similarity.

- Removal of recurring strings: Recurring Strings were identified for each country and a removal routine was introduced.
- Translation of Latin numbering to Arabic: Necessary to enable matching with the commercial databases were Arabic numbering is used.
- Replacement of all non-letter or number characters with space.

3.2.2 K-shingling

The sets of k-shingles are composed of all the k-consecutive characters present in the string. For example for k=3 the string Abono would give {"abo","bon","ono"}. Furthermore the creation of k-shingles was implemented for individual words in the strings (spaces not ignored).

By applying the above rules the k-shingles set (k=3) based on the string "Ag. Dimitrios I" would give the following set of k=3 shingles:

{"ag", "dim","imi","mit","itr","tri","rio","ios","1"}.

3.3 Filters

For those EIC_code records where either the ENTSO-E fuel based description "plant type" or the plant capacity were available, filters were devised in order to look for matches only among records in the same technology, fuel or size class.

3.3.1 Technology & fuel matching

Information on fuel and in some cases technology (hydro power plants) at generation level was available for a certain number of ENTSO-E records. In order to use this information (where available) interim tables were created to map the classification used by ENTSO-E to the fuel & technology codes used in the commercial plant databases. Table 1 presents the technology classification used in the JRC-PPDB, while Table 2 presents an example of a technology classification table used in PowerVision.

Table 1. Technology classification

AA	name	Dispa_code
1	st	STUR
2	cc	COMC
3	gt	GTUR
4	chp	OTHE
5	onshore	WTON
6	offshore	WTOF
7	run of river	HROR
8	reservoir	HDAM
9	pumped storage	HPHS
10	pv	PHOT
11	other	OTHE

Table 2. Example of a technology classification table for PowerVision

No	Unit_Prime_Mover	Plant_Type	Unit_PM_Tech_Detail_Desc	AA
1	ST	all	all	1
2	CC	all	all	2
3	GT	all	all	3
4	all	%Cogen	all	4
5	all	Wind	all	5
6	all	Offshore Wind	all	6
7	all	Hydro	all	7
8	all	Hydro	all	8
9	all	Pumped Storage	all	9
10	all	Solar	PHOTOVOLTAIC CELLS	10
11	OTHER	OTHER	OTHER	11

Use of the technology filter helped to reduce errors in some cases however it was found that inconsistencies present in the datasets (i.e. a gas fired plant in one dataset is reported as oil fired in another), as well as missing or partially correct information in ENTSO-E station lists reduced the initially anticipated benefit.

3.3.2 Capacity matching

Information on power plant capacity at generation level was available for a certain number of ENTSO-E records. For these cases a Gaussian function was used in order to consider similar capacities in the matching criteria.

The following capacity coefficient sc was introduced:

$$\mathrm{sc} = 1 + \alpha \cdot e^{-\beta \cdot (1 - p1/p2)^2}$$

Typical figures used for α ranged between 0.3 and 0.6 and for β between 1 and 10. The final matching score used was the product of the Jaccard similarity and the capacity coefficient *sc*.

3.3.3 Filter assessment

Although each record in the commercial databases contains several fields that could be used for matching purposes, it was found that only the plant name, and the country (in most cases) were of practical significance in the auto-matching process.

4 Database structure and creation

The JRC-PPDB database consists of 3 types of tables, the input tables, the intermediate or linking tables and the output tables. The following paragraphs provide an overview of the most significant tables currently used.

4.1 Input tables

The input tables contain the raw information provided by external sources, with limited processing required for the joining queries. The input tables in JRC-PPDB currently are provided in Table 3. Input tables.

Table 3. Input tables

Table name	Description
entsoe_station_capacities	ENTSO-E data at production facility level
wepp_full_2014	Table based on the 2014 update of the WEPP database
wepp_full	Table based on the 2016 update of the WEPP database
powervision_full	Table based on the 2016 update of the PowerVision database

The input tables are introduced (or may be updated) by using the PowerPlantDatabase java application described in 11.1.

4.2 Linking tables

The unit linking process described in the previous sections it was possible to associate power plant identification codes reported by ENTSO-E created the two interim linking tables called entsoe_[*db*]_links where *db* :{powervision, wepp}. These two tables contain the many to many relationships between eic_g & eic_p codes and *db* codes as presented in Table 4.

Table 4. Linking tables

Table name	Description
entsoe_[db]_links	Tables containing the links between [db]_full records and ENTSO-E eic_codes (production & generation).
tech_[*db*]_dispa	Tables associating [*db*] field categories to Dispa-SET technology codes [8].
fuel_[*db*]_dispa	Tables associating [*db*] field categories to Dispa-SET fuel codes [8].
country_code	Table associating country names, codes, areas and eic_code prefixes necessary to perform linkages between sources.
lookuptbl	Table associating plant classes with jrc_from_[db] records.

([1]) Modelling Future EU Power Systems Under High Shares of Renewables. The Dispa-SET 2.1 open-source model

The entsoe_[db]_links tables are populated (or may be updated) by using the PowerPlantDatabase java application described in 11.1 (option "p").

4.3 Output tables

The output tables provided in Table 5 are the result of outer join statements between input tables and linking tables. Presently the output tables contain the minimum fields necessary to identify power plants. These tables can be expanded to include fields present in any of the input tables.

Table 5. Output tables

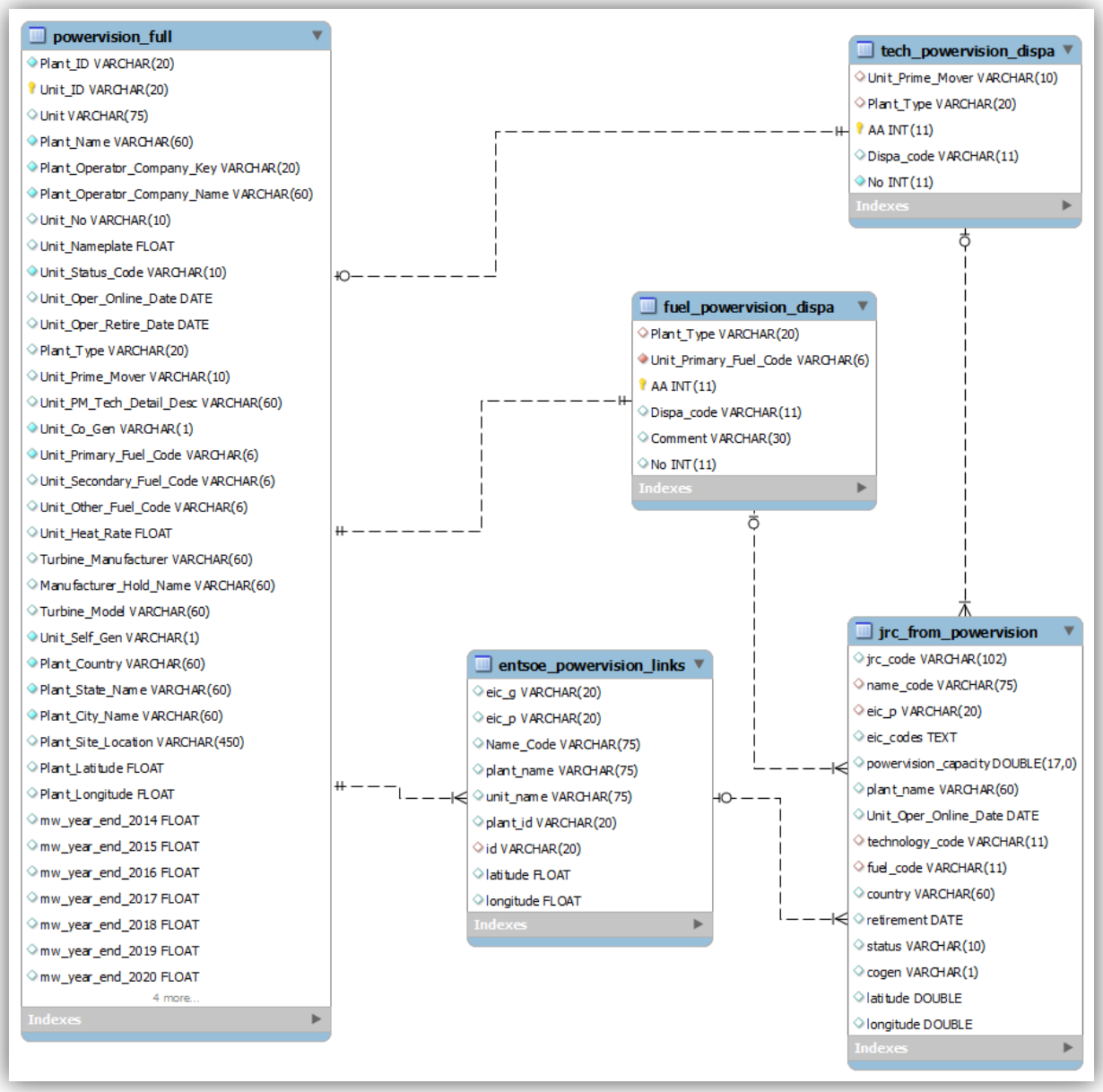
Table name	**Description**
jrc_from_[db]	Tables containing the links between [db]_full records and ENTSO-E eic_codes (production & generation).
capacity_aggregated_fuel	Table containing aggregate capacities per fuel and country from the jrc_from_[db] tables, the ENTSO-E Scenario Outlook & Adequacy Forecast SO&AF 2015 tables [15] and the Installed Capacity per Production Type([1]) published by ENTSO-E [1].
capacity_aggregated_fuel2	Table containing aggregate capacities per fuel and country from the jrc_from_[db] tables, and the Installed Capacity Per Production Unit([2]) published by ENTSO-E [1].
jrc_joined	Table linking jrc_from_[db] tables into one superset table.
dispaset_lookup	Table containing all the power plant information present in the input file for Dispa_SET.

([1]) Installed Generation Capacity Aggregated [14.1.A]
([2]) Installed Generation Capacity Aggregated [14.1.B]

The jrc_from_[db] tables are created by running the sql script "*Run me after loading entsoe_dbs linking tables.sql*" described in 11.2.

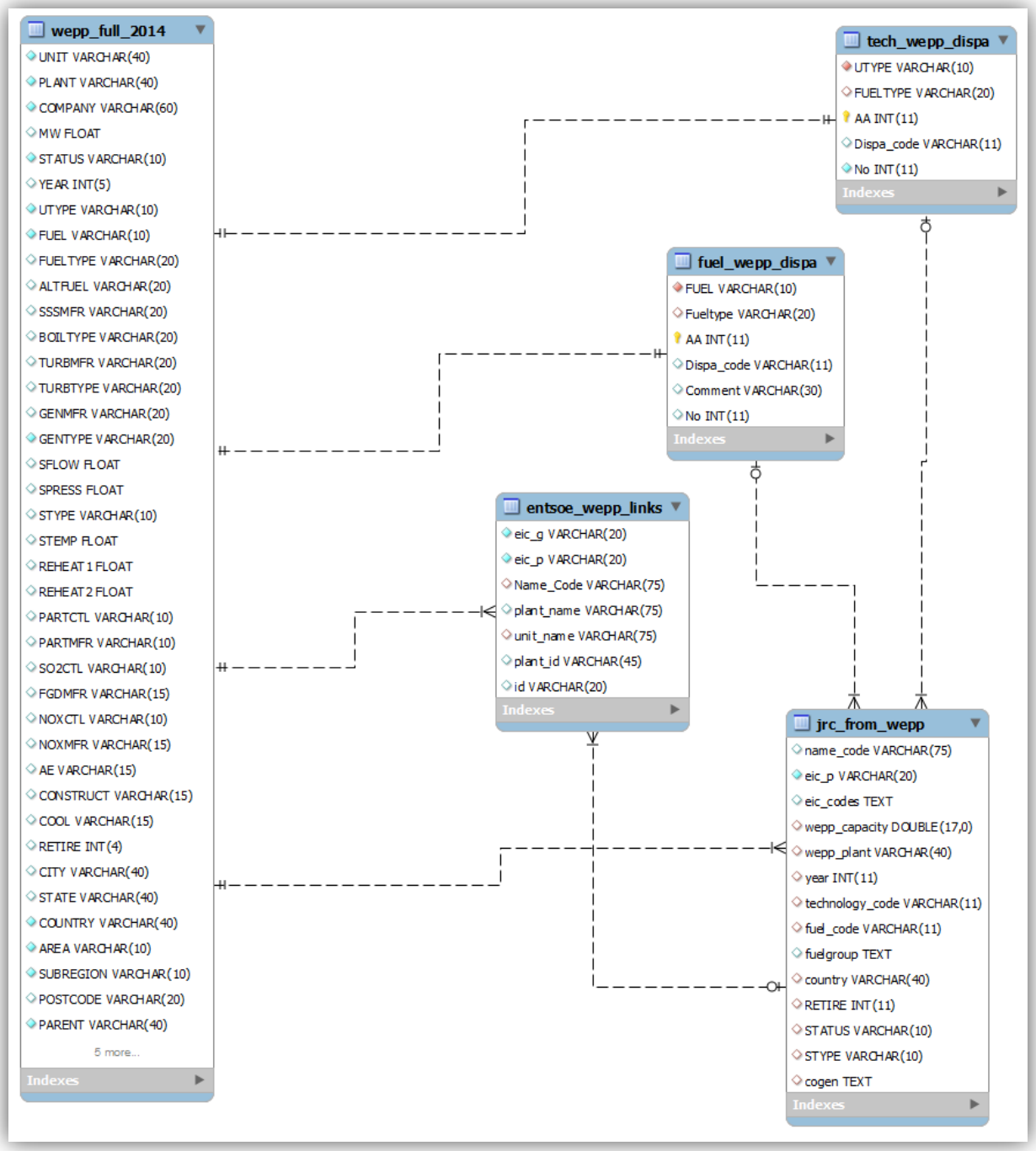
The entity-relationship diagram in Figure 3 provides a schematic indication of the links between the tables. The table jrc_from_powervision is created by joining with table entsoe_powervision_links and three other intermediate tables.

Figure 3 - Entity-relationship diagram for the Powervision-based table



The entity-relationship diagram in Figure 4 provides a schematic indication of the links between the tables. The table jrc_from_wepp is created by joining with table entsoe_wepp_links and three other intermediate tables.

Figure 4 - Entity-relationship diagram for the WEPP-based output table



4.4 Joining the tables

So far the tables *jrc_from_[db]* contain all power plant records based on database [db] (where *db* :{powervision, wepp}) which have been identified and linked with ENTSO-E records. This means that different plant datasets containing [db] and ENTSO-E fields can be easily created from each table. This is exemplified in chapter 5 where the records in *jrc_from_[db]* tables are compared to *entsoe_station_capacities* records in order to assess the completeness of each *jrc_from_[db].*

The three widely different power plant datasets: ENTSO-E, PowerVision and WEPP where through this process rendered similar. However the *jrc_from_[db]* tables are to a large extent complementary, as information presented in Table 6 indicates.

Table 6. Input tables

Dataset	**No of records**
jrc_from_wepp : W	2889
jrc_from_powervision : P	2858
entsoe_station_capacities: E	2885
$E \cap W \cap P$	2268
$W \cup P$	3197

Therefore the creation of a superset $W \cup P$ of the two tables was worth considering, especially if this was realised in a way that would allow the creation of a unique record identifier. In order to present significant added value the joined table should adhere to the following requirements:

1. The joined table should contain all records in W & P.
2. The joined table should include a unique identifier for each plant (primary key).
3. The fields of the joined table should be the union of the fields in the *jrc_from_[db]*.
4. The joined table should be generated and updated automatically.

4.4.1 Automated generation process

The difficulty in achieving this automatically was due to the fact that the only common field between the two tables was the *name_code* which is not unique. The joined table is generated automatically by sequential SQL scripts, all of which are included in ANNEX 2. The process is implemented in the following steps:

1. Primary keys are created for the *jrc_from_[db]* tables based on the field *name_code.* This is implemented by creating interim MyISAM tables *primary_[db].*
2. The table jrc_joined is created and populated with the intersection of the two datasets using an inner join between the *jrc_from_[db]* tables.
3. The table jrc_joined is created and populated with the remaining records from each of the *jrc_from_[db]* tables in two sequential insert statements.
4. A primary key is created for the generated table by using the interim MyISAM table *jrc_joined_pk.*

4.4.2 JRC code

The unique plant identifier (JRC Code) is created within the process described in the previous paragraph. It is composed of the following digits:

CC _FFF_TTTT_XXXX

Where:

CC : The country code.

TTTT : The four digit technology code adopted for use in Dispa_SET.

FFF : The three or four digit fuel code adopted for use in Dispa_SET.

XXXX : A four digit – zero filled – integer.

Example of a gas fired CCGT in Greece: *EL_GAS_COMC_0001*

4.5 Adding generic class data

The Mineral and Energy Economy Research Institute of the Polish Academy of Sciences, [16] has performed the work of classifying thermal power plants into generic classes [17]. The resulting table, which groups plants into classes based on fuel, age and technology, is a first attempt to assign generic technical characteristics to power plants where no specific information is available.

The classification of thermal power plants was made by energy carriers, reference technology, age, size, and turbine types (just for power plants based on gas turbines). Energy carriers considered were hard coal and lignite, natural gas and oil, uranium, biomass, biogas and water.

4.5.1 Plants using hard coal or lignite fuels

For plants using hard coal or lignite, reference technologies which were considered were subcritical, supercritical and ultra-supercritical steam turbine using hard coal or lignite, integrated gasification combined cycle and CHP technologies with back pressure and extraction steam turbines.

Age bins for plants with subcritical and supercritical steam turbines were below 10 years, between 10 and 30 years and above 30 years. For plants with ultra-supercritical steam turbines, the age bins were below 10 years and between 10 and 20 years. For other technologies using hard coal or lignite, no age bins was used.

Size bins which were used for plants using hard coal and lignite were the following: Large (400 MW and above), Medium (150 MW – 400 MW), Small (7.5 MW – 150 MW) and Extra small (below 7.5 MW).

4.5.2 Plants using natural gas or oil fuels

For plants using natural gas or oil, reference technologies which were considered were combined cycle plants with aeroderivative gas turbines, combined cycle plants with heavy-duty gas turbines, aeroderivative gas turbines in open cycle, heavy-duty gas turbines in open cycle, plants based on steam turbines, internal combustion plants, and CHP technologies based on combined cycle with gas turbines, gas turbines with waste heat boiler, plants based on internal combustion engines and technologies based on back pressure and extraction steam turbines.

Age bins within plants using natural gas or oil were made only in cases of heavy duty turbines and in relation to turbine types in the following way: E class until 2000, Fa class from 2000 to 2005, Fb class from 2006 to 2010 and H class after 2010.

Size bins which was used for plants using natural gas or oil fuels in any of turbine technologies were the following: Large (350 MW and above), Medium (100 MW – 350 MW), Small (9 MW – 100 MW) and Extra small (below 9 MW). For heavy duty gas turbines and steam turbines, all of these sizes were taken into consideration. Combined cycle plants with aeroderivative gas turbines were considered for Medium, Small and Extra small size, while gas turbine plants with aeroderivative gas turbines in open cycle were considered for Small and Extra small size. For plants based on internal combustion engines using natural gas or oil fuels, size bins were the following: Large (10 MW and above), Medium (5 MW – 10 MW), Small (2 MW – 5 MW) and Extra small (Below 2 MW). No size bins were used in cases of CHP technologies.

4.5.3 Nuclear power plants

Nuclear plants are currently classified in two categories in terms of technology: PWR (Pressurized Water Reactor) and BWR (Boiling Water Reactor). Each technology category is distinguished in two size bins: Large (above 750 MW) and Medium (below 750 MW). Age bins were not used. This classification is used only for assigning generic plant characteristics and in the future, will be updated to include gas cooled reactors (AGR) which are currently assigned with PWR generic class characteristics.

