

Bachelor Thesis

Open Data in the domain of Cultural Heritage

Florian Winkler

Date of Birth: 28.01.1982

Student ID: 0150820

Subject Area: Information Business

Studienkennzahl: J033561

Supervisor: Axel Polleres

Date of Submission: 12. Jänner 2015

*Institute of Information Business, Department of Information Systems and
Operations, Vienna University of Economics and Business, Welthandelsplatz
1, 1020 Vienna, Austria*



Contents

1	Introduction	6
2	Background	7
2.1	Definition of Open Data	7
2.2	History of Open Data	8
2.3	Benefits and barriers	8
2.4	Government and private data	10
2.5	Open Government Implementation Model	12
2.6	Publishing data on the Web	17
2.7	Linked Open Data	18
2.8	Measuring published Open Data	19
3	A practical use-case for publishing Open Data	20
3.1	Use-case: data of a museum's art collection	20
3.2	Process overview	21
3.3	Publishing data	22
4	Evaluation	45
5	Related Works	48
5.1	Amsterdam Museum	48
5.2	Smithsonian American Art Museum (SAAM)	49
6	Future Works and Conclusion	50
	References	52
	Appendix A - Email conversation	56
	Appendix B - Excel file with field mappings	58
	Appendix C - Python scripts for data transformation	60
	Appendix D - Metadata file for CSV	70
	Appendix E - Python script: CKAN API access	73
	Appendix F - Resulting files: CSV, JSON, JSON-LD	75

List of Figures

1	KDZ implementation model overall view. Source: [Krabina et al., 2012, p.44]	13
2	Overview of the publishing process.	23
3	Entity relationship diagram of the used tables from the mumok database.	29
4	JSON notation of an object. Source: [Ecma International, 2001].	32
5	JSON notation of an array. Source: [Ecma International, 2001].	32
6	RDF graph of textual representation example. Source: [Schreiber and Raimond, 2014].	34
7	Email conversation with Mag. Jörg Wolfert	56
8	Email conversation with Dr. Susanne Neuburger	57

Abstract

The trend of publishing collected or self-created information in form of human- and machine-readable Open Data is supported by recent legislations from the European Union and other governments in the world. This document gathers existing standards and best practices on this topic and presents them in the process of publishing data from start to finish. The cultural point of view gets added by a practical example of the data of a museum's collection getting extracted, transformed and made available on an Open Data portal. The documentation of the process serves as a handbook for other organizations being eager to follow this trend. The published data gets analysed with the help of three different evaluation concepts. Finally additional steps for improvement of the resulting Open Data are proposed.

1 Introduction

This document describes the different aspects of publishing documents and information to the Web in form of Open Data. The approach of increasing transparency and making data available for free to everyone has started many years ago and is carefully adopted by the European Union and its governments. In June 2015 the Austrian parliament decided on the incorporation of the European Union's legislation on reuse of public sector information into national Austrian law [Open Knowledge Foundation Austria, 2015]. This grants the citizens the right to use all documents financed by their own tax money. On the Austrian central data platform data.gv.at there is already a number of documents and data sets available, and also more private organizations finally got the possibility of releasing their data on a common platform, namely the Open Data Portal Austria.

The main idea is to make as many data sets as possible available in formats that can be used by humans and machines alike. The raw data by itself is often of no real value per se but when combined with other data sets valuable applications can be developed. The present thesis starts with the definition and history of Open Data beginning in the area of scientific research and then spreading the idea to governments all over the world. To maintain the link to the Austrian situation, the Open Government Implementation Model is explained, which defines the theoretical process for governmental organizations to open up their data. Then the general concept of publishing data on the web is outlined and the instruments of measuring its success are presented. The process of publishing data from start to finish is then documented based on an actual example. Every step is explained from a general viewpoint as well as with concrete measures, that have been undertaken on the way. The actual example is the publication of the art collection of the Museum of Modern Art in Vienna, Austria (mumok) with their corresponding artists and exhibitions. The process is divided into twelve steps starting with the application for permissions from the museum followed by a discussion on privacy concerns and the hands-on experience with extracting the data from the database and transforming it to the resulting file formats. The data will then be enriched with metadata and published to an existing Open Data portal. Examples of code for transformation and the automatic update process can be found in the appendix. The mechanics for providing a system for different versions, which may change very day, and a feedback system for integrating the users of the data are not included in this document.