4.5.4 Plants using biomass fuels

Four categories of biomass fuelled power plants were considered: Plants using steam turbine to generate only electricity, CHP plants using back pressure or extraction steam turbines and CHP plants using ORC (Organic Rankine Cycle). No age nor size bins were used.

4.5.5 Plants using biogas fuels

Only two categories of biogas fuelled power plants were considered: Plants based on internal combustion engines aimed at generating only electricity and CHP plants based on internal combustion engines. No age nor size bins were used.

4.5.6 Water powered power plants

Water powered power plants were considered as conventional hydroelectric turbine (run-off-river and reservoir), pumped storage, tidal and wave. No age bins were used. For run-off-river hydroelectric plants size bins were the following: Large (30 MW and above), Medium (10 MW – 30 MW), Small (1 MW – 10 MW) and Extra small (Below 1 MW). For reservoir hydroelectric plants size bins were the following: Large (75 MW and above), Medium (10 MW – 75 MW), Small (1 MW – 10 MW) and Extra small (Below 1 MW). No size bins were used in cases of tidal and wave technologies.

4.5.7 Integration of the generic class table into the JRC-PPDB

The proposed classification table containing the "generic plant characteristics" is integrated into the table *lookuptbl*, which is used for assigning generic class categories to each record the jrc_from_[db] tables, as shown in Table 7.

Table 7. A fraction of the records in lookuptbl

id	fuel	tech_code	cogen	Stype	year_from	year_to	size_from	size_to
10	HRD	STUR	N	SUBCR	1900	1976	150	400
11	HRD	STUR	N	SUBCR	1900	1976	8	150
12	HRD	STUR	N	SUBCR	1900	1976	0	8
13	HRD	STUR	N	SUPERC	2006	2016	400	9999
14	HRD	STUR	N	SUPERC	2006	2016	150	400
15	HRD	STUR	N	SUPERC	2006	2016	8	150
16	HRD	STUR	N	SUPERC	2006	2016	0	8
17	HRD	STUR	N	SUPERC	1976	2006	400	9999
18	HRD	STUR	N	SUPERC	1976	2006	150	400
19	HRD	STUR	N	SUPERC	1976	2006	8	150
20	HRD	STUR	N	SUPERC	1976	2006	0	8
21	HRD	STUR	N	SUPERC	1900	1976	400	9999
22	HRD	STUR	N	SUPERC	1900	1976	150	400
23	HRD	STUR	N	SUPERC	1900	1976	8	150
24	HRD	STUR	N	SUPERC	1900	1976	0	8
25	HRD	STUR	N	ULTRSC	2006	2016	400	9999
26	HRD	STUR	N	ULTRSC	2006	2016	150	400
27	HRD	STUR	N	ULTRSC	2006	2016	8	150

The fields (fuel, tech_code, Stype, year_from, year_to size_from, size_to) are the primary key of this lookup table and are used to assign the id class to each record in jrc_from_wepp and therefore any characteristic associated with the class. An example is shown in Table 8 with the class efficiency.

Table 8. A fraction of the records in table linked_lookup

name_code	wepp_plant	fuel_code	technology_code	wepp_capacity	cogen	stype	year	id	best_efficiency_of_power_generation
AG_DIMITRIOS1	AGIOS DIMITRIOS	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
AG_DIMITRIOS2	AGIOS DIMITRIOS	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
AG_DIMITRIOS3	AGIOS DIMITRIOS	LIG	STUR	310	N	SUBCR	GREECE	38	36.456
AG_DIMITRIOS4	AGIOS DIMITRIOS	LIG	STUR	310	N	SUBCR	GREECE	38	36.456
AMYNDEO1	AMYNTAIO	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
AMYNDEO2	AMYNTAIO	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
KARDIA3	KARDIA	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
KARDIA4	KARDIA	LIG	STUR	300	N	SUBCR	GREECE	38	36.456
MEGALOPOLI4	MEGALOPOLIS-A	LIG	STUR	310	N	SUBCR	GREECE	38	36.456
KARDIA1	KARDIA	LIG	STUR	300	N	SUBCR	GREECE	42	35.281
KARDIA2	KARDIA	LIG	STUR	300	N	SUBCR	GREECE	42	35.281
MEGALOPOLI3	MEGALOPOLIS-A	LIG	STUR	300	N	SUBCR	GREECE	42	35.281
PTOLEMAIDA4	PTOLEMAIS	LIG	STUR	300	N	SUBCR	GREECE	42	35.281
MEGALOPOLIS-A NO 1	MEGALOPOLIS-A	LIG	STUR	125	N	SUBCR	GREECE	43	29.919
MEGALOPOLIS-A NO 2	MEGALOPOLIS-A	LIG	STUR	125	N	SUBCR	GREECE	43	29.919
PTOLEMAIS 1	PTOLEMAIS	LIG	STUR	70	N	SUBCR	GREECE	43	29.919
PTOLEMAIDA2	PTOLEMAIS	LIG	STUR	125	N	SUBCR	GREECE	43	29.919
PTOLEMAIDA3	PTOLEMAIS	LIG	STUR	125	N	SUBCR	GREECE	43	29.919
AG_DIMITRIOS5	AGIOS DIMITRIOS	LIG	STUR	366	Y	SUPERC	GREECE	74	42
MELITI	FLORINA	LIG	STUR	330	Y	SUPERC	GREECE	74	42
LAVRIO4	LAVRIO	GAS	COMC	570	N	N/A	GREECE	81	55.34
KOMOTINI	KOMOTINI	GAS	COMC	500	N	N/A	GREECE	85	59.35
LAVRIO5	LAVRIO	GAS	COMC	378	N	SUBCR	GREECE	85	59.35
ELPEDISON_THESS	THESSALONIKI POWER	GAS	COMC	400	N	SUBCR	GREECE	85	59.35

4.6 Accessing the JRC-PPDB

The JRC-PPDB is currently hosted on a server at the Directorate C of the JRC in Petten. It can be accessed either through a simple web interface, or by connecting to the server via a MySQL administration tool.

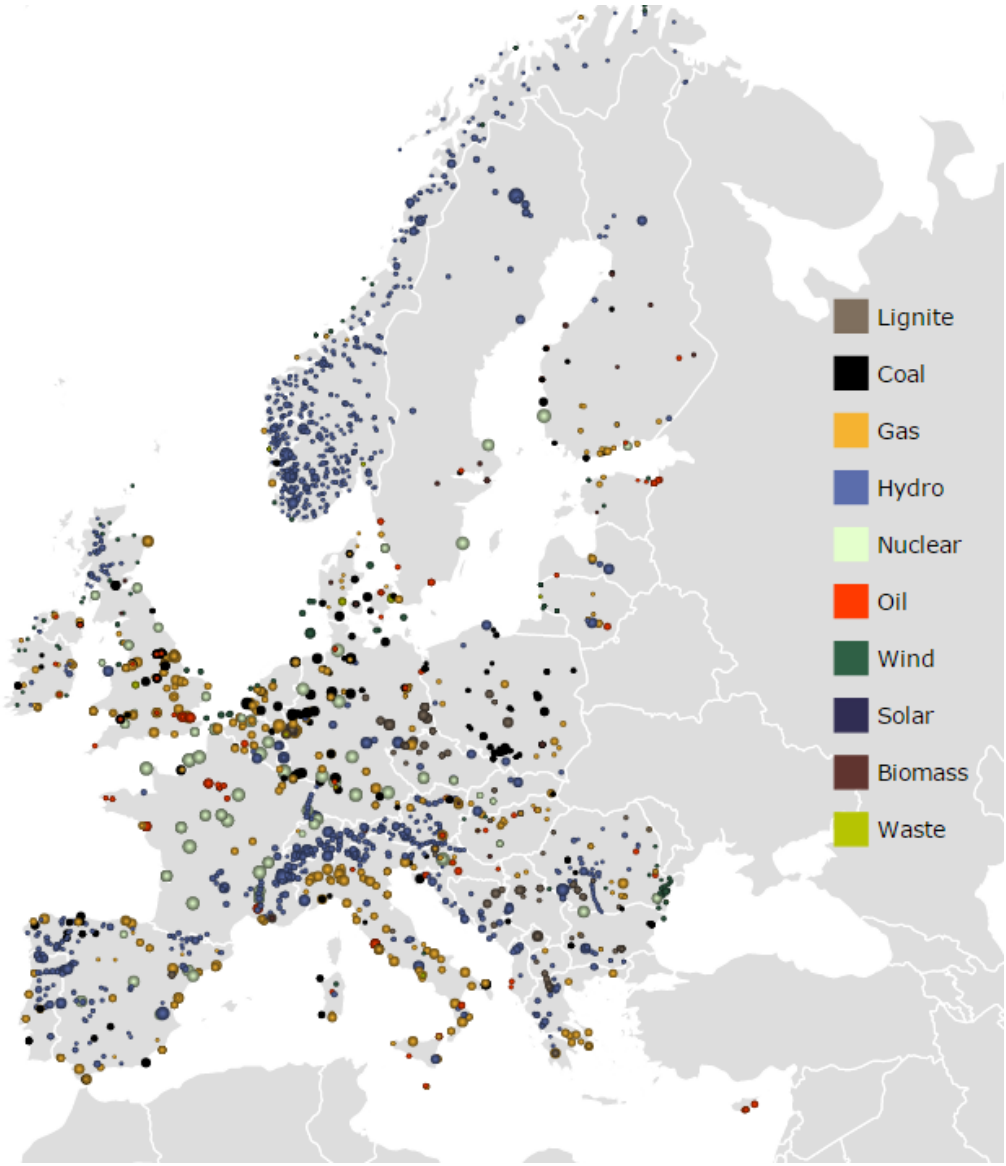
4.6.1 Accessing through the web interface

A simple web interface is available to all users at the JRC Petten site through the following link:

http://ppdb-dev.jrc.nl/ppdb.php

The site allows the user to access the jrc_from_powervision table. By selecting the combo boxes it is possible to query data from the table per country or per fuel. Multiple country and fuel type selections are also possible. Figure 5 provides the resulting map containing the result of the query.

Figure 5 – JRC-PPDB on a map



On the lower left corner of the output screen a button enables the user to download the query results.

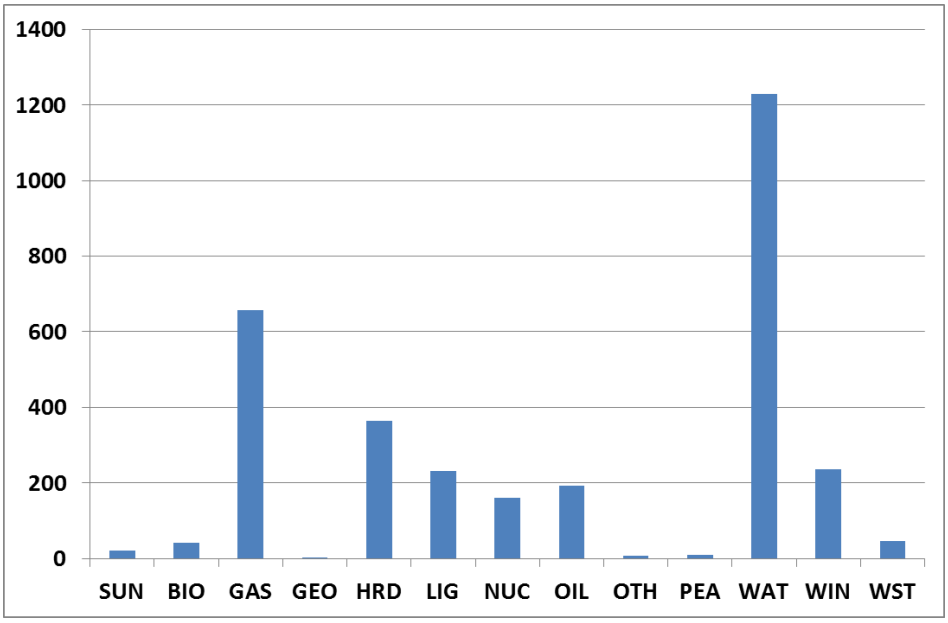
4.6.2 Accessing through a MySQL administration tool

For accessing data in more tables or joining tables a connection to the server "ppdb-mysql.jrc.nl" is possible through MySQL management tools such as *phpMyAdmin* or *MySQL Workbench* by using a public user profile which currently has read access to the JRC-PPDB.

5 A validation of the JRC-PPDB completeness

In this chapter the database completeness against ENTSO-E reported station capacity [1] is presented. In particular the work involved collecting and adapting the ENTSO-E capacity at Production Unit level in order to assess the level of completeness of the power plants currently present in the JRC-PPDB output tables. The current number of records in the JRC-PPDB, aggregated per fuel type, is provided in Figure 6.

Figure 6 - Number of records in JRC-PPDB per fuel type



The results shown in the following paragraphs provide a comparison of the aggregated per energy source installed capacity. The data are queried from three different sources tables (*jrc_from_powervision, jrc_from_wepp and entsoe_station_capacities*).

For each of the *jrc_from[db]* two numbers are reported:

- *jrc_[db]_capacity_total*: The total capacity of plants in the *jrc_from[db]* table.
- *jrc_[db]_capacity_2016:* The total capacity of plants in the *jrc_from[db]* table which are reported as active during 2016.

Similarly two numbers are presented from the *entsoe_station_capacities* table:

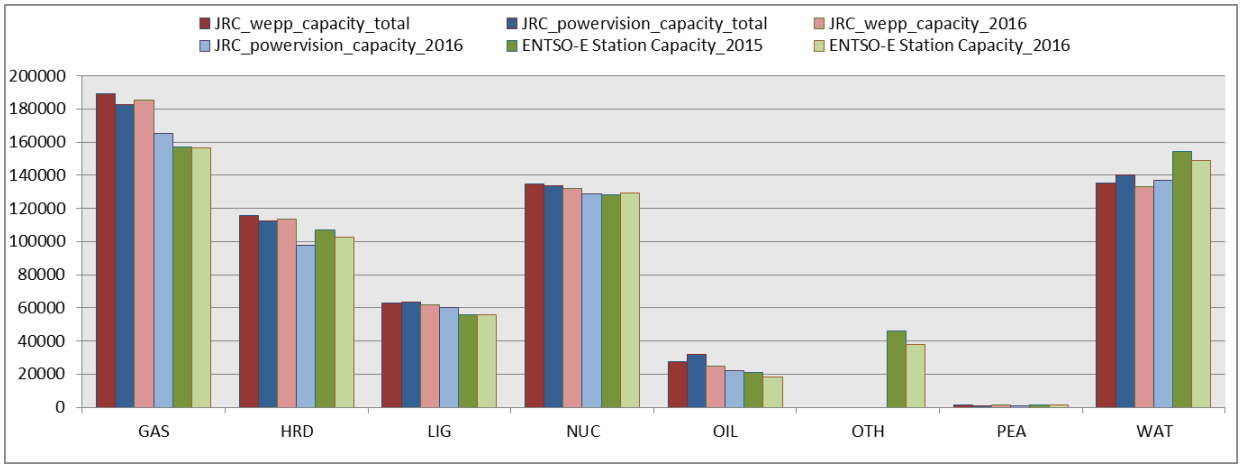
- *ENTSO-E Station Capacity_2015: The total capacity of stations reported as active during 2015 the capacity_aggregated_fuel2 table. (Sums per country and fuel)*
- *ENTSO-E Station Capacity_2016: The total capacity of stations reported as active during 2016 the capacity_aggregated_fuel2 table. (Sums per country and fuel).*

5.1 Overview

Figure 7 provides the total aggregated capacity per fuel type in the area covered by the JRC-PPDB. The series with the darker shades (red or blue) contain the total capacity, regardless of commissioning or de-commissioning information. The series with the lighter shades (red or blue) contain the total capacity for 2016, according to the Platts databases. The type "other" is predominantly composed of oil, gas and coal fired power plants reported by the Italian TSO as such. This the primary cause of the observed discrepancy in the aggregate sums for gas, coal and to a lesser extent oil.

Small hydro (sub-100 MW) plants not reported by ENTSO-E or the TSOs are the reason behind the missing hydro capacity for some counties (France, Italy, Sweden and UK). This capacity will be represented as cluster of units.

Figure 7 - Capacity (MW) aggregates per fuel in the JRC-PPDB



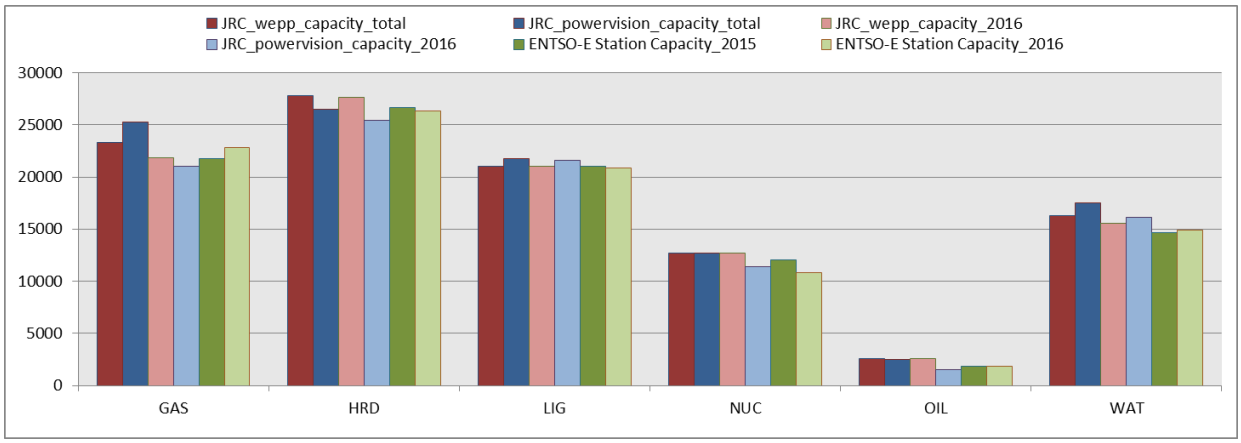
In the following paragraphs the aggregate capacities per fuel are reported for each bidding zone (BZN) or for each control area (CTA) for those countries with multiple bidding zones within one country (Italy, Denmark Norway and Sweden). Countries which do not report *Installed Capacity per Production Unit* figures on the ENTSO-E Transparency Platform are not included. These countries are Albania, Bosnia and Herzegovina, Croatia, Cyprus, Malta and Kosovo.

The conventions used in the charts in the following paragraphs are the same as those used in Figure 7 and presented above.

5.2 Austria-Germany-Luxembourg

This is a very large area where the database allows us to identify some inconsistencies between databases and ENTSO-E information, particularly regarding fuel type. Completeness level is very high for large power plants.

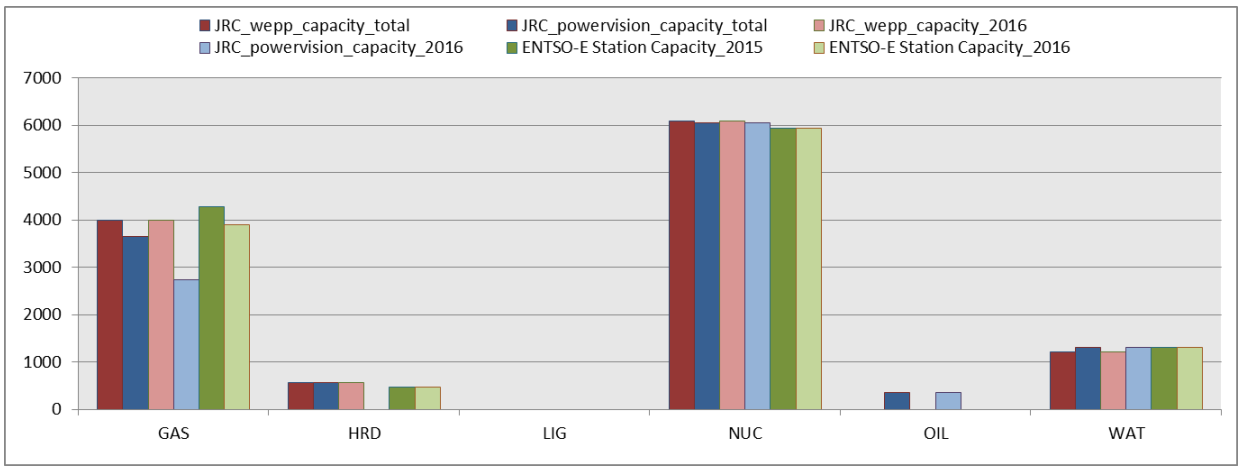
Figure 8 - Capacity (MW) aggregates per fuel for the area AT-DE-LU



5.3 Belgium

Belgium is 100% complete on all major power plants as illustrated in Figure 9. The gap observed in 2016 capacity in Powervision-based information is caused by the retirement of DROGENBOS GT2 HERDERSBRUG GT2 and Langerloo 1&2 at the end of 2015.

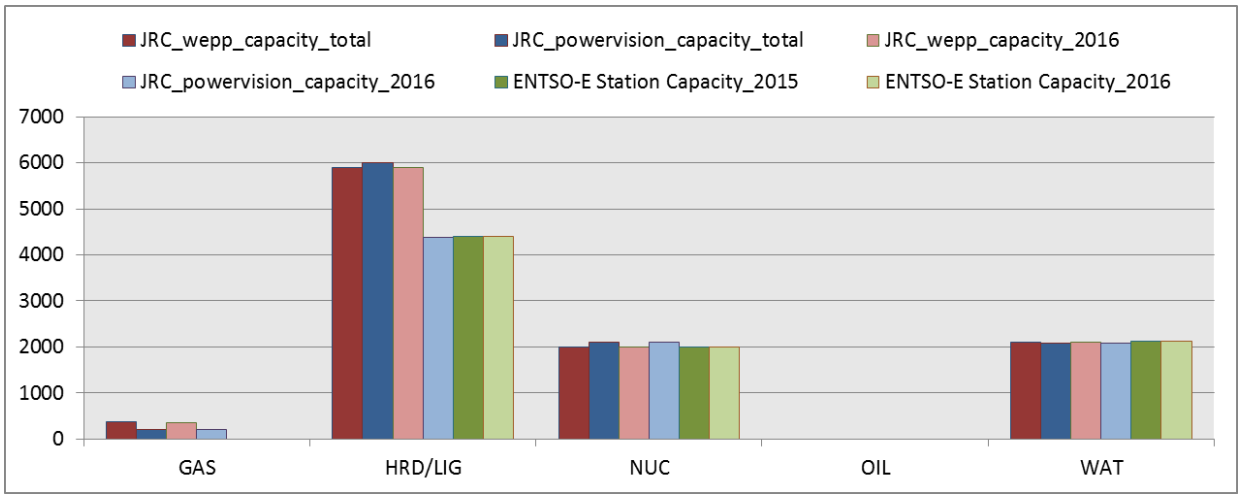
Figure 9 - Capacity (MW) aggregates per fuel in Belgium



5.4 Bulgaria

Bulgaria is 100% complete on all major power plants. Aggregate data on sub-bituminous coal and lignite fired units are presented in Figure 10 as one category due to a discrepancy in the fuel naming conventions (Sub-bituminous coal vs lignite).

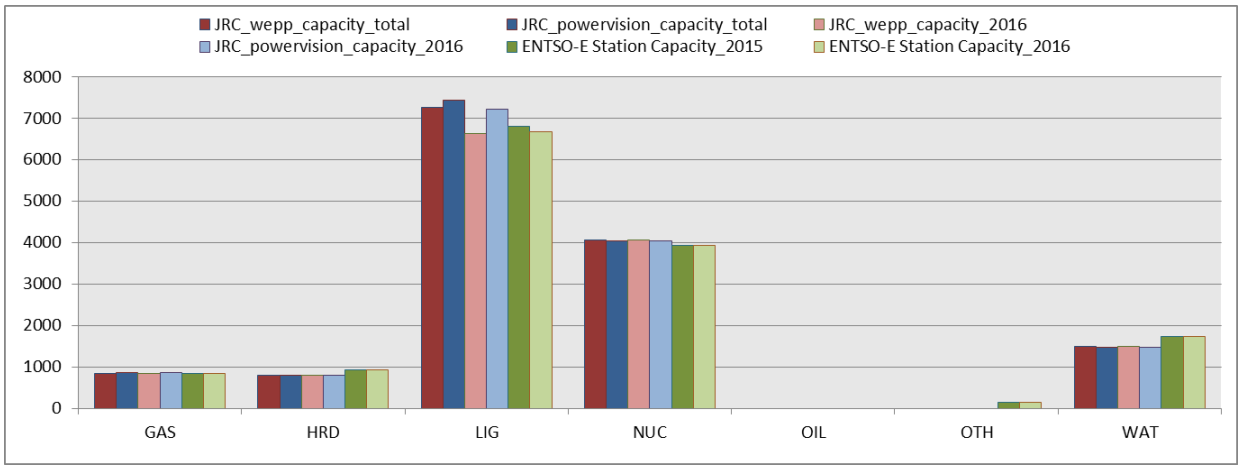
Figure 10 - Capacity (MW) aggregates per fuel in Bulgaria



5.5 Czech Republic

The Czech Republic is 100% complete on all major power plants as illustrated in Figure 11.

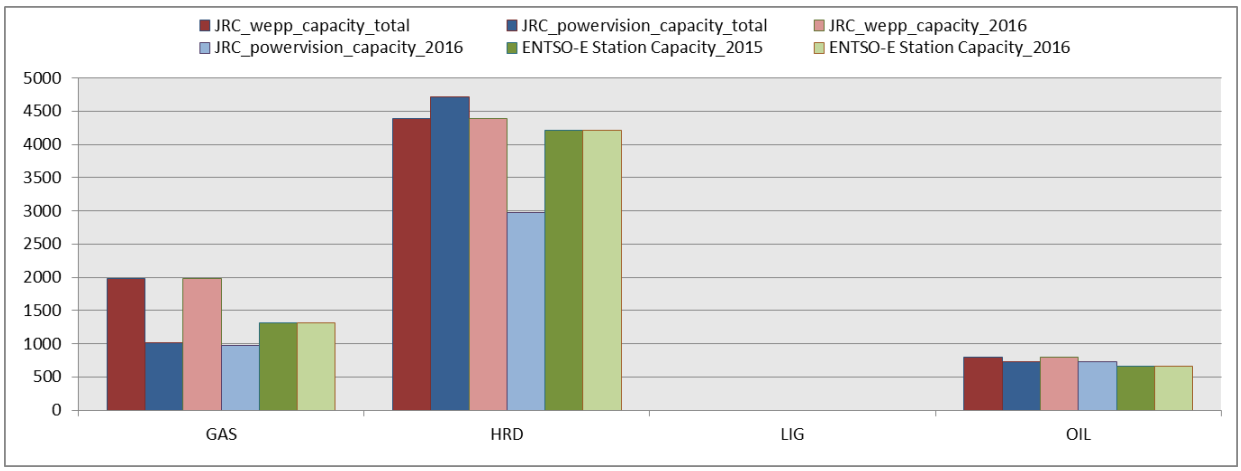
Figure 11 - Capacity (MW) aggregates per fuel in the Czech republic



5.6 Denmark

Denmark is 100% complete on all major power plants. The discrepancy observed (Figure 12) on coal and lignite fired plants is due to the reporting of Stigsnæsværket Blok 1&2 and Enstedsværket as units in Reserve.

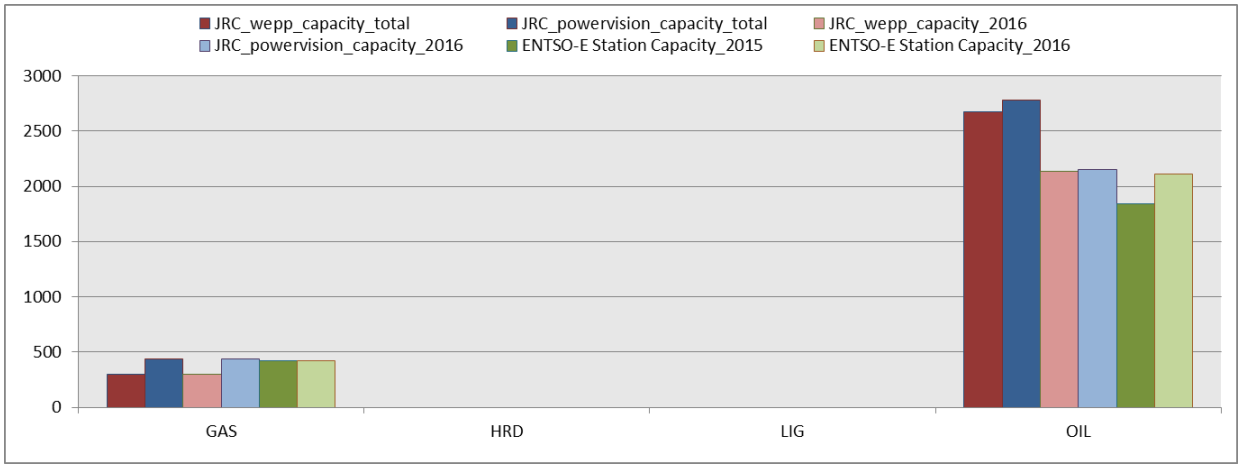
Figure 12 - Capacity (MW) aggregates per fuel in Denmark



5.7 Estonia

Estonia is 100% complete on all major power plants as illustrated in Figure 13.

Figure 13 - Capacity (MW) aggregates per fuel in Estonia



5.8 Finland

Finland is 100% complete on all major power plants. However as Figure 14 illustrates, the data on station availability for 2016 require further research in order to verify the status of plants that according to one database are retired, but are included in the ENTSO-E lists. These are:

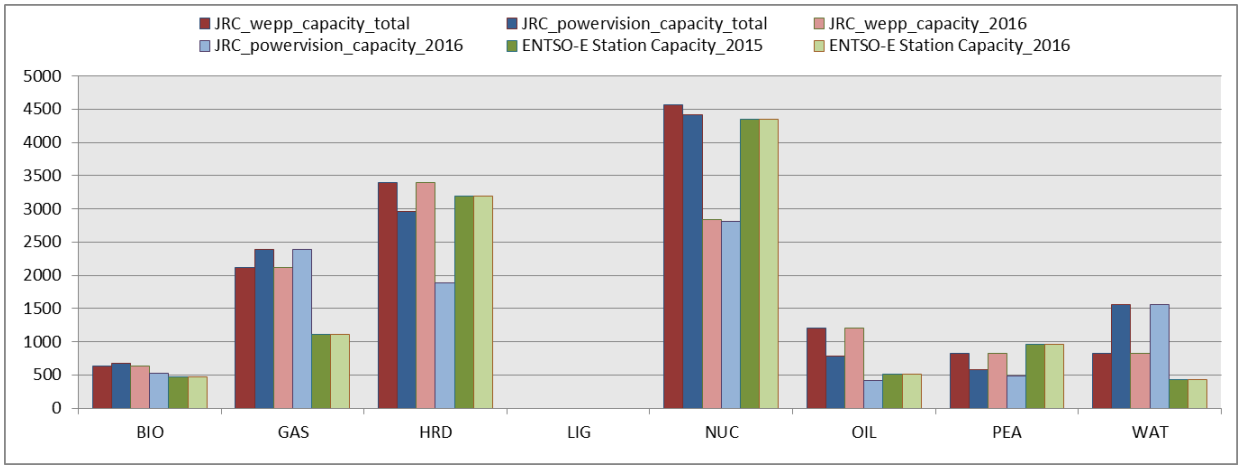
- Kristiina 2 and Inkoo 1-3 coal fired plants with a total capacity of 1150 MW

The nuclear plant of Okiluoto 3 is reported in both databases as under construction with a commissioning date in 2018.

There are also inconsistencies regarding the fuel type:

- Kristiina reported as Fossil hard coal is composed of coal and oil fired units.
- Forssa is reported as Fossil oil while the commercial sources identify it as gas fired.

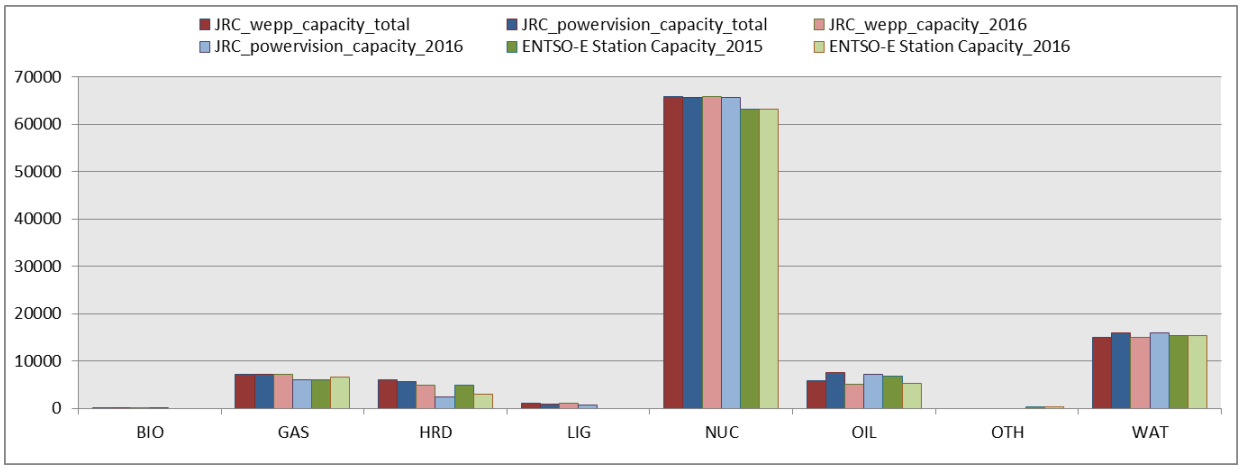
Figure 14 - Capacity (MW) aggregates per fuel in Finland



5.9 France

France is 100% complete on all major power plants. Both databases appear to report gross capacities for nuclear plants (typically 4-5% higher than ENTSO-E figures).

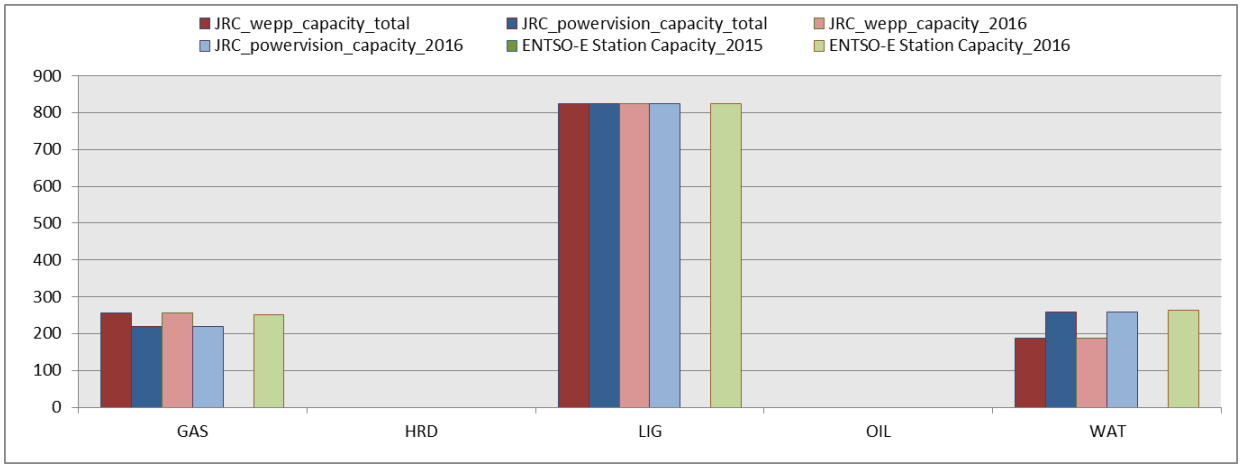
Figure 15 - Capacity (MW) aggregates per fuel in France



5.10 The former Yugoslav Republic of Macedonia

The plant list is complete for the former Yugoslav Republic of Macedonia.

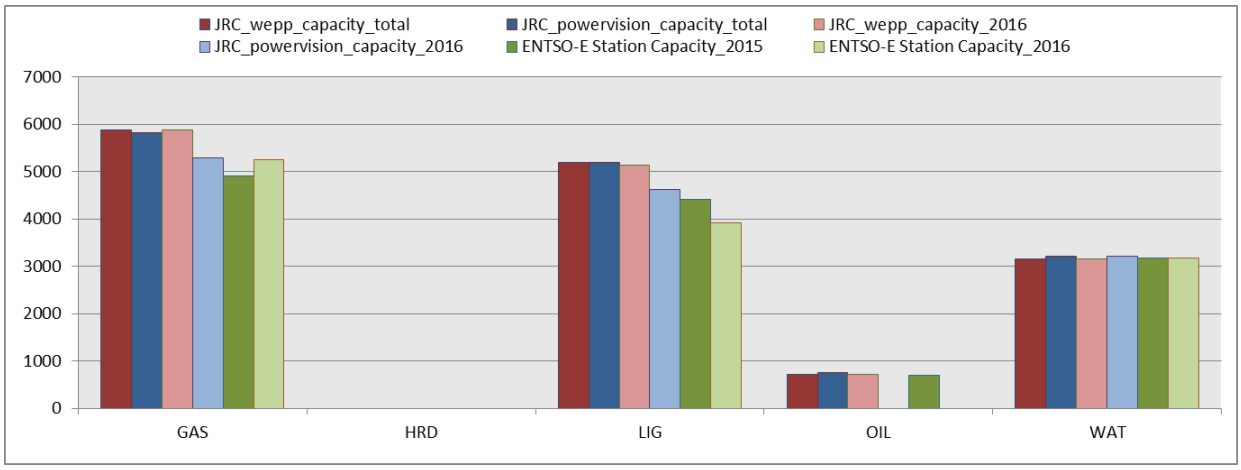
Figure 16 - Capacity (MW) aggregates per fuel in The former Yugoslav Republic of Macedonia



5.11 Greece

Greece is 100% complete on all major power plants. Both databases report gross capacities for lignite-fired plants (typically 10% higher than ENTSO-E data).

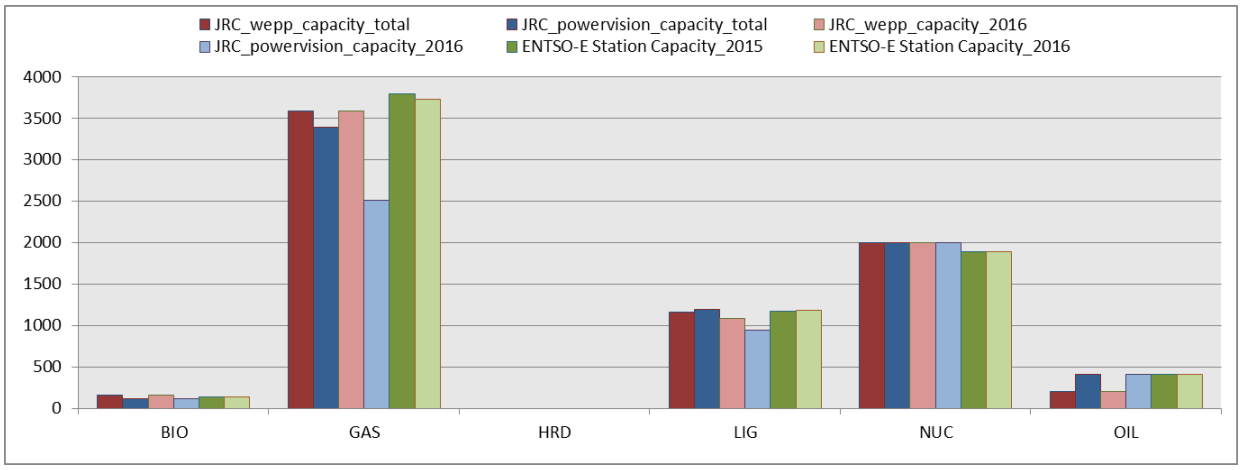
Figure 17 - Capacity (MW) aggregates per fuel in Greece



5.12 Hungary

All major power plants in Hungary are identified and linked. The discrepancy on 2016 observed for gas capacity in 2016 is due to the units Tisza II (880 MW) which are reported in one database as "out of operation". Similarly Oroszlany (220 MW) is mentioned as due for retirement in 2016.