2 Background

2.1 Definition of Open Data

The term *open* in Open Data is ultimately derived from the Definition of Free Cultural Works that defines "works or expressions which can be freely studied, applied, copied and/or modified, by anyone, for any purpose" [Möller, 2006]. This definition covers all sorts of works, created by humanity, which may be of use to any other person and should therefore be freely and easily available to everybody. The final goal here is to enrich human culture. Another source describing the meaning of *open* in this context is the Open Source Definition, initially developed for the creation and usage of source code for computer programs. This definition states, that open source works should be freely distributable and bound to a license, that does not discriminate in any way [Opensource.org, 2004]. The actual definition for the term *open* is explained in the Open Definition, which uses three categories [Pollock, 2014]:

Open License. An open license has to be used to publish the work. This license must not be overridden by any terms of use or other regulatory documents that are attached to the data. The different available licenses will be further discussed in section 3.3.2.

Access. The work has to be freely accessible to everyone. It should be available free of charge, preferably over the internet. For other ways of publication the cost for a one-time reproduction may be applicable. All necessary information in order to fulfill the license terms has to be included. Different portals for publishing data will be presented in section 2.4.1.

Open Format. The data has to be in an open format which does not impose any technological impedance, like proprietary file formats. It shall also be machine-readable for automatic processing. The most popular formats available are listed and described in section 3.3.7.

In connection with the term *data*, the meaning of *open* is applied to all sorts of (electronic) data. This combination then results in Open Data, which may be consumed by everyone without any monetary, technological or re-distributional obstacles. Governments are publishing Open Data for some years, providing the public with all sorts of numbers, coordinates and other contents.

According to the Open Data Handbook [Dietrich et al., 2014] the biggest advantage of the before mentioned three categories for different Open Data sets, that come from a large number of publishers, is interoperability. It allows everybody to combine data sets, that have never before been put into a relation, in order to generate new and richer data out of them. An

already existing example for this case is an application called moovel (<https://www.moovel.com>) combining data of public transportation with data from car-sharing providers as well as taxi companies. This program presents a full portfolio of possibilities to the user to reach a given destination.

2.2 History of Open Data

In regard to [Chignard, 2013] the term Open Data made its first appearance in an article from an American scientific agency in 1995, in which scientists propose "a complete and open exchange of scientific information between different countries" [Chignard, 2013]. He also says that other people supported the matter of opening up scientific data to the public for many years. Scientists around the world should share their research data and findings completely to increase global knowledge. One of the first projects in the scientific world making use of the open approach was the Human Genome Project that involved several different institutes. The research project started in 1990 with the goal to create a complete map of the human DNA, which was finished by 2005 [National Human Genome Research Institute, 1999]. All of the researched data has been made available through the internet with the National Human Genome Research Institute's principle of "ensuring that sequencing data from the human genome is available to the world's scientists rapidly, freely and without restriction" [National Human Genome Research Institute, 2012]. Chignard further reports [Chignard, 2013] that in December of the year 2007 a group of internet activists got together in Sebastopol, California, to collect different thoughts and practices in order to form a concept of public data. Eight principles were built, describing the rules and actions for governments to successfully publish data [Tauberer, 2014]. Later on the Sunlight Foundation, the main sponsor of the event in Sebastopol, restructured those principles and added two more [Sunlight Foundation, 2010, p.1]. Those Ten Principles for Opening Up Government Information are used all over the world by different countries to define their own ways of publishing Open Data. The principles as well as their usage in the Austrian Open Government Implementation Model, will be discussed in section 2.5.1. Nowadays the majority of published Open Data derives from governmental information but there are also private companies and other institutions from the public sector catching up with this trend.

2.3 Benefits and barriers

As one of the first steps for starting the process of publishing a government's or private company's data, the benefits and drawbacks for this intent have

to be discovered and estimated. There are ideological as well as economical factors, which need to be considered. Marijn Janssen and Anneke Zuiderwijk from the Delft University of Technology in Netherlands together with Yannis Charalabidis from the University of the Aegean in Greece put together a list of benefits and barriers for this case. For this list they conducted a series of interviews and a workshop with several people from governments and private companies, that have already been involved with this topic [Janssen et al., 2012, p.259]. The discovered benefits can be separated into three categories [Janssen et al., 2012, p.261]:

1. *Political and social*: This category includes an increase in transparency, more trust in governments and improvements for public services.
2. *Economic*: Published Open Data in its pure form has no market price and only generates value when it is used by others or combined with different data sets. This fact concludes that the return on investment for publishing Open Data cannot be calculated upfront, which hinders economic decisions [Janssen et al., 2012, p.261]. Further gains in the economic area are stimulation of innovation and use of the wisdom of crowds.
3. *Operational and technical*: Benefits with operational and technical aspects include the external quality check of data, easier access to data as well as the ability to merge, integrate and mesh public and private data [Janssen et al., 2012, p.261].