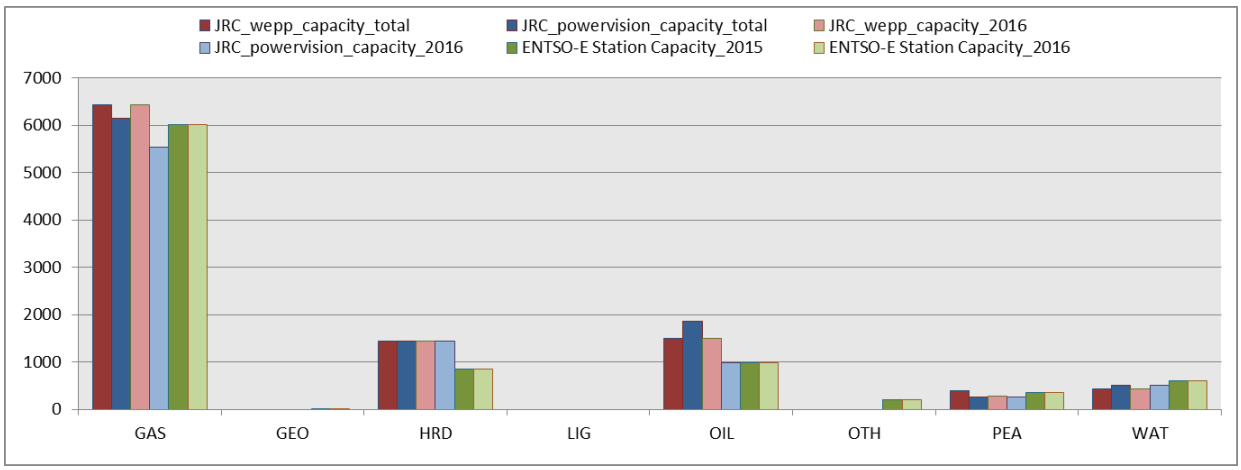
Figure 18 - Capacity (MW) aggregates per fuel in Hungary



5.13 Ireland

The JRC-PPDB is 100% complete on all major power plants in Ireland. The discrepancy observed is due to ENTSO-E's probably mistaken reporting of Kilroot as a gas fired plant. Both databases report it as "Coal".

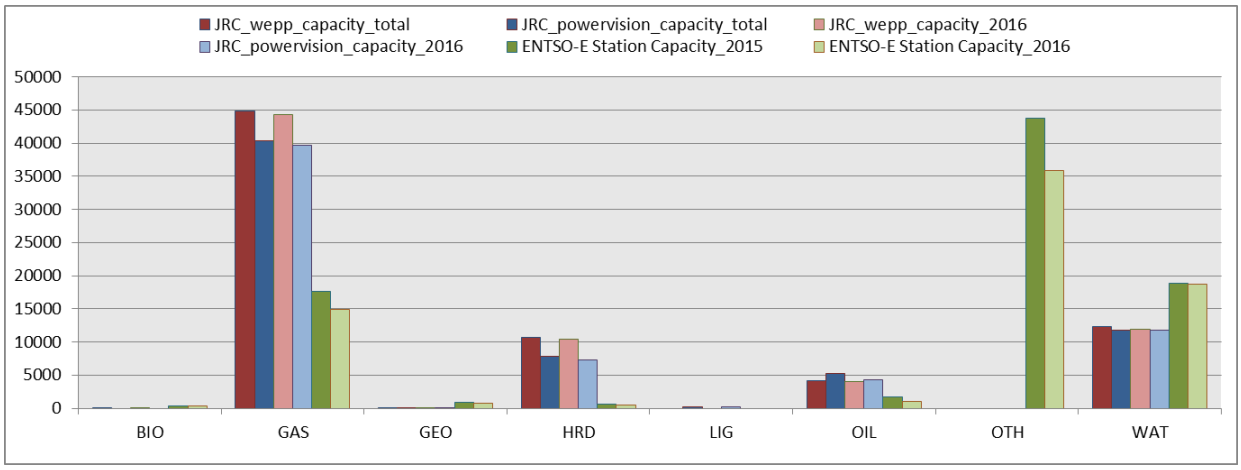
Figure 19 - Capacity (MW) aggregates per fuel in Ireland



5.14 Italy

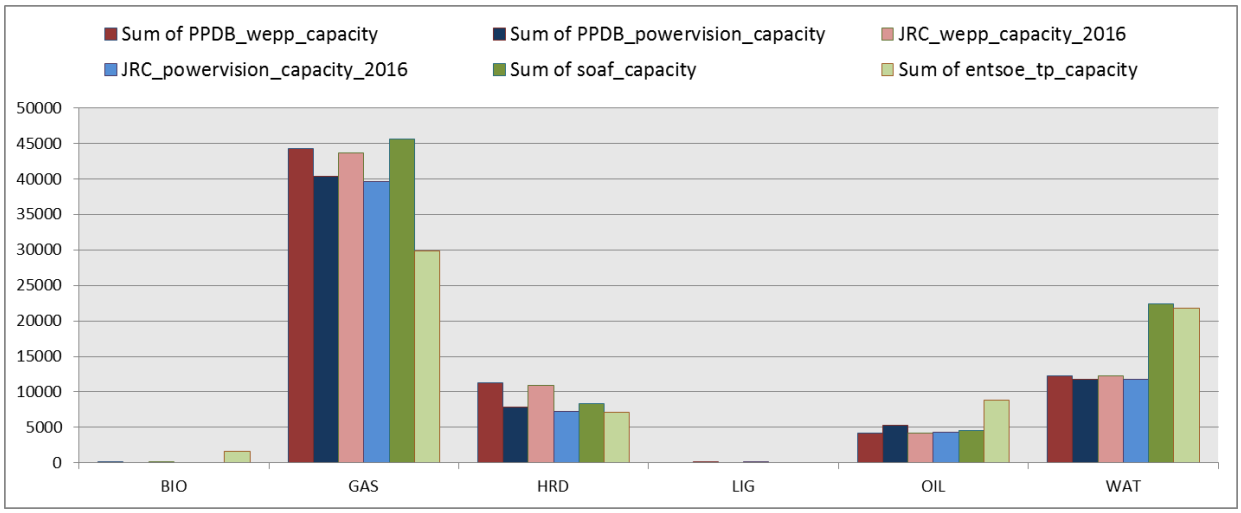
Completeness level is very high for large power plants in Italy. The Italian plants cannot be verified from the Installed Capacity per Production Unit data since approximately 40 GW of plants are reported as "Other" as illustrated in Figure 20.

Figure 20 - Capacity (MW) aggregates per fuel in Italy



Due to this fact, Figure 21 is additionally provided, which compares the aggregate capacities in the jrc_from_[db] tables with two **alternate** ENTSO-E sources (the SO&AF 2015 and the Installed Capacity per Production Type 2016).

Figure 21 - Capacity (MW) aggregates per fuel in Italy compared to ENTSO-E aggregates([1])



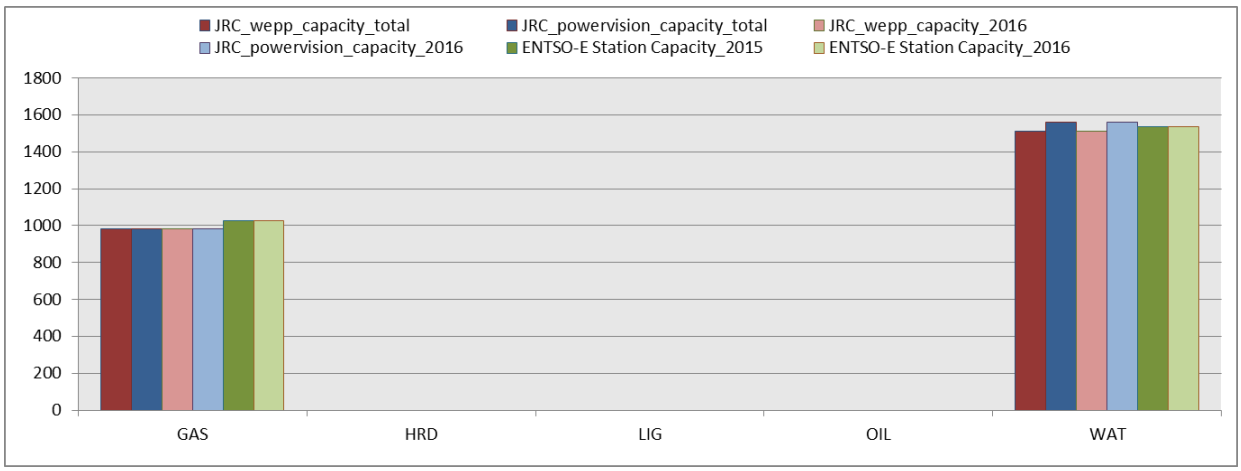
([1]) ENTSO-E SOAF 2015 tables and Installed Generation Capacity Aggregated [14.1.A] as reported on the transparency platform

The discrepancy observed on gas and oil in 2016 ENTSO-E data is probably due to mothballing of gas-fired plants or due to provisional participation to the market with backup fuel oil. Gas for 2017 is reported by ENTSO-E back to the level of 40 GW and oil is reported down to the level of 5GW.

5.15 Latvia

The plant lists for Latvia are complete and consistent, as indicated in Figure 22.

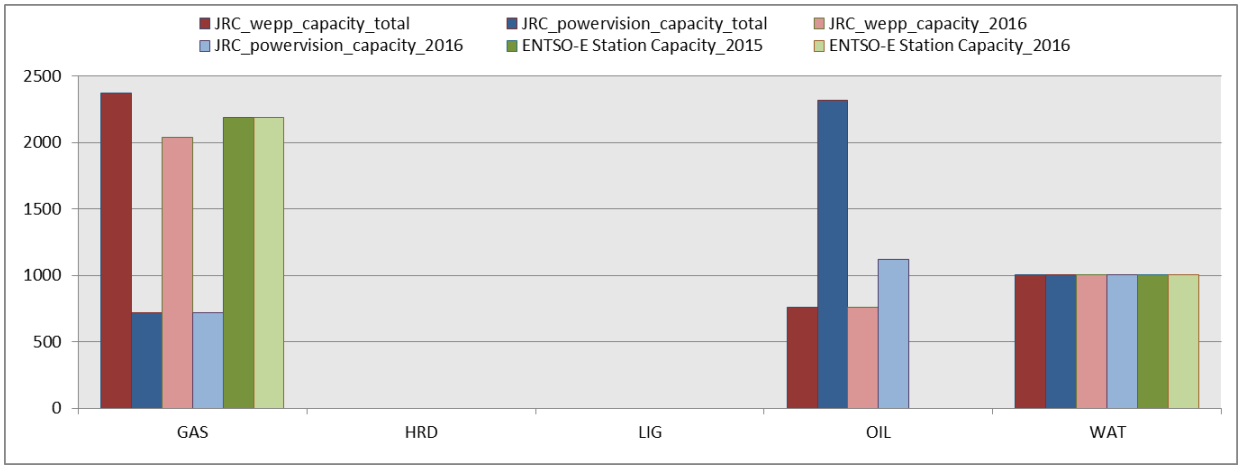
Figure 22 - Capacity (MW) aggregates per fuel in Latvia



5.16 Lithuania

The plant lists for Lithuania are complete. There is an inconsistency caused by the fact that ENTSO-E reports on Lietuvos as gas-fired, while the two databases report oil-fired capacity (600 MW) within the plant. Some not reported decommissioning may have taken place in this plant.

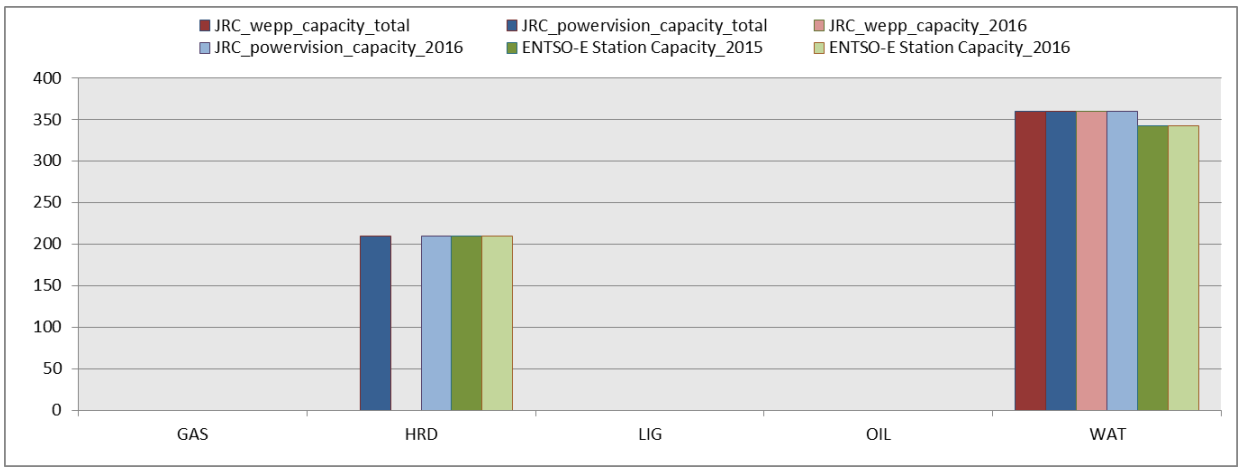
Figure 23 - Capacity (MW) aggregates per fuel in Lithuania



5.17 Montenegro

Plant lists for Montenegro is complete as indicated in Figure 24.

Figure 24 - Capacity (MW) aggregates per fuel in Montenegro

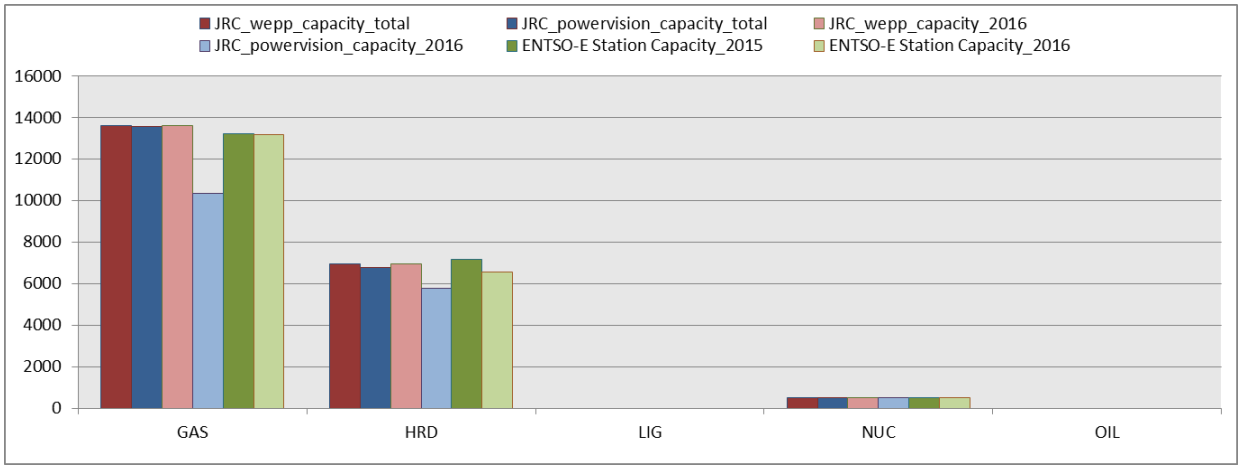


5.18 The Netherlands

The JRC-PPDB is 100% complete on all major power plants in the Netherlands as indicated in Figure 25. The discrepancy observed is explained by the following:

- Hemweg plant (1090 MW) is reported by ENTSO-E as coal fired. However it includes a 500 MW gas fired CCGT.
- One database reports Borsele12 (coal - 403 MW) as retired in 2016 and Eemshaven20-30, Moerdijk 2 and Claus as "out of operation".

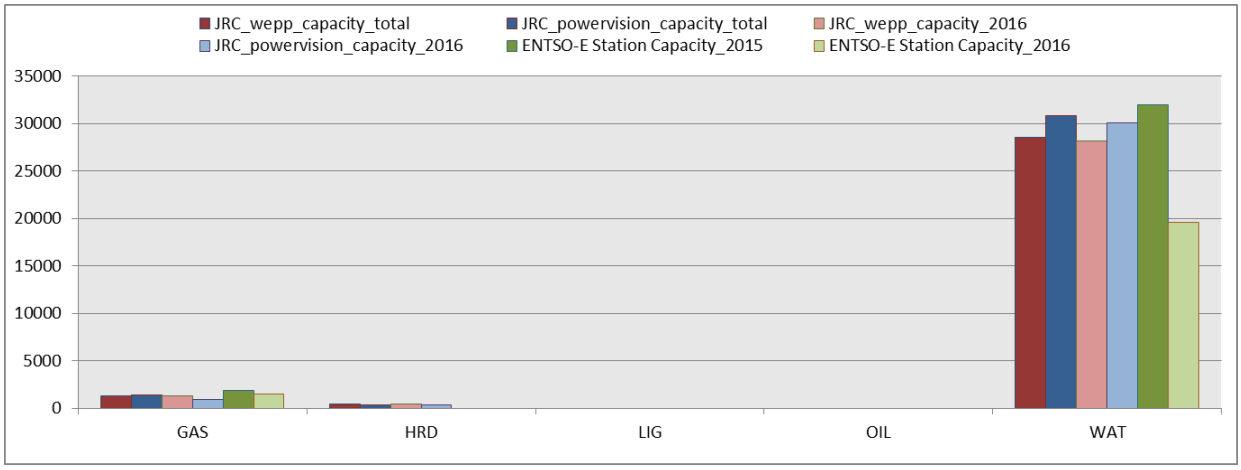
Figure 25 - Capacity (MW) aggregates per fuel in the Netherlands



5.19 Norway

Completeness level is very high for large power plants in Norway as indicated in Figure 26. ENTSO-E reports higher capacities for some thermal plants (i.e Mongstadt) while hydro capacity is probably linked to the respective reservoir levels as this varies significantly.

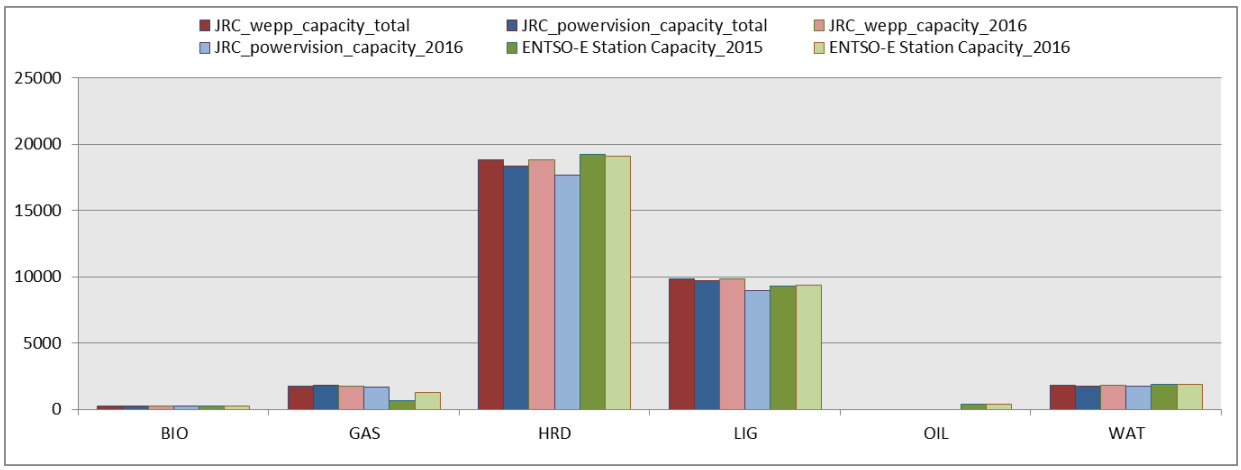
Figure 26 - Capacity (MW) aggregates per fuel in Norway



5.20 Poland

The JRC-PPDB is 100% complete on all major power plants in Poland as indicated in Figure 27. Moderate discrepancies exist on capacities reported by ENTSO-E and the equivalent figures reported in both databases.

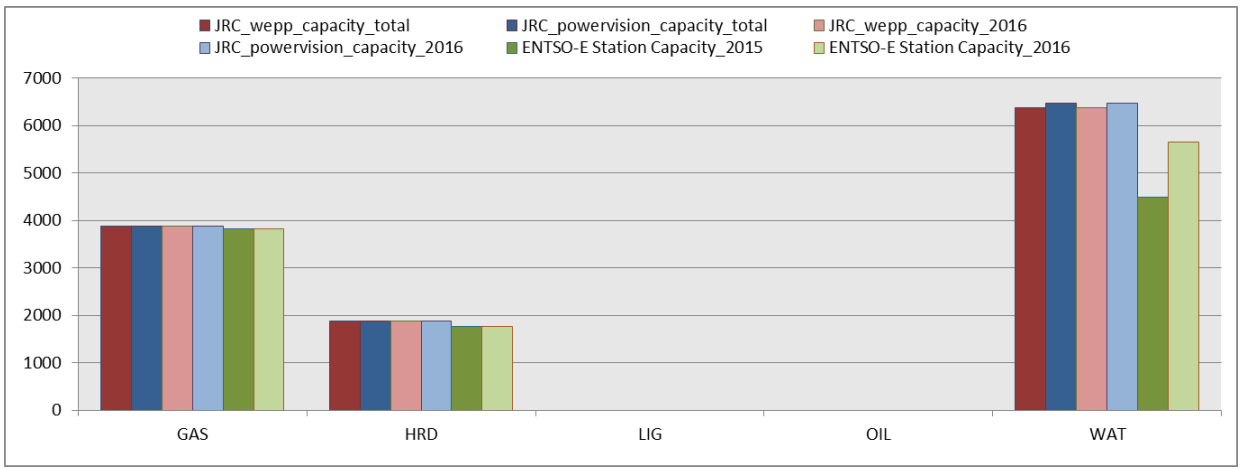
Figure 27 - Capacity (MW) aggregates per fuel in Poland



5.21 Portugal

The JRC-PPDB is 100% complete on all major power plants in the Portugal as indicated in Figure 28.

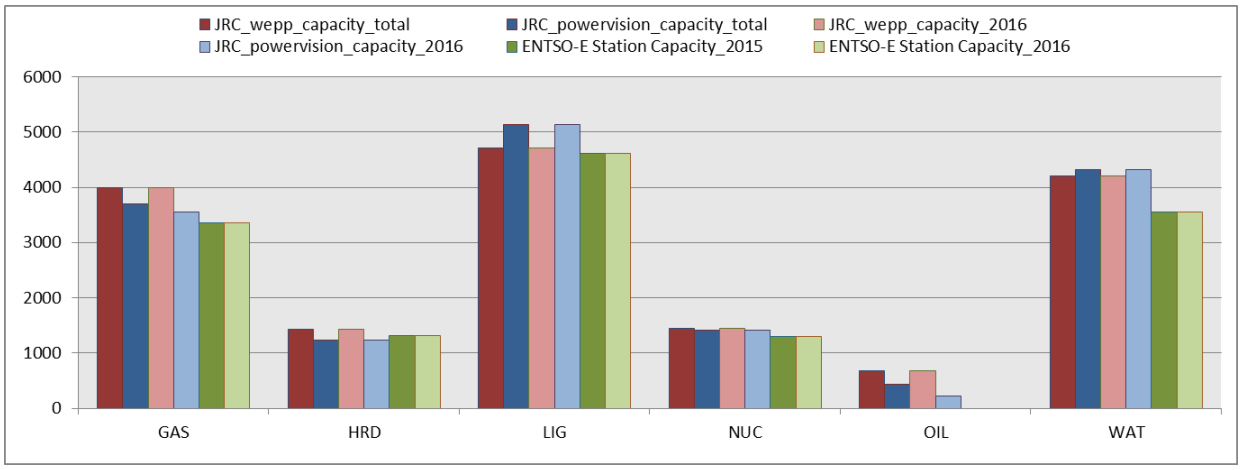
Figure 28 - Capacity (MW) aggregates per fuel in Portugal



5.22 Romania

The JRC-PPDB completeness level is very high for large power plants in Romania as indicated in Figure 29. It is likely that some gas or oil-fired capacity has been removed.

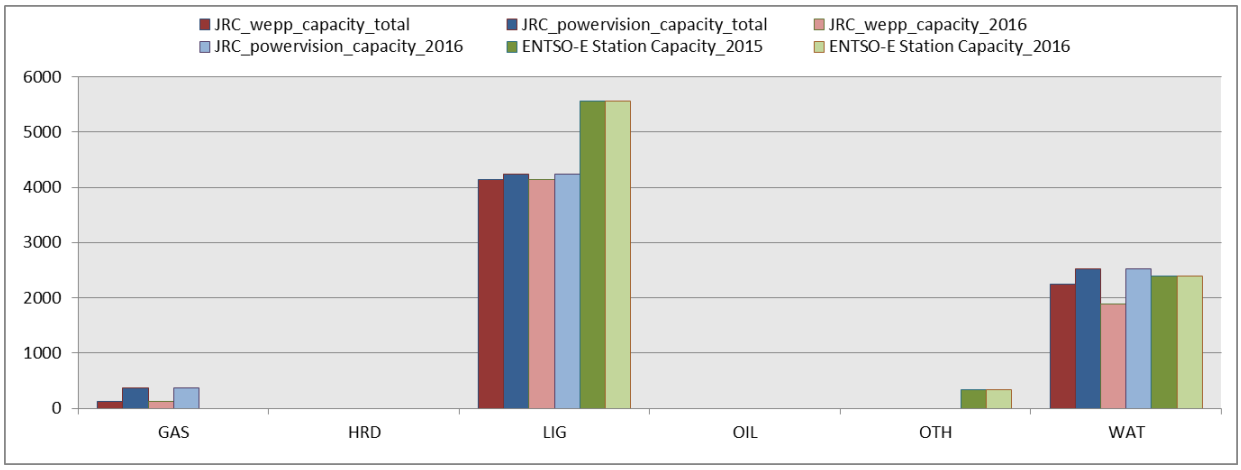
Figure 29 - Capacity (MW) aggregates per fuel in Romania



5.23 Serbia

The plant list is complete on Serbia as indicated in Figure 30. The surplus on lignite capacity is due to the reporting of lignite plants in Kosovo with Serbian EIC_codes.

Figure 30 - Capacity (MW) aggregates per fuel in Serbia

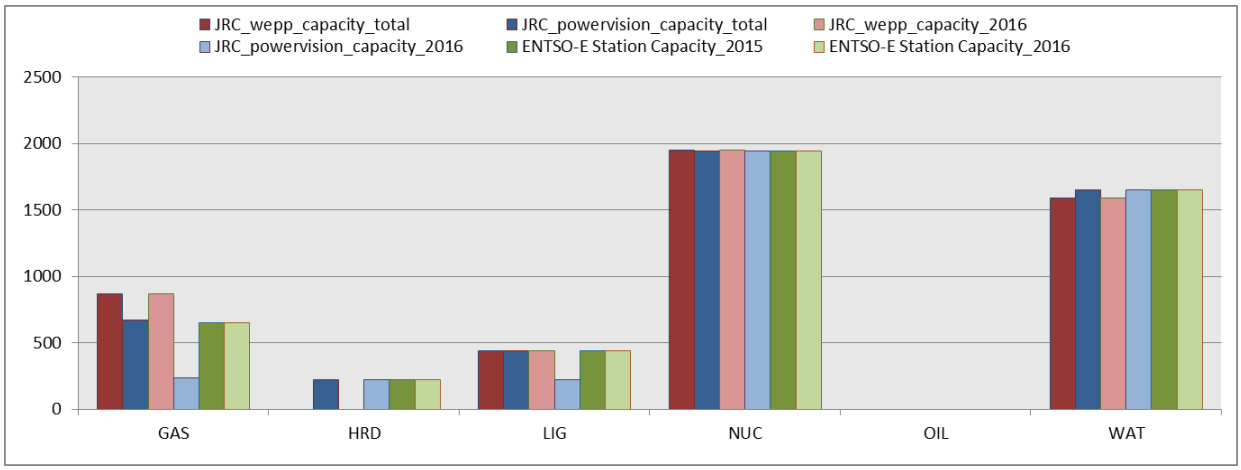


5.24 Slovakia

The plant list is complete on Slovakia for all major power plants as indicated in Figure 31. The following discrepancies are observed:

- Vojany (220 MW) is reported in one database as gas-fired, whereas ENTSO-E reports the plant as coal-fired.
- Malzenice CCGT(434 MW) is reported as "out of service".
- Novaky 3&4 are reported due for retirement in 2016 in one database.

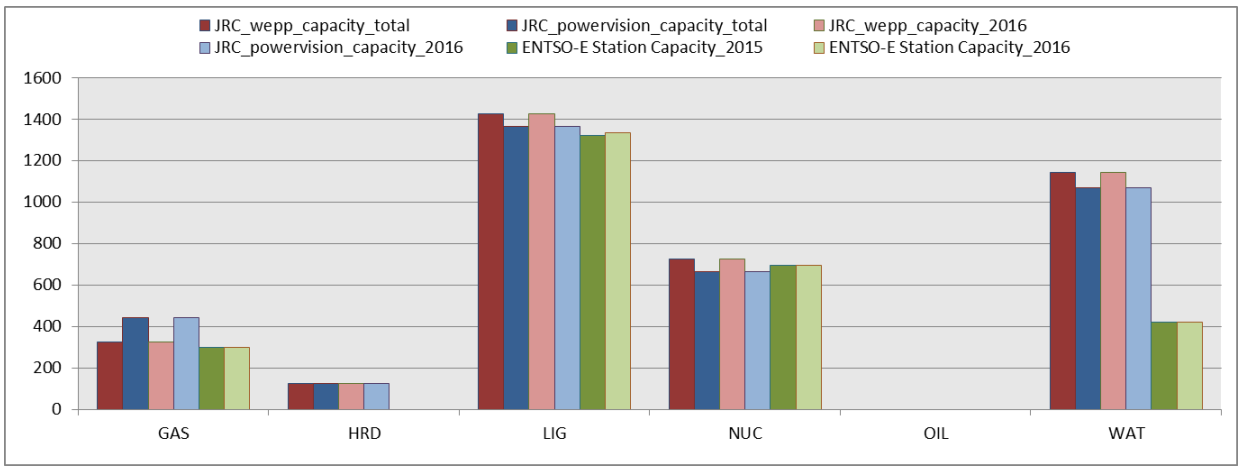
Figure 31 - Capacity (MW) aggregates per fuel in Slovakia



5.25 Slovenia

The plant list is complete on Slovenia as indicated in Figure 32. The Hydro lists are close to 100% of the installed capacity in the country.

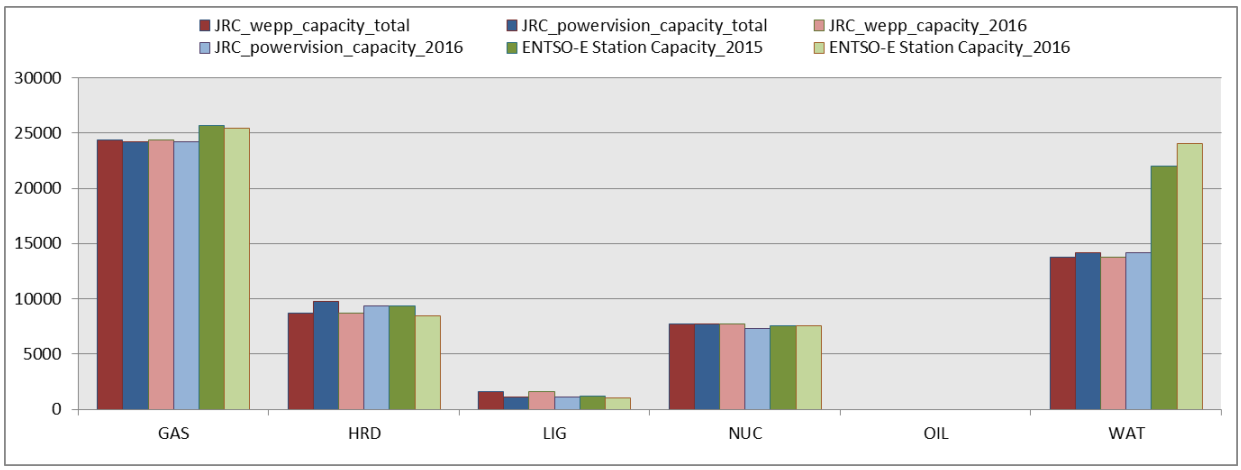
Figure 32 - Capacity (MW) aggregates per fuel in Slovenia



5.26 Spain

Completeness level is very high for large power plants in Spain as indicated in Figure 33. The hydro plants are reported by ENTSO-E at a threshold much lower than the 100 MWe. The missing hydro capacity is attributed to run-of river units below 9MW.

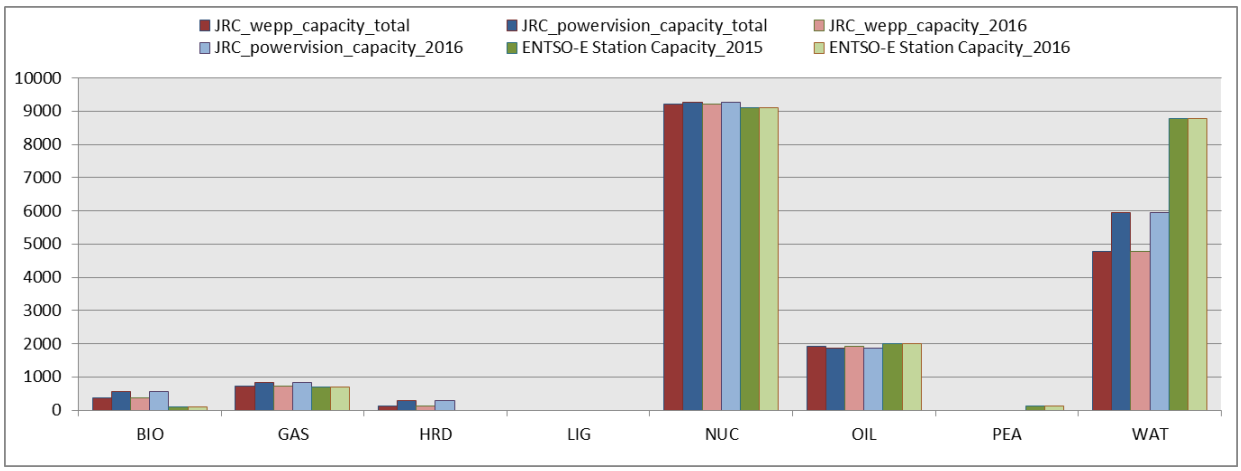
Figure 33 - Capacity (MW) aggregates per fuel in Spain



5.27 Sweden

Sweden is 100% complete on all major power plants as indicated in Figure 35. There is missing hydro capacity which is expected to be below the ENTSO-E reporting threshold.

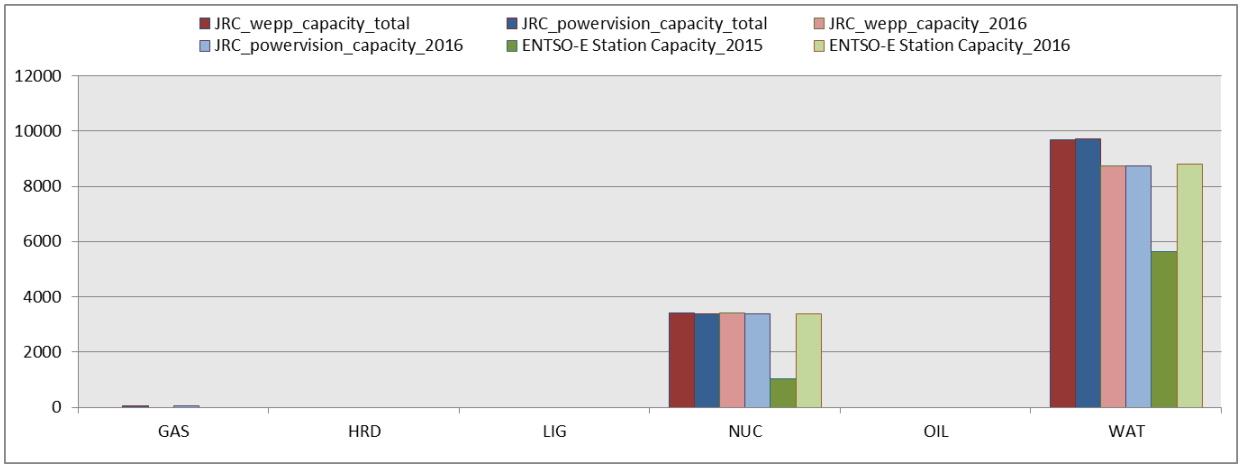
Figure 34 - Capacity (MW) aggregates per fuel in Sweden



5.28 Switzerland

Switzerland is 100% complete on all major power plants as indicated in Figure 35.

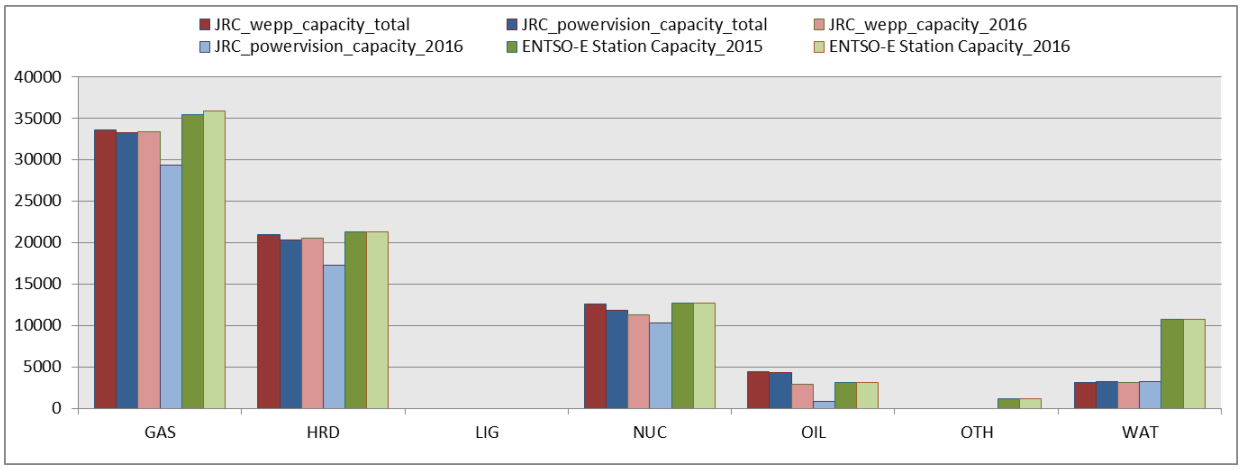
Figure 35 - Capacity (MW) aggregates per fuel in Switzerland



5.29 United Kingdom (Bidding zone GB)

The JRC-PPDB is 100% complete on all major power plants in the United Kingdom – bidding zone GB as indicated in Figure 36.