The identified barriers concentrate on the following six different categories:

1. *Institutional*: There is currently no uniform policy to publish data and also the resources for doing so are mostly not at hand in smaller companies. To improve already existing data, a feedback system should be planned and included during the publication of data [Janssen et al., 2012, p.259] which also presents a difficulty for the publishers. This establishment of a feedback system is also mentioned in the Austrian Open Government Implementation Model [Krabina et al., 2012] and will be looked upon in section 2.5.1.
2. *Task complexity*: This barrier accounts the fact that data formats may be too complex to be used by everybody, particularly when the description of the meaning of the data is missing. This meaning can be added through the use of metadata as described in section 3.3.9.

3. *Use and participation*: Using Open Data can be obstructed for some people when a fee has to be paid to access the information. In order to integrate the users into the process, an organization has to provide the necessary resources, which comes in the form of an additional employee or a dedicated department. Such resources will probably not be available within public organizations.
4. *Legislation*: Takes care of issues where the privacy of data has to be paid attention to and the decision for an appropriate license has to be considered.
5. *Information quality*: This stands for the completeness and correctness of data, which should be validated before every release and also involves additional costs.
6. *Technical*: The technical barriers consist of the lack of standards for formats and metadata along with the missing possibility of publishing the data on a central portal where it can be found by everybody.

Those benefits and barriers should be taken into account when planning the publication of Open Data. The reason as why to publish data at all may differ a lot, depending on the type of organization e.g. whether it's a governmental or a private one.

2.4 Government and private data

There seems to be a different motivation for publishing data coming from a government organization as opposed to being released by a private company. The data from the former is already owned by the public, as the production of this data is financed with tax money. In the latter case, produced data is a very valuable resource, which can distinguish one's success against the competitors. As the whole Open Data movement has started in the scientific world (see section 2.2), governments were the first to make a step to more transparency. In 2003 the "EU directive on the re-use of public sector information [...] got implemented into [Austrian] national law" [Höchtel et al., 2014, p.1]. At that time there haven't been any steps made or even thought about on how to deal with that new regulation, although Austria has always been on track with its eGovernment initiative [Höchtel et al., 2014, p.1], providing a lot of different services to the citizens on the Web to conduct matters with public authority. In the year 2009 the United States administration under Barack Obama declared their future tendency to more openness and a transparency strategy [Huijboom and Broek, 2011, p.1]. So in the years

from 2009 to 2011 several countries started with their own Open Data initiatives. In Germany the Dresden Agreement was launched in December 2010 in order to create the platform Govdata (<http://www.govdata.de>) which will serve Open Data to the public from many sources in one place [Klessmann et al., 2012, p.1]. In Austria the Cooperation Open Government Data, short OGD, was launched in 2011 by the cities Vienna, Linz, Graz and Salzburg. The OGD used the Ten Principles for Opening Up Government Information [Sunlight Foundation, 2010] and started to define a common portal for Open Data, a set of metadata and the used licenses for the published data [Höchtel et al., 2014, p.1]. In July 2015 the Austrian government decided upon an amendment to the Informationsweitergabegesetz, which regulates the permission for sharing and using documents, that are owned and produced by public governmental organizations. The changes include that every citizen has the right to use those documents, be it for non-commercial or commercial use. This also accounts for documents from libraries, museums and archives. The law also sticks to the Ten Principles (see section 2.5.1) in regard to the used file format, which has to be machine-readable and open as defined by the Open Definition (see section 2.1).

The creation of new laws and the need for more transparency in the democratic world made many governments start their own initiative for Open Data. In the sector of private companies, they first started using the data provided by the government to create profit, but soon realized that opening up their own information can help to increase transparency and openness towards the stakeholders [Herzberg, 2014]. Benjamin Herzberg, the Program Lead for Private Sector Engagement for Good Governance (PSGG) at the World Bank Institute, gives some examples of private sector companies that are already engaged in the Open Data business, like for instance a mining company in Burkina Faso which publishes its contracts with the government to allow the local citizens to validate the company's actions in their area [Herzberg, 2014]. Hardly any central private Open Data portals can be seen today, whereas nearly every country and of course the European Union promote a central portal for public Open Data from the government or other public agencies.