Figure 36 - Capacity (MW) aggregates per fuel in the United Kingdom



The discrepancy observed is explained by the following:

- One database reports Littlebrook 2x685 MW oil-fired units as due for retirement in 2015 and Grain 2 x 675 MW oil-fired units as retired in 2012.
- The capacity reported on several nuclear powerplants by ENTSO-E is approximately higher than the values in one database. Oldebury is reported by both databases as retired.
- Ferrybridge (1500 MW), Drax 1 (645 MW) and Eggborough 2 (500MW) coal fired are reported as due for retirement in 2015 whereas ENTSO-E reports production units at full capacity.

- One of the databases reports Peterhead 1 (1060 MW) as retired and Peterhead 2 (660 MW) and Keadby (724MW) as "standby".
- ENTSO-E appears to host mistaken reports by on several run-of river hydro units - all reported with a capacity of 999 MW.

6 Conclusions and further development

The present work has led to the creation of an interface between the information presently published by ENTSO-E and the information contained in two prominent commercial powerplant databases. This interface was used to create a power plant database which enables access to all the information available in these three sources, at any aggregation level required for energy system modelling or reporting. This information was further extended by applying class based characteristics not currently present in any of our databases (efficiency, transient performance and cost). The database is created with the update process in mind. Updates are introduced via automated scripts.

6.1 Further development

Our initial analysis indicates that the datasets are sufficiently complete for modelling the entire geographic area covered by JRC-PPDB. However the following improvements have the potential of providing significant added value:

1. Further investigation of the inconsistencies identified in the present report and further validation of the datasets.
2. Improvements to the classification of all plants, including nuclear plants, for applying extended plant parameters.
3. The integration of other datasets containing supplementary information.
4. Projection into the future of class-based tables, aligned with the main energy policy scenarios.

7 References

[1] "ENTSO-E Transparency Platform," [Online]. Available: https://transparency.entsoe.eu/.

[2] "open Power System Data," [Online]. Available: http://open-power-system-data.org/.

[3] J. Kjärstad and F. Johnsson, "The European power plant infrastructure—Presentation of the Chalmers energy infrastructure database with applications," *Energy Policy,* vol. 35, no. 7, pp. 3643-3664, July 2007.

[4] J. Deane, Á. Driscoll and B. O'Gallachóir, "Quantifying the impacts of national renewable electricity ambitions," *Renewable Energy,* vol. 80, pp. 604-609, August 2015.

[5] "Argus," [Online]. Available: http://www.argusmedia.com/power/.

[6] E. Michetti, "EUROPEAN ENERGY MARKETS TRANSPARENCY REPORT," Florence School of Regulation, 2012.

[7] S. Pfenninger, J. DeCarolis, L. Hirth, S. Quoilin and I. Staffell, "The importance of open data and software: Is energy research lagging behind?," *Energy Policy,* pp. 211-2015, 2017.

[8] S. Quoilin, I. H. Gonzalez and A. Zucker, "Modelling Future EU Power Systems Under High Shares of Renewables. The Dispa-SET 2.1 open-source model," Joint Research Centre, 2017.

[9] "WEPP," [Online]. Available: http://www.platts.com/products/world-electric-power-plants-database.

[10] "Powervision," [Online]. Available: http://www.platts.com/products/powervision.

[11] "Regulation 2009/714," [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32009R07141.

[12] "Regulation 2013/543," [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32013R0543.

[13] "ENTSO-E Energy Identification Codes," [Online]. Available: https://www.entsoe.eu/data/energy-identification-codes-eic/Pages/default.aspx.

[14] J. Leskovec, A. R. Milliway and J. D. Ullman, Mining of Massive Datasets, Stanford Univ.

[15] "ENTSO-E Scenario Outlook and Adequacy Forecast," 2015. [Online]. Available: https://www.entsoe.eu/Documents/SDC%20documents/SOAF/150630_SOAF_2015_publication_wcover.pdf.

[16] "Mineral and Energy Economy Research Institute of the Polish Academy of Sciences," [Online]. Available: https://www.min-pan.krakow.pl/en/.

[17] P. Kaszyński, J. Kamiński, P. Nalepka, M. Malec and P. Benalcazar, "Development of improvements to the inputs of the Dispa-SET model - Task 1: Estimation of technical parameters of European thermal power plants," 2017.

8 List of abbreviations and definitions

EIC	Energy identification code
ENTSO-E	European Network of Transmission System Operators for Electricity
IEM	Internal Energy Market
JRC	Joint Research Centre
JRC-PPDB	The JRC Power Plant Database
SO&AF	Scenario Outlook & Adequacy Forecast
TSO	Transmission System Operator
WEPP	World Electric Power plant Database

9 List of figures

10 List of tables

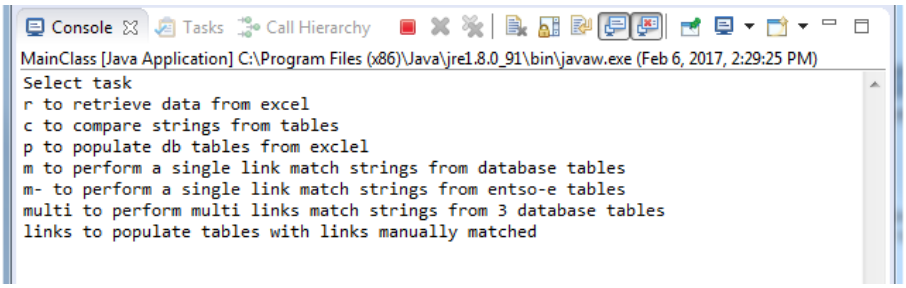
11 Annexes

11.1 Annex 1. PowerPlantDatabase code

Java code available in https://gitlab.jrc.nl/kanekon/ppdb-sql was written with the aim of

- Reading and comparing dataset records
- Populating a mysql database with data in an Excel file
- Populating the interim linkage tables entsoe_[db]_links

The following is a console menu informing the user of the options.



11.2 Annex 2. SQL code

After uploading the Database the script ***Run me after loading databases.sql*** adjusts the country names in order to comply with the conventions used in the JRC-PPDB

```
update powervision_full set Plant_Country = "FYR of Macedonia" where
Plant_Country like "Macedonia%";
update powervision_full set Plant_Country = "Ireland" where
Plant_State_Name like "%Northern Ireland%";
drop table if exists northernIrelandweppcodes;
create table northernIrelandweppcodes (select UNIT,UNITID from wepp_full
where COUNTRY like "%Northern Ireland%");
update wepp_full set COUNTRY = "Ireland" where COUNTRY  like "%Northern
Ireland%";
drop  table if exists Englandandwaleswepcodes;
create table Englandandwaleswepcodes (select UNIT,UNITID from wepp_full
where COUNTRY like "%ENGLAND &%");
update wepp_full set COUNTRY = "United Kingdom" where COUNTRY  like
"%ENGLAND &%";
drop table if exists Scotlandweppcodes;
create table Scotlandweppcodes (select UNIT,UNITID from wepp_full where
COUNTRY like "%Scotland%");
update wepp_full set COUNTRY = "United Kingdom" where COUNTRY  like
"%Scotland%";
update wepp_full set COUNTRY = "FYR of Macedonia" where COUNTRY like
"Macedonia%";
update wepp_full set COUNTRY = "BOSNIA AND HERZEGOVINA" where COUNTRY like
"BOSNIA%";
```