2.4.1 Open data portals in Austria and in the EU

In Austria there is the central portal managed by the government, namely data.gv.at, where public agencies offer different sets of Open Data in various data formats. Currently, there is a total number of 1.897 data sets available (<https://www.data.gv.at/suche/>, accessed: 2016-01-09). The Open Data Portal Österreich is available for the private sector. It was founded in spring

2014 by Wikimedia Österreich, Open Knowledge foundation Österreich and the Cooperation OGD Österreich. This portal provides the possibility of publishing data on a central platform for NGOs, private companies and even private individuals under an open license. The European Union has a central portal as well, the European Union Open Data Portal with a total number of 7.894 data sets (<http://open-data.europa.eu>, accessed: 2016-01-09). The Open Knowledge Foundation has created a global Open Data Index which currently compares the state of open government data in 122 different countries (<http://index.okfn.org/>, accessed: 2016-01-09).

2.5 Open Government Implementation Model

Several strategies exist for governments to start publishing data to a central portal. The one that is most applicable in Austria is the Open Government Implementation model by the KDZ - Centre for Public Administration Research in Vienna. According to [Krabina et al., 2012, p.5] there are four stages needed for a successful implementation that build upon each predecessor's success: Increasing data transparency, Improving Open Participation, Enhancing Open Collaboration and Realizing Ubiquitous Engagement. The overall view on this model can be seen in figure 1. Currently the first stage is documented in detail, whereas the remaining three stages are roughly drafted and shall be completed in the near future.

2.5.1 Increasing data transparency

The main focus in this first stage is to pick out the data that may be most valuable for the users of Open Data and to assess and increase the quality of the already existing data upfront [Krabina et al., 2012, p.9]. The first step herein is to establish an Open Government Competence Centre in the organization. It should consist of members of different departments and therefore exist as "a virtual organizational unit" [Krabina et al., 2012, p.9], that is not anchored in the traditional hierarchy. Employees as well as the public take part in the publication of Open Data. This accounts for the two main roles of the competence center. It is the contact for all employees, organizes internal training and monitors the relevant data. On the other side it can be contacted by consumers of Open Data to get help on data structures and how to handle them. Furthermore it receives requests from the public for additional data to be released. Other duties of the competence center include the discovery of potential data through interviews or discussions with different departments. The factors for measuring the success like number of published data sets, frequency of data updates or public awareness, have to

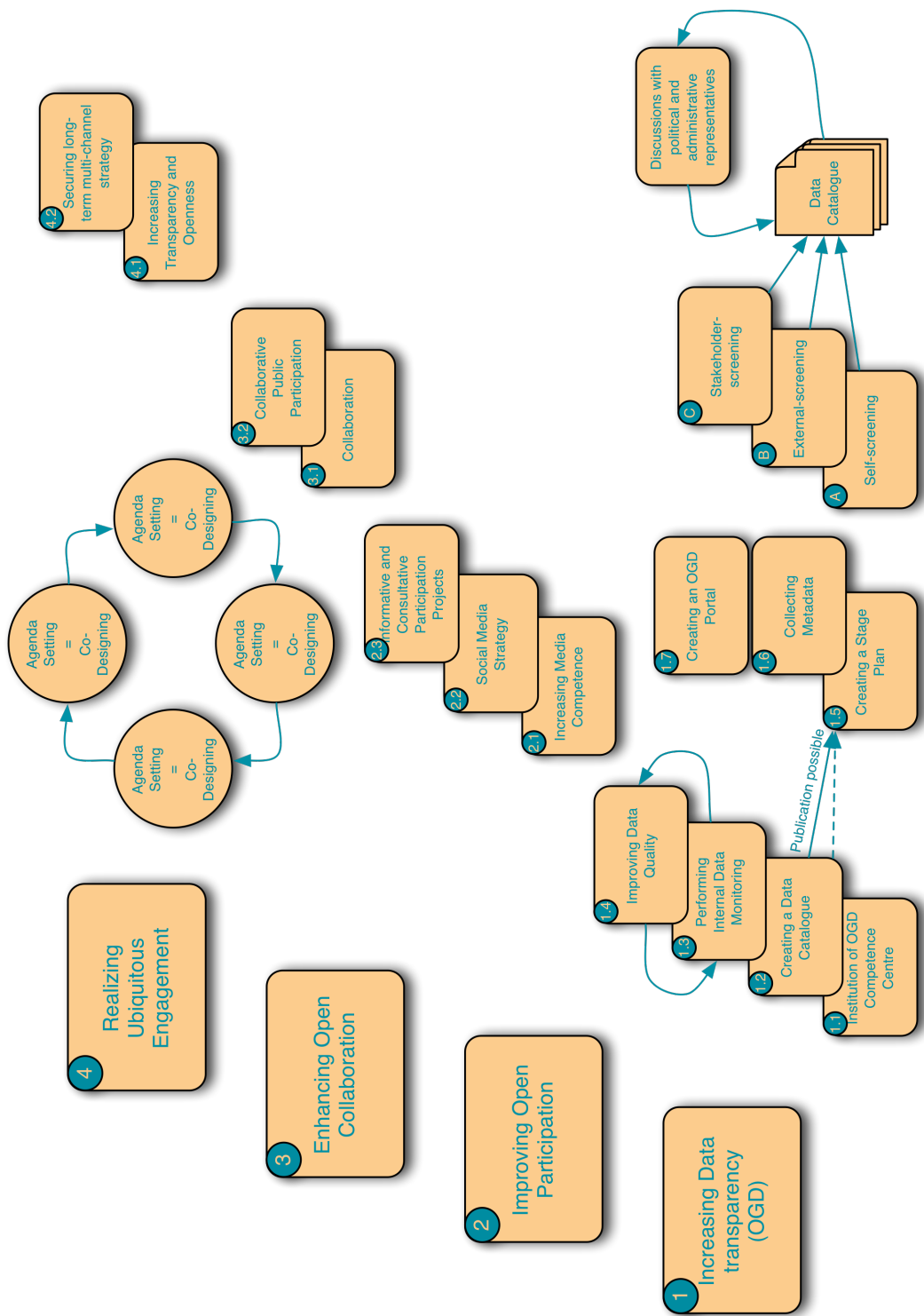


Figure 1: KZD implementation model overall view. Source: [Krabina et al., 2012, p.44]

be defined. Finally the compliance with Open Government Data principles has to be monitored.

Following the establishment of the competence center, the creation of the organization's data catalog can begin. In order to identify the most important and valuable data sets, three screening methods should be used:

- *Self-screening* takes into account all the data that has already been released in the past, because it had been necessary to provide them to the public, but does not comply with the requirements of Open Data. This could for example affect PDF documents that have to be converted to CSV format. More on Open Data formats like CSV will be discussed in section 3.3.7.
- The goal of *external-screening* is to get an overview of already published data by others that may give an idea on what kind of data is important and how the finished documents look like.
- Finally, the *stakeholder-screening* should produce a collection of relevant parties, that may benefit from Open Data, which leads to a possible order of importance for the upcoming releases.

With the result of these three screening methods a possible catalog can be populated with ideas. Those will have to be further evaluated with the help of a set of criteria like legal restrictions, data value or technical availability and should be figured out for every set of data. This method of internal data monitoring leads to a more concrete data catalog. As already mentioned in section 2.2, the Ten Principles for Opening Up Government Information are also being used in this implementation model. The collection of those ten principles is the augmented version of the eight principles that have been defined by internet activists in Sebastopol 2007. The Sunlight Foundation, a non profit company with the goal to ensure more democratic participation through the use of technology, presents the complete list of the ten principles as follows [Sunlight Foundation, 2010, p.1]:

Completeness. A released data set should be as complete as possible, containing all data regarding the subject as well as metadata and formulas so that the meaning, origin and composition of the data can be understood by everyone.

Primacy. Data shall always be released by the originating or creating entity so that it comes from the primary source.

Timeliness. Real time publication of information would be the optimum. Otherwise produced data should be made available as quickly as possible.

Ease of Physical and Electronic Access. In order to access the data, no special procedures like forms or applications have to be necessary. The location of access should be reachable from everywhere as opposed to be physically located solely in an office in a specific city. For the electronic access no proprietary software like Adobe Flash or any other complicated programs have to be necessary.

Machine readability. The documents should be available in a format which is readable and interpretable by a machine. Often a PDF is used for publication, which may be nice to read for a person but the content can not be further processed by a computer.

Non-discrimination. This concentrates on who may access the data and how this can be done. No membership or identification measures may be used. The Sunlight Foundation summarizes this principle as follows: "any person can access the data at any time without having to identify him/herself or provide any justification for doing so" [Sunlight Foundation, 2010, p.2].

Use of Commonly Owned Standards. The software to produce and to use the data has to be open, therefore not being owned by a company. No license may be required to use the software. Data may be published in non open formats as long as it is available as another copy in the same quality for free software as well.

Licensing. Open data should be clearly marked as governmental