11.3 Creation of the JRC_PPDB_tables

After uploading the linkage tables the following code found in the script ***Run me after loading entsoe_dbs linking tables.sql*** creates the jrc_from_[db] tables

```
#1st remove non linked codes form powervision
drop table if exists entsoe_powervision_no_links;
create table entsoe_powervision_no_links select * from
ppdb.entsoe_powervision_links where id ="" or id like "#%";
delete FROM ppdb.entsoe_powervision_links where id ="" or id like "#%";
#2nd populated ids in wepp_links table
update  entsoe_wepp_links
left join wepp_full_2014 on
wepp_full_2014.UNIT= entsoe_wepp_links.unit_name
set id = wepp_full_2014.UNITID where id is not null and unit_name is not
null and unit_name <>"";
#3rd remove non linked codes from wepp
drop table if exists entsoe_wepp_2016_links ;
create table entsoe_wepp_2016_links select * from ppdb.entsoe_wepp_links
where id is null or  id ="" or id like "#%";
#3.1 populate ids in wepp_no_links table from wepp_2016
update  entsoe_wepp_2016_links
left join wepp_full on
wepp_full.UNIT= entsoe_wepp_2016_links.unit_name
set id = wepp_full.UNITID where unit_name is not null and unit_name <>"";

drop table if exists entsoe_wepp_no_links;
create table entsoe_wepp_no_links select * from ppdb.entsoe_wepp_links
where id is null or id ="" or id like "#%";
insert into entsoe_wepp_no_links select * FROM ppdb.entsoe_wepp_2016_links
where id is null or id ="" or id like "#%";
```

```
delete FROM ppdb.entsoe_wepp_links where id is null or id ="" or id like
"#%";
delete FROM ppdb.entsoe_wepp_2016_links where id is null or id ="" or id
like "#%";


#4th populate wepp_links with unit names
update  entsoe_wepp_links
left join wepp_full_2014 on
wepp_full_2014.UNITID=entsoe_wepp_links.id
set unit_name = wepp_full_2014.UNIT;

update  entsoe_wepp_2016_links
left join wepp_full on
wepp_full.UNITID=entsoe_wepp_2016_links.id
set unit_name = wepp_full.UNIT;


#5th populate wepp_links with plant names
update  entsoe_wepp_links
left join wepp_full_2014 on
wepp_full_2014.UNITID=entsoe_wepp_links.id
set plant_name = wepp_full_2014.PLANT;

update  entsoe_wepp_2016_links
left join wepp_full on
wepp_full.UNITID=entsoe_wepp_2016_links.id
set plant_name = wepp_full.PLANT;

#4.1 populate powervision_links with plant names
update  entsoe_powervision_links
left join powervision_full on
powervision_full.Unit_ID=entsoe_powervision_links.id
set entsoe_powervision_links.plant_name = powervision_full.Plant_Name;


# 6 entsoe_powervision_links_coord creation
#rename table entsoe_powervision_links_coord to
entsoe_powervision_links_coord_bak ;
drop table if exists  entsoe_powervision_links_coord ;
create table entsoe_powervision_links_coord
select eic_coords.eic_p,eic_coords.eic_g,Name_Code, plant_name, unit_name,
id, eic_coords.latitude,eic_coords.longitude from entsoe_powervision_links
right join eic_coords on
(entsoe_powervision_links.eic_g = eic_coords.eic_g and
entsoe_powervision_links.eic_p = eic_coords.eic_p);

#6.1 fill empty Name_Codes with powervision plant names
update entsoe_powervision_links left join powervision_full
on
entsoe_powervision_links.id = powervision_full.Unit_ID
set
Name_Code = powervision_full.plant_name  where Name_Code ="";

#6.2 fill empty Name_Codes with powervision plant names
update entsoe_wepp_links left join wepp_full_2014
on
entsoe_wepp_links.id = wepp_full_2014.UNITID
set
Name_Code = wepp_full_2014.PLANT  where Name_Code ="";
```

```
#7 create jrc_from_powervision_lim (see country request script)

#7.1 temprary table for many to one case
#select * from onemanypowervision;
drop table  if exists onemanypowervision;
create temporary table onemanypowervision
select
count(*) as countmto,
powervision_full.Unit,
any_value(id) as id,
any_value(Unit_Nameplate)
FROM powervision_full
right join ppdb.entsoe_powervision_links on id = Unit_ID
group by Unit;

#7.2 jrc_from powervision creation
#---------Tested query for creating the jrc codes -----------------------
drop table if exists  jrc_from_powervision;

CREATE TABLE `jrc_from_powervision` (
  `id` mediumint not null auto_increment,
  `jrc_code` varchar(102) DEFAULT NULL,
  `name_code` varchar(75) DEFAULT NULL,
  `eic_p` varchar(20) DEFAULT NULL,
  `eic_codes` text,
  `powervision_capacity` double(17,0) DEFAULT NULL,
  `plant_name` varchar(60) DEFAULT NULL,
  `Unit_Oper_Online_Date` date DEFAULT NULL,
  `technology_code` varchar(11) DEFAULT NULL,
  `fuel_code` varchar(11) DEFAULT NULL,
  `country` varchar(60) DEFAULT NULL,
  `country_code` varchar(2) DEFAULT NULL,
  `retirement` date DEFAULT NULL,
  `status` varchar(10) DEFAULT NULL,
  `cogen` varchar(1) DEFAULT NULL,
  `latitude` double DEFAULT NULL,
  `longitude` double DEFAULT NULL,
  primary key (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


insert into jrc_from_powervision(`jrc_code`,
  `name_code`,
  `eic_p`,
  `eic_codes`,
  `powervision_capacity`,
  `plant_name`,
  `Unit_Oper_Online_Date`,
  `technology_code`,
  `fuel_code`,
  `country`,
  `country_code`,
  `retirement`,
  `status`,
  `cogen`,
  `latitude`,
  `longitude`)
SELECT
min(concat(country_codes.country_code,"_",tech_powervision_dispa.Dispa_code
,"_",fuel_powervision_dispa.Dispa_code,"_",
entsoe_powervision_links.Name_Code)) as jrc_code,
```

```
min(entsoe_powervision_links.Name_Code) as name_code,
any_value(entsoe_powervision_links.eic_p) as eic_p,
group_concat(entsoe_powervision_links.eic_g separator ';') as eic_codes,
any_value(round(sum(powervision_full.Unit_Nameplate)/countmto,0)) as
powervision_capacity,
any_value(powervision_full.plant_name) as plant_name,
any_value(powervision_full.Unit_Oper_Online_Date) as Unit_Oper_Online_Date,
any_value(tech_powervision_dispa.Dispa_code) as technology_code,
any_value(fuel_powervision_dispa.Dispa_code) as fuel_code,
any_value(powervision_full.Plant_Country) as country,
any_value(country_codes.country_code) as country_code,
any_value(powervision_full.Unit_Oper_Retire_Date) as retirement,
any_value(powervision_full.Unit_Status_Code) as status,
any_value(powervision_full.Unit_Co_Gen) as cogen,
any_value(powervision_full.Plant_latitude) as latitude,
any_value(powervision_full.Plant_longitude) as longitude
FROM ppdb.powervision_full
right join entsoe_powervision_links on
powervision_full.Unit_ID=entsoe_powervision_links.id
left join onemanypowervision using(id)
left join  tech_powervision_dispa on (tech_powervision_dispa.Plant_Type=
powervision_full.Plant_Type and
tech_powervision_dispa.Unit_Prime_Mover=powervision_full.Unit_Prime_Mover)
left join fuel_powervision_dispa on
(fuel_powervision_dispa.Plant_Type=powervision_full.Plant_Type and
fuel_powervision_dispa.Unit_Primary_Fuel_Code =
powervision_full.Unit_Primary_Fuel_Code)
left join country_codes on country_codes.country_name =
powervision_full.Plant_Country
where entsoe_powervision_links.id<>""
#and powervision_full.Plant_Country like "Spain"
group by (case when entsoe_powervision_links.eic_p <>"" then
entsoe_powervision_links.eic_p end),
(case when entsoe_powervision_links.eic_g <>""
then (CASE WHEN powervision_full.Unit_Primary_Fuel_Code regexp("WAT") or
(powervision_full.Unit_Prime_Mover regexp("CC") and countmto>1)
      THEN powervision_full.Plant_Name
      ELSE entsoe_powervision_links.eic_g
      END)
else entsoe_powervision_links.Name_Code end),(case when
entsoe_powervision_links.eic_g <>"" then
(CASE WHEN powervision_full.Unit_Primary_Fuel_Code regexp("WAT") then
ppdb.powervision_full.Unit_PM_Tech_Detail_Desc end) end)
,powervision_full.Unit_Primary_Fuel_Code;
#---------Tested query for creating the jrc codes -----------------------

#7.3 jrc_from powervision_lim creation (Locations manually checked in the
waterflex project
drop table if exists jrc_from_powervision_lim;
create table jrc_from_powervision_lim
SELECT
min(concat(country_codes.country_code,"_",tech_powervision_dispa.Dispa_code
,"_",fuel_powervision_dispa.Dispa_code,"_",
entsoe_powervision_links_coord.Name_Code)) as jrc_code,
any_value(entsoe_powervision_links_coord.eic_p) as eic_p,
group_concat(entsoe_powervision_links_coord.eic_g separator ';') as
eic_codes,
any_value(sum(powervision_full.Unit_Nameplate)/countmto) as
powervision_capacity,
any_value(powervision_full.plant_name) as plant_name,
any_value(powervision_full.Unit_Oper_Online_Date) as Unit_Oper_Online_Date,
```

```
any_value(tech_powervision_dispa.Dispa_code) as technology_code,
any_value(fuel_powervision_dispa.Dispa_code) as fuel_code,
any_value(powervision_full.Plant_Country) as country,
any_value(country_codes.country_code) as country_code,
any_value(entsoe_powervision_links_coord.latitude) as latitude,
any_value(entsoe_powervision_links_coord.longitude) as longitude
FROM ppdb.powervision_full
right join entsoe_powervision_links_coord on
powervision_full.Unit_ID=entsoe_powervision_links_coord.id
left join onemanypowervision using(id)
left join  tech_powervision_dispa on (tech_powervision_dispa.Plant_Type=
powervision_full.Plant_Type and
tech_powervision_dispa.Unit_Prime_Mover=powervision_full.Unit_Prime_Mover)
left join fuel_powervision_dispa on
(fuel_powervision_dispa.Plant_Type=powervision_full.Plant_Type and
fuel_powervision_dispa.Unit_Primary_Fuel_Code =
powervision_full.Unit_Primary_Fuel_Code)
left join country_codes on country_codes.country_name =
powervision_full.Plant_Country
where entsoe_powervision_links_coord.id<>""
#and powervision_full.Plant_Country like "Croatia"
group by
(case when entsoe_powervision_links_coord.eic_p <>"" then
entsoe_powervision_links_coord.eic_p end),
(case when entsoe_powervision_links_coord.eic_g <>""
then (CASE WHEN powervision_full.Unit_Primary_Fuel_Code regexp("WAT") or
(powervision_full.Unit_Prime_Mover regexp("CC") and countmto>1)
      THEN powervision_full.Plant_Name
      ELSE entsoe_powervision_links_coord.eic_g
      END) else entsoe_powervision_links_coord.Name_Code end),(case when
entsoe_powervision_links_coord.eic_g <>"" then
(CASE WHEN powervision_full.Unit_Primary_Fuel_Code regexp("WAT") then
ppdb.powervision_full.Unit_PM_Tech_Detail_Desc end) end)
,powervision_full.Unit_Primary_Fuel_Code;


#8 update
update jrc_from_powervision right join coordupdated using(plant_name) set
jrc_from_powervision.latitude = coordupdated.latitude;
update jrc_from_powervision right join coordupdated using(plant_name) set
jrc_from_powervision.longitude = coordupdated.longitude;


update jrc_from_powervision right join jrc_from_powervision_lim
on
jrc_from_powervision.jrc_code = jrc_from_powervision_lim.jrc_code
and
jrc_from_powervision.plant_name=jrc_from_powervision_lim.plant_name
set jrc_from_powervision.latitude = jrc_from_powervision_lim .latitude;

update jrc_from_powervision right join jrc_from_powervision_lim
on
jrc_from_powervision.jrc_code = jrc_from_powervision_lim.jrc_code
and
jrc_from_powervision.plant_name=jrc_from_powervision_lim.plant_name
set jrc_from_powervision.longitude = jrc_from_powervision_lim .longitude;

#9 Create frc_from_wepp tables
#9.1 temporary table for many to one cases
drop table  if exists onemanywepp ;
create temporary table onemanywepp
```

```
select
count(*) as countmto,
wepp_full_2014.UNIT as UNIT,
any_value(id)
FROM wepp_full_2014
right join ppdb.entsoe_wepp_links on
wepp_full_2014.UNIT=entsoe_wepp_links.unit_name
#where wepp_full_2014.COUNTRY like "Austria"
group by UNIT;

#9.2 creation of jrc_from_wepp
drop table if exists jrc_from_wepp;

CREATE TABLE `jrc_from_wepp` (
  `id` mediumint not null auto_increment,
  `name_code` varchar(75) DEFAULT NULL,
  `eic_p` varchar(20) NOT NULL,
  `eic_codes` text,
  `wepp_capacity` double(17,0) DEFAULT NULL,
  `wepp_plant` varchar(40) DEFAULT NULL,
  `year` int(11) DEFAULT NULL,
  `technology_code` varchar(11) DEFAULT NULL,
  `fuel_code` varchar(11) DEFAULT NULL,
  `fuelgroup` text,
  `country` varchar(40) DEFAULT NULL,
  `country_code` varchar(2) DEFAULT NULL,
  `RETIRE` int(11) DEFAULT NULL,
  `STATUS` varchar(10) DEFAULT NULL,
  `STYPE` varchar(10) DEFAULT NULL,
  `cogen` text,
   primary key (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into jrc_from_wepp (
  `name_code`,
  `eic_p`,
  `eic_codes`,
  `wepp_capacity`,
  `wepp_plant`,
  `year` ,
  `technology_code`,
  `fuel_code`,
  `fuelgroup`,
  `country`,
  `country_code`,
  `RETIRE`,
  `STATUS`,
  `STYPE`,
  `cogen` )
SELECT min(entsoe_wepp_links.Name_Code) AS name_code,
      ANY_VALUE(entsoe_wepp_links.eic_p) as eic_p,
    GROUP_CONCAT(entsoe_wepp_links.eic_g
        SEPARATOR ';') AS eic_codes,
    (ROUND(SUM(wepp_full_2014.MW)/countmto, 0)) AS wepp_capacity,
    ANY_VALUE(wepp_full_2014.PLANT) AS wepp_plant,
    ANY_VALUE(YEAR) AS year,
    ANY_VALUE(tech_wepp_dispa.Dispa_code) AS technology_code,
    ANY_VALUE(fuel_wepp_dispa.Dispa_code) AS fuel_code,
    GROUP_CONCAT(wepp_full_2014.FUEL) AS fuelgroup,
    ANY_VALUE(wepp_full_2014.COUNTRY) AS country,
      any_value(country_codes.country_code) as country_code,
```

```
    ANY_VALUE(wepp_full_2014.RETIRE) AS RETIRE,
    ANY_VALUE(wepp_full_2014.STATUS) AS STATUS,
    ANY_VALUE(wepp_full_2014.STYPE) AS STYPE,
      group_concat(wepp_full_2014.UTYPE separator ':') AS cogen

    FROM
    ppdb.wepp_full_2014
        RIGHT JOIN
    entsoe_wepp_links ON wepp_full_2014.UNIT = entsoe_wepp_links.unit_name
        LEFT JOIN
    onemanywepp USING (UNIT)
        LEFT JOIN
    tech_wepp_dispa ON (tech_wepp_dispa.UTYPE = wepp_full_2014.UTYPE
        AND tech_wepp_dispa.FUELTYPE = wepp_full_2014.FUELTYPE)
        LEFT JOIN
    fuel_wepp_dispa ON (fuel_wepp_dispa.FUEL = wepp_full_2014.FUEL
        AND fuel_wepp_dispa.FUELTYPE = wepp_full_2014.FUELTYPE)
        LEFT JOIN
    country_codes ON country_codes.country_name = wepp_full_2014.COUNTRY
WHERE
    entsoe_wepp_links.unit_name <> ''
GROUP BY
(case when entsoe_wepp_links.eic_p <>"" then entsoe_wepp_links.eic_p end),
(case when entsoe_wepp_links.eic_g <>""
            then (CASE
                        WHEN
                              wepp_full_2014.FUEL REGEXP ('WAT')
                              OR ((wepp_full_2014.FUEL REGEXP
('WSTH|GAS|LNG|OIL')
                              AND wepp_full_2014.UTYPE REGEXP
('GT/C|ST/C|ST/S|GT/R')))
                        THEN
                              ppdb.wepp_full_2014.PLANT
                        ELSE
                              entsoe_wepp_links.eic_g
                        END)
            ELSE entsoe_wepp_links.Name_code
END) , (CASE
    WHEN wepp_full_2014.FUEL REGEXP ('WAT') THEN
ppdb.wepp_full_2014.Fueltype
END) , (CASE
    WHEN wepp_full_2014.UTYPE REGEXP ('ST/S|ST/T|GT/T') THEN
ppdb.wepp_full_2014.FUEL
END);

#9.3 temporary table for many to one cases
drop table  if exists onemanywepp ;
create temporary table onemanywepp
select
count(*) as countmto,
wepp_full.UNIT,
any_value(id)
FROM wepp_full
right join ppdb.entsoe_wepp_2016_links on
wepp_full.UNIT=entsoe_wepp_2016_links.unit_name
#where wepp_full_2014.COUNTRY like "Austria"
group by UNIT;

#9.4 creation of jrc_from_wepp_2016
drop table if exists jrc_from_wepp_2016;
```

```sql
CREATE TABLE jrc_from_wepp_2016 SELECT
min(entsoe_wepp_2016_links.Name_Code) AS name_code,
      ANY_VALUE(entsoe_wepp_2016_links.eic_p) as eic_p,
    GROUP_CONCAT(entsoe_wepp_2016_links.eic_g
        SEPARATOR ';') AS eic_codes,
    ANY_VALUE(ROUND(SUM(wepp_full.MW) / countmto, 0)) AS wepp_capacity,
    ANY_VALUE(wepp_full.PLANT) AS wepp_plant,
    ANY_VALUE(YEAR) AS year,
    ANY_VALUE(tech_wepp_dispa.Dispa_code) AS technology_code,
    ANY_VALUE(fuel_wepp_dispa.Dispa_code) AS fuel_code,
    GROUP_CONCAT(wepp_full.FUEL) AS fuelgroup,
    ANY_VALUE(wepp_full.COUNTRY) AS country,
      any_value(country_codes.country_code) as country_code,
    ANY_VALUE(wepp_full.RETIRE) AS RETIRE,
    ANY_VALUE(wepp_full.STATUS) AS STATUS,
    ANY_VALUE(wepp_full.STYPE) AS STYPE,
      group_concat(wepp_full.UTYPE separator ':') AS cogen

    FROM
    ppdb.wepp_full
        RIGHT JOIN
    entsoe_wepp_2016_links ON wepp_full.UNIT =
entsoe_wepp_2016_links.unit_name
        LEFT JOIN
    onemanywepp USING (UNIT)
        LEFT JOIN
    tech_wepp_dispa ON (tech_wepp_dispa.UTYPE = wepp_full.UTYPE
        AND tech_wepp_dispa.FUELTYPE = wepp_full.FUELTYPE)
        LEFT JOIN
    fuel_wepp_dispa ON (fuel_wepp_dispa.FUEL = wepp_full.FUEL
        AND fuel_wepp_dispa.FUELTYPE = wepp_full.FUELTYPE)
        LEFT JOIN
    country_codes ON country_codes.country_name = wepp_full.COUNTRY
WHERE
    entsoe_wepp_2016_links.unit_name <> ''
GROUP BY
(case when entsoe_wepp_2016_links.eic_p <>"" then
entsoe_wepp_2016_links.eic_p end),
(case when entsoe_wepp_2016_links.eic_g <>""
            then (CASE
                        WHEN
                              wepp_full.FUEL REGEXP ('WAT')
                              OR ((wepp_full.FUEL REGEXP
('WSTH|GAS|LNG|OIL')
                              AND wepp_full.UTYPE REGEXP
('GT/C|ST/C|ST/S|GT/R')))
                        THEN
                              ppdb.wepp_full.PLANT
                        ELSE
                              entsoe_wepp_2016_links.eic_g
                        END)
            ELSE entsoe_wepp_2016_links.Name_code
END) , (CASE
    WHEN wepp_full.FUEL REGEXP ('WAT') THEN ppdb.wepp_full.Fueltype
END) , (CASE
    WHEN wepp_full.UTYPE REGEXP ('ST/S|ST/T|GT/T') THEN ppdb.wepp_full.FUEL
END);
```

```
insert into ppdb.jrc_from_wepp(
  `name_code`,
  `eic_p`,
  `eic_codes`,
  `wepp_capacity`,
  `wepp_plant`,
  `year` ,
  `technology_code`,
  `fuel_code`,
  `fuelgroup`,
  `country`,
  `country_code`,
  `RETIRE`,
  `STATUS`,
  `STYPE`,
  `cogen` ) (SELECT * FROM ppdb.jrc_from_wepp_2016);

#9.3 fix wsth from grouped ccgts
update jrc_from_wepp set fuel_code = "GAS" where fuel_code like "WST" and
fuelgroup regexp"GAS";
update jrc_from_wepp set fuel_code = "OIL" where fuel_code like "WST" and
fuelgroup regexp"OIL";
update jrc_from_wepp set fuel_code = "GAS" where fuel_code like "WST" and
fuelgroup regexp"LNG";
update jrc_from_wepp set cogen = "Y" where cogen like "ST/S";
update jrc_from_wepp set cogen = "Y" where cogen like "GT/S";
update jrc_from_wepp set cogen = "N" where cogen not like "Y";
update jrc_from_wepp set year =1901 where year is null;

#9.4 update run of river info
update jrc_from_wepp left join entsoe_station_capacities using(eic_p)
set technology_code = "HROR" where entsoe_station_capacities.Type like
"%Run_of%";

update jrc_from_powervision left join entsoe_station_capacities
using(eic_p)
set technology_code = "HROR" where entsoe_station_capacities.Type like
"%Run_of%";


#10 duplicate name_code treatment
drop  table if exists duplicates_powervision;
create temporary table duplicates_powervision
select name_code,
count(name_code) as occ,
group_concat(plant_name),
group_concat(technology_code),
group_concat(fuel_code)
from jrc_from_powervision
group by name_code ;
delete from duplicates_powervision where occ=1;
select * from duplicates_powervision ;

drop  table if exists duplicates_wepp;
create temporary table duplicates_wepp
select name_code,
count(name_code) as occ,
group_concat(wepp_plant),
group_concat(technology_code),
group_concat(fuel_code)
from jrc_from_wepp
```

```
group by name_code ;
delete from duplicates_wepp where occ=1;
select * from duplicates_wepp ;

update jrc_from_wepp set name_code =
concat(name_code,case when jrc_from_wepp.technology_code regexp("GTUR")
then
"GTUR" else "" end)
where name_code in (select name_code from duplicates_wepp);
update jrc_from_wepp set name_code =
concat(name_code,case when jrc_from_wepp.technology_code regexp("HPHS")
then
"PS" else "" end)
where name_code in (select name_code from duplicates_wepp);

update jrc_from_powervision set name_code =
concat(name_code,case when jrc_from_powervision.technology_code
regexp("GTUR") then
"GTUR" else "" end)where name_code in (select name_code from
duplicates_powervision);

update jrc_from_powervision set name_code =
concat(name_code,case when jrc_from_powervision.technology_code
regexp("HPHS") then
"PS" else "" end)where name_code in (select name_code from
duplicates_powervision);

drop table if exists primary_wepp;
create table primary_wepp (
`idn` mediumint not null auto_increment,
`id` mediumint not null,
`name_code` varchar(75),
 primary key (`name_code`, `idn`)
)ENGINE MyISAM;

insert into primary_wepp(`id`,`name_code`) select id, name_code from
jrc_from_wepp;

update jrc_from_wepp left join primary_wepp
using(`id`)
set jrc_from_wepp.name_code =concat(primary_wepp.name_code,"_",idn)
where primary_wepp.idn>1;

drop table if exists primary_powervision;
create table primary_powervision (
`idn` mediumint not null auto_increment,
`id` mediumint not null,
`name_code` varchar(75),
 primary key (`name_code`, `idn`)
)ENGINE MyISAM;

insert into primary_powervision(`id`,`name_code`) select id, name_code from
jrc_from_powervision;

update jrc_from_powervision left join primary_powervision
using(`id`)
set jrc_from_powervision.name_code
=concat(primary_powervision.name_code,"_",idn)

where primary_powervision.idn>1;
```

11.4 Aggregate Capacity tables

The script capacity_check_table.sql should be run after any update of the jrc_from_[db] tables.

```
/*
delete FROM ppdb.entsoe_cap ;
update entsoe_cap set capacity_type = "Coal" where capacity_type = "Hard
Coal";
update entsoe_cap set capacity_type = "Hydro" where capacity_type = "Hydro
power (total)";
update entsoe_cap set capacity_type = "Nuclear" where capacity_type =
"Nuclear Power";
update entsoe_cap set country = "SLOVAKIA" where country = "SLOVAK
REPUBLIC";
update entsoe_cap set country = "BOSNIA AND HERZEGOVINA" where country =
"BOSNIA-HERZEGOVINA";
update entsoe_cap set country = "IRELAND_N" where country = "GB NORTHERN
IRELAND";
update entsoe_cap set country = "UNITED KINGDOM" where country = "GREAT
BRITAIN";
update entsoe_cap set country = "NETHERLANDS" where country = "THE
NETHERLANDS";
update jrc_from_powervision set Plant_Country = "FYR OF MACEDONIA" where
Plant_Country = "Macedonia";
*/

#GB NORTHERN IRELAND

#update entsoe_cap set capacity_type =(select  Dispa_Code  from
fuel_codes_soaf where fuel_codes_soaf.name like capacity_type) where
capacity_type
regexp("Nuclear|Lignite|Coal|Oil|Wind|Solar|Biomass|Hydro|Gas");

# 1create interin tables
drop  table if exists countries_temp;
create temporary table countries_temp select  distinct(country) from
jrc_from_powervision;

#select * from soafcap;
drop  table if exists soafcap;
create temporary table soafcap
select
jrc_from_powervision.country,
fuel_code,
entsoe_cap.Y2016w as soaf_capacity
from jrc_from_powervision
left join fuel_codes_soaf on fuel_codes_soaf.Dispa_code like
jrc_from_powervision.fuel_code
left join entsoe_cap on jrc_from_powervision.country like
entsoe_cap.country and fuel_codes_soaf.name like entsoe_cap.capacity_type
group by  jrc_from_powervision.country,fuel_code ;


#select * from powvcap;
drop  table if exists powvcap;
create temporary table powvcap
select
country,
fuel_code,
sum(round (powervision_capacity,0)) as jrc_powervision_capacity
```

```
from jrc_from_powervision
group by  jrc_from_powervision.country,fuel_code ;
alter table powvcap add Unique Index country_fuel(country, fuel_code);
insert ignore into powvcap (country, fuel_code,jrc_powervision_capacity)
select country,"SUN",null from countries_temp;
insert ignore into powvcap (country, fuel_code,jrc_powervision_capacity)
select country,"BIO",null from countries_temp;
insert ignore into powvcap (country, fuel_code,jrc_powervision_capacity)
select country,"WIN",null from countries_temp;

#select * from powvcap2015;
drop  table if exists powvcap2015;
create temporary table powvcap2015
select
country,
fuel_code,
sum(round (powervision_capacity,0)) as jrc_powervision_capacity
from jrc_from_powervision
where
(jrc_from_powervision.retirement is null or
jrc_from_powervision.retirement>"2014-12-31") and
(jrc_from_powervision.Unit_Oper_Online_Date is null or
jrc_from_powervision.Unit_Oper_Online_Date<"2015-12-31")

group by  jrc_from_powervision.country,fuel_code ;
alter table powvcap2015 add Unique Index country_fuel(country, fuel_code);
insert ignore into powvcap2015 (country,
fuel_code,jrc_powervision_capacity)
select country,"SUN",null from countries_temp;
insert ignore into powvcap2015 (country,
fuel_code,jrc_powervision_capacity)
select country,"BIO",null from countries_temp;
insert ignore into powvcap2015 (country,
fuel_code,jrc_powervision_capacity)
select country,"WIN",null from countries_temp;

#select * from powvcap2015;
drop  table if exists powvcap2016;
create temporary table powvcap2016
select
country,
fuel_code,
sum(round (powervision_capacity,0)) as jrc_powervision_capacity
from jrc_from_powervision
where
(jrc_from_powervision.retirement is null or
jrc_from_powervision.retirement>"2015-12-31") and
(jrc_from_powervision.Unit_Oper_Online_Date is null or
jrc_from_powervision.Unit_Oper_Online_Date<"2016-12-31")

group by  jrc_from_powervision.country,fuel_code ;
alter table powvcap2016 add Unique Index country_fuel(country, fuel_code);
insert ignore into powvcap2016 (country,
fuel_code,jrc_powervision_capacity)
select country,"SUN",null from countries_temp;
insert ignore into powvcap2016 (country,
fuel_code,jrc_powervision_capacity)
select country,"BIO",null from countries_temp;
insert ignore into powvcap2016 (country,
fuel_code,jrc_powervision_capacity)
select country,"WIN",null from countries_temp;
```

```
#select * from weppcap;
drop  table if exists weppcap;
create temporary table weppcap select
country,
fuel_code,
sum(round (wepp_capacity,0)) as jrc_wepp_capacity
from jrc_from_wepp
group by country, fuel_code;
alter table weppcap add Unique Index country_fuel(country, fuel_code);
insert ignore into weppcap (country, fuel_code,jrc_wepp_capacity)
select country,"SUN",null from countries_temp;
insert ignore into weppcap (country, fuel_code,jrc_wepp_capacity)
select country,"BIO",null from countries_temp;
insert ignore into weppcap (country, fuel_code,jrc_wepp_capacity)
select country,"WIN",null from countries_temp;

#select * from weppcap2015;
drop  table if exists weppcap2015;
create temporary table weppcap2015 select
country,
fuel_code,
sum(round (wepp_capacity,0)) as jrc_wepp_capacity
from jrc_from_wepp
where (jrc_from_wepp.RETIRE is null or jrc_from_wepp.RETIRE >2015) and
jrc_from_wepp.STATUS not like "RET" and
(jrc_from_wepp.year is null or jrc_from_wepp.year <2016)
group by country, fuel_code;
alter table weppcap2015 add Unique Index country_fuel(country, fuel_code);
insert ignore into weppcap2015 (country, fuel_code,jrc_wepp_capacity)
select country,"SUN",null from countries_temp;
insert ignore into weppcap2015 (country, fuel_code,jrc_wepp_capacity)
select country,"BIO",null from countries_temp;
insert ignore into weppcap2015 (country, fuel_code,jrc_wepp_capacity)
select country,"WIN",null from countries_temp;

#select * from weppcap2016;
drop  table if exists weppcap2016;
create temporary table weppcap2016 select
country,
fuel_code,
sum(round (wepp_capacity,0)) as jrc_wepp_capacity
from jrc_from_wepp
where (jrc_from_wepp.RETIRE is null or jrc_from_wepp.RETIRE >2016) and
jrc_from_wepp.STATUS not like "RET" and
(jrc_from_wepp.year is null or jrc_from_wepp.year <2017)
group by country, fuel_code;
alter table weppcap2016 add Unique Index country_fuel(country, fuel_code);
insert ignore into weppcap2016 (country, fuel_code,jrc_wepp_capacity)
select country,"SUN",null from countries_temp;
insert ignore into weppcap2016 (country, fuel_code,jrc_wepp_capacity)
select country,"BIO",null from countries_temp;
insert ignore into weppcap2016 (country, fuel_code,jrc_wepp_capacity)
select country,"WIN",null from countries_temp;


#2 Create table with entsoe capacities with Dispaset fuel grouping from
entsoe_capacities which is the raw input
#table from entsoe
drop table if exists entsoe_capacities_2015;
create table entsoe_capacities_2015
```

```
select
entsoe_capacities.AreaName,
entsoe_country_mapping.country,
Dispa_code,ProductionType_Name,
AggregatedInstalledCapacity
from
entsoe_capacities
left join fuel_codes_soaf on entsoe_capacities.ProductionType_Name regexp
(fuel_codes_soaf.entsoe_code)
left join ppdb.entsoe_country_mapping on entsoe_capacities.AreaName
regexp(entsoe_country_mapping.area_name);
#where AreaName regexp(SELECT area_name FROM ppdb.entsoe_country_mapping
where country like "Ireland");
#2.1 clean null linkages (Duplicates)
delete from entsoe_capacities_2015  where country is null;

drop table if exists caps;
create temporary table caps
select
country,
Dispa_code,
sum(AggregatedInstalledCapacity) as entsoe_tp_capacity
from
entsoe_capacities_2015 group by country, Dispa_code
;

#select * from caps where Dispa_code like "SUN";
alter table caps add Unique Index country_fuel(country, Dispa_code);
insert ignore into caps (country, Dispa_code,entsoe_tp_capacity)
select country,"SUN",null from countries_temp;
insert ignore into caps (country, Dispa_code,entsoe_tp_capacity)
select country,"BIO",null from countries_temp;

drop table if exists capacity_aggregated_fuel;
create table capacity_aggregated_fuel
select
powvcap.country,
powvcap.fuel_code,
weppcap.jrc_wepp_capacity as PPDB_wepp_capacity,
weppcap2015.jrc_wepp_capacity as PPDB_wepp_capacity_2015,
weppcap2016.jrc_wepp_capacity as PPDB_wepp_capacity_2016,
powvcap.jrc_powervision_capacity as PPDB_powervision_capacity,
powvcap2015.jrc_powervision_capacity PPDB_powervision_capacity_2015,
powvcap2016.jrc_powervision_capacity PPDB_powervision_capacity_2016,
soaf_capacity,
entsoe_tp_capacity
from powvcap
left join powvcap2016 on powvcap.country  like  powvcap2016.country and
powvcap.fuel_code  like  powvcap2016.fuel_code
left join powvcap2015 on powvcap.country  like  powvcap2015.country and
powvcap.fuel_code  like  powvcap2015.fuel_code
left join weppcap on powvcap.country  like  weppcap.country and
powvcap.fuel_code  like  weppcap.fuel_code
left join weppcap2016 on powvcap.country  like  weppcap2016.country and
powvcap.fuel_code  like  weppcap2016.fuel_code
left join weppcap2015 on powvcap.country  like  weppcap2015.country and
powvcap.fuel_code  like  weppcap2015.fuel_code
left join caps on caps.country like weppcap.country and caps.Dispa_code
like weppcap.fuel_code
left join soafcap on powvcap.country  like  soafcap.country and
powvcap.fuel_code  like  soafcap.fuel_code;
```

```
delete from capacity_aggregated_fuel where country is null;
```

The script capacity_check_table2.sql should be run after any update of the jrc_from_[db] tables.

```
drop table if exists sta;
create temporary table sta
select
area,
Type,
sum(Capacity_2015) as Capacity_2015, sum(Capacity_2016) as Capacity_2016
FROM ppdb.entsoe_station_capacities group by area,Type;

drop table if exists entsoe_station_capacities_agg_ft;
create table entsoe_station_capacities_agg_ft
select
ppdb.country_codes.country_name as country,
Type,
fuel_codes_soaf.Dispa_code as Dispa_code,
Capacity_2015,
Capacity_2016
from
sta
left join fuel_codes_soaf on sta.Type regexp (fuel_codes_soaf.entsoe_code)
left join ppdb.country_codes on sta.area =ppdb.country_codes.country_area;

delete from entsoe_station_capacities_agg_ft where country regexp
("Austria|Luxemb");
update entsoe_station_capacities_agg_ft set country =
"Germany|Austria|Luxembourg" where country like "Germany";

drop table if exists entsoe_station_capacities_agg_f;
create table entsoe_station_capacities_agg_f
select
country,
Dispa_code,
sum(Capacity_2015) as Capacity_2015,
sum(Capacity_2016) as Capacity_2016
from entsoe_station_capacities_agg_ft
group by country, Dispa_code;

drop  table if exists countries_temp;
create temporary table countries_temp select  distinct(country) from
entsoe_station_capacities_agg_f;
alter table entsoe_station_capacities_agg_f add Unique Index
country_fuel(country, Dispa_code);
insert ignore into entsoe_station_capacities_agg_f (country,
Dispa_code,Capacity_2015,Capacity_2016)
select country,"LIG",null,null from countries_temp;
insert ignore into entsoe_station_capacities_agg_f (country,
Dispa_code,Capacity_2015,Capacity_2016)
select country,"GAS",null,null from countries_temp;
insert ignore into entsoe_station_capacities_agg_f (country,
Dispa_code,Capacity_2015,Capacity_2016)
select country,"HRD",null,null from countries_temp;
insert ignore into entsoe_station_capacities_agg_f (country,
Dispa_code,Capacity_2015,Capacity_2016)
```

```
select country,"OIL",null,null from countries_temp;

drop table if exists capacity_aggregated_fuel2;
create table capacity_aggregated_fuel2
SELECT
entsoe_station_capacities_agg_f.country,
entsoe_station_capacities_agg_f.Dispa_code,
sum(PPDB_wepp_capacity) as PPDB_wepp_capacity_total,
sum(PPDB_wepp_capacity_2015) as PPDB_wepp_capacity_2015,
sum(PPDB_wepp_capacity_2016) as PPDB_wepp_capacity_2016,
sum(PPDB_powervision_capacity) as PPDB_powervision_capacity_total,
sum(PPDB_powervision_capacity_2015) as PPDB_powervision_capacity_2015,
sum(PPDB_powervision_capacity_2016) as PPDB_powervision_capacity_2016,
entsoe_station_capacities_agg_f.Capacity_2015,
entsoe_station_capacities_agg_f.Capacity_2016
from entsoe_station_capacities_agg_f
left join ppdb.capacity_aggregated_fuel
on ppdb.capacity_aggregated_fuel.country
regexp(entsoe_station_capacities_agg_f.country)
and ppdb.capacity_aggregated_fuel.fuel_code =
entsoe_station_capacities_agg_f.Dispa_code
group by entsoe_station_capacities_agg_f.country,
entsoe_station_capacities_agg_f.Dispa_code
;

select * from capacity_aggregated_fuel2;
```

11.5 Generic plant class characteristics addition to output tables

The following sql code provides an example of using the lookutbl table to assign generic class characteristics to power plants in the jrc_from_wepp output table.

```
select
name_code,
wepp_plant,
fuel_code,
technology_code,
wepp_capacity,
jrc_from_wepp.cogen,
jrc_from_wepp.stype,
jrc_from_wepp.country
year,
lookuptbl.id,
lookuptbl.best_efficiency_of_power_generation,
lookuptbl.Best_Ramping_up_rate,
lookuptbl.Best_startup_costs_for_warm_start,
lookuptbl.Best_Variable_OM_costs
from `jrc_from_wepp`
left join
`lookuptbl` on
jrc_from_wepp.`technology_code` like concat("%",lookuptbl.`tech_code`) and
jrc_from_wepp.`fuel_code` regexp(lookuptbl.`fuel`) and
jrc_from_wepp.`STYPE` like concat("%",lookuptbl.`Stype`) and
jrc_from_wepp.cogen like lookuptbl.cogen and
`jrc_from_wepp`.`year` <= lookuptbl.`year_to` and
`jrc_from_wepp`.`year` > lookuptbl.`year_from` and
 jrc_from_wepp.wepp_capacity <= lookuptbl.size_to and
```

```
 jrc_from_wepp.wepp_capacity > lookuptbl.size_from;
```

11.6 Joining the jrc_from_[db] tables

With the following code the jrc_joined table is created and populated.

```
drop table if exists jrc_joined;
CREATE TABLE `jrc_joined` (
  `id` mediumint not null auto_increment,
  `jrc_code` varchar(102) DEFAULT NULL,
  `name_code` varchar(75) DEFAULT NULL,
  `powervision_eic_p` varchar(20) DEFAULT NULL,
  `wepp_eic_p` varchar(20) NOT NULL,
  `powervision_eic_codes` text,
  `wepp_eic_codes` text,
  `powervision_capacity` double(17,0) DEFAULT NULL,
  `wepp_capacity` double(17,0) DEFAULT NULL,
  `powervision_entry` date DEFAULT NULL,
  `wepp_entry` int(11) DEFAULT NULL,
  `powervision_tech` varchar(11) DEFAULT NULL,
  `wepp_tech` varchar(11) DEFAULT NULL,
  `powervision_fuelcode` varchar(11) DEFAULT NULL,
  `wepp_fuelcode` varchar(11) DEFAULT NULL,
  `wepp_plant` varchar(40) DEFAULT NULL,
  `powervision_ret` date DEFAULT NULL,
  `wepp_ret` int(11) DEFAULT NULL,
  `powervision_status` varchar(10) DEFAULT NULL,
  `wepp_status` varchar(10) DEFAULT NULL,
  `stype` varchar(10) DEFAULT NULL,
  `powervision_cogen` varchar(1) DEFAULT NULL,
  `wepp_cogen` text,
  `country` varchar(60) DEFAULT NULL,
  `country_code` varchar(2) DEFAULT NULL,
  `latitude` double DEFAULT NULL,
  `longitude` double DEFAULT NULL,
  primary key (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


insert into jrc_joined(
  `jrc_code`,
  `name_code`,
  `powervision_eic_p`,
  `wepp_eic_p`,
  `powervision_eic_codes`,
  `wepp_eic_codes`,
  `powervision_capacity`,
  `wepp_capacity`,
  `powervision_entry`,
  `wepp_entry`,
  `powervision_tech`,
  `wepp_tech`,
  `powervision_fuelcode`,
  `wepp_fuelcode`,
  `wepp_plant`,
  `powervision_ret`,
  `wepp_ret`,
  `powervision_status`,
  `wepp_status`,
```

```
  `stype`,
  `powervision_cogen`,
  `wepp_cogen`,
  `country`,
  `country_code`,
  `latitude`,
  `longitude`
)
select
PPDB.jrc_from_powervision.jrc_code,
PPDB.jrc_from_powervision.name_code,
PPDB.jrc_from_powervision.eic_p as powervision_eic_p,
PPDB.jrc_from_wepp.eic_p as wepp_eic_p,
PPDB.jrc_from_powervision.eic_codes as powervision_eic_codes,
PPDB.jrc_from_wepp.eic_codes as wepp_eic_codes,
PPDB.jrc_from_powervision.powervision_capacity,
PPDB.jrc_from_wepp.wepp_capacity,
PPDB.jrc_from_powervision.Unit_Oper_Online_Date as powervision_entry,
PPDB.jrc_from_wepp.year as wepp_entry,
PPDB.jrc_from_powervision.technology_code as powervision_tech,
PPDB.jrc_from_wepp.technology_code as wepp_tech,
PPDB.jrc_from_powervision.fuel_code as powervision_fuelcode,
PPDB.jrc_from_wepp.fuel_code as wepp_fuelcode,
PPDB.jrc_from_wepp.wepp_plant,
PPDB.jrc_from_powervision.retirement as powervision_ret,
PPDB.jrc_from_wepp.RETIRE as wepp_ret,
PPDB.jrc_from_powervision.status as powervision_status,
PPDB.jrc_from_wepp.STATUS as wepp_status,
PPDB.jrc_from_wepp.STYPE as stype,
PPDB.jrc_from_powervision.cogen as powervision_cogen,
PPDB.jrc_from_wepp.cogen as wepp_cogen,
PPDB.jrc_from_powervision.country as country,
PPDB.jrc_from_powervision.country_code as country_code,
PPDB.jrc_from_powervision.latitude,
PPDB.jrc_from_powervision.longitude
FROM PPDB.jrc_from_powervision
inner join PPDB.jrc_from_wepp using (name_code)
#where jrc_from_powervision.eic_p =""
limit 5000 ;

update jrc_joined set powervision_eic_p=wepp_eic_p where powervision_eic_p
=""  and wepp_eic_p<>"";
update jrc_joined set wepp_eic_p=powervision_eic_p where wepp_eic_p =""
and powervision_eic_p<>"";

set @num := (select count(*) from jrc_joined);
set @ps = concat("alter table jrc_joined auto_increment = ",@num+1);
prepare stmt from @ps;
execute stmt;

#alter table jrc_joined auto_increment = @num+1;

insert into jrc_joined(
  `jrc_code`,
  `name_code`,
  `powervision_eic_p`,
  `wepp_eic_p`,
  `powervision_eic_codes`,
  `wepp_eic_codes`,
  `powervision_capacity`,
  `wepp_capacity`,
```

```
  `powervision_entry`,
  `wepp_entry`,
  `powervision_tech`,
  `wepp_tech`,
  `powervision_fuelcode`,
  `wepp_fuelcode`,
  `wepp_plant`,
  `powervision_ret`,
  `wepp_ret`,
  `powervision_status`,
  `wepp_status`,
  `stype`,
  `powervision_cogen`,
  `wepp_cogen`,
  `country`,
  `country_code`,
  `latitude`,
  `longitude`
)
select
PPDB.jrc_from_powervision.jrc_code,
PPDB.jrc_from_powervision.name_code,
PPDB.jrc_from_powervision.eic_p as powervision_eic_p,
"" as wepp_eic_p,
PPDB.jrc_from_powervision.eic_codes as powervision_eic_codes,
null as wepp_eic_codes,
PPDB.jrc_from_powervision.powervision_capacity,
null as wepp_capacity,
PPDB.jrc_from_powervision.Unit_Oper_Online_Date as powervision_entry,
null as wepp_entry,
PPDB.jrc_from_powervision.technology_code as powervision_tech,
null as wepp_tech,
PPDB.jrc_from_powervision.fuel_code as powervision_fuelcode,
null as wepp_fuelcode,
null as wepp_plant,
PPDB.jrc_from_powervision.retirement as powervision_ret,
null as wepp_ret,
PPDB.jrc_from_powervision.status as powervision_status,
null as wepp_status,
null as stype,
PPDB.jrc_from_powervision.cogen as powervision_cogen,
null as  wepp_cogen,
PPDB.jrc_from_powervision.country as country,
PPDB.jrc_from_powervision.country_code as country_code,
PPDB.jrc_from_powervision.latitude,
PPDB.jrc_from_powervision.longitude
from jrc_from_powervision where
name_code not in (select name_code from jrc_joined)
#and jrc_from_powervision.eic_p <>""
;

set @num := (select count(*) from jrc_joined);
execute stmt;

insert into jrc_joined(
  `jrc_code`,
  `name_code`,
  `powervision_eic_p`,
  `wepp_eic_p`,
  `powervision_eic_codes`,
  `wepp_eic_codes`,
```

```
  `powervision_capacity`,
  `wepp_capacity`,
  `powervision_entry`,
  `wepp_entry`,
  `powervision_tech`,
  `wepp_tech`,
  `powervision_fuelcode`,
  `wepp_fuelcode`,
  `wepp_plant`,
  `powervision_ret`,
  `wepp_ret`,
  `powervision_status`,
  `wepp_status`,
  `stype`,
  `powervision_cogen`,
  `wepp_cogen`,
  `country`,
  `country_code`,
  `latitude`,
  `longitude`
)
select
null as jrc_code,
PPDB.jrc_from_wepp.name_code,
"" as powervision_eic_p,
PPDB.jrc_from_wepp.eic_p as wepp_eic_p,
null as powervision_eic_codes,
PPDB.jrc_from_wepp.eic_codes as wepp_eic_codes,
null as powervision_capacity,
PPDB.jrc_from_wepp.wepp_capacity,
null as powervision_entry,
PPDB.jrc_from_wepp.year as wepp_entry,
null as powervision_tech,
PPDB.jrc_from_wepp.technology_code as wepp_tech,
null as powervision_fuelcode,
PPDB.jrc_from_wepp.fuel_code as wepp_fuelcode,
PPDB.jrc_from_wepp.wepp_plant,
null as powervision_ret,
PPDB.jrc_from_wepp.RETIRE as wepp_ret,
null as powervision_status,
PPDB.jrc_from_wepp.STATUS as wepp_status,
PPDB.jrc_from_wepp.STYPE as stype,
null as powervision_cogen,
PPDB.jrc_from_wepp.cogen as wepp_cogen,
PPDB.jrc_from_wepp.country as country,
PPDB.jrc_from_wepp.country_code as country_code,
null as latitude,
null as longitude
from jrc_from_wepp where name_code not in (select name_code from
jrc_joined)
#and jrc_from_wepp.eic_p <>""
;


#check for duplicates
drop  table if exists duplicates_joined;
create temporary table duplicates_joined
select name_code,
count(name_code) as occ
from jrc_joined
group by name_code ;
```

```
delete from duplicates_joined where occ=1;
select * from duplicates_joined;


drop table if exists jrc_joined_pk;
create table jrc_joined_pk (
`idn` int(4)  zerofill not null auto_increment,
`id` mediumint not null,
`jrc_code` varchar(20),
`country_code` varchar(2),
 primary key (`country_code`, `idn`)
)ENGINE MyISAM;

DELETE FROM jrc_joined_pk;
insert into jrc_joined_pk (
`id`,
`jrc_code`,
`country_code`)
select id,
case when powervision_fuelcode is not null then
concat(country_code,"_",powervision_fuelcode,"_",powervision_tech) else
concat(country_code,"_",wepp_fuelcode,"_",wepp_tech) end
,country_code from jrc_joined;

UPDATE jrc_joined_pk
SET
    jrc_code = CONCAT(jrc_code, '_', idn);

UPDATE jrc_joined inner join jrc_joined_pk using(id)
set  jrc_joined.jrc_code = jrc_joined_pk.jrc_code;

#update cogen flags
update jrc_joined set powervision_cogen = "N" where powervision_cogen is
null;
update jrc_joined set stype = "N/A" where stype is null;
```

Europe Direct is a service to help you find answers to your questions about the European Union.

Freephone number (*)**:**

00 800 6 7 8 9 10 11

(*) The information given is free, as are most calls (though some operators, phone boxes or hotels may charge you).

More information on the European Union is available on the internet (http://europa.eu).

HOW TO OBTAIN EU PUBLICATIONS

Free publications:

- one copy:
  via EU Bookshop (http://bookshop.europa.eu);
- more than one copy or posters/maps:
  from the European Union's representations (http://ec.europa.eu/represent_en.htm);
  from the delegations in non-EU countries (http://eeas.europa.eu/delegations/index_en.htm);
  by contacting the Europe Direct service (http://europa.eu/europedirect/index_en.htm) or
  calling 00 800 6 7 8 9 10 11 (freephone number from anywhere in the EU) (*).

  (*) The information given is free, as are most calls (though some operators, phone boxes or hotels may charge you).

Priced publications:

- via EU Bookshop (http://bookshop.europa.eu).

KJ-NA-28549-EN-N

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

Joint Research Centre

EU Science Hub

Publications Office

doi:10.2760/329310

ISBN 978-92-79-67687-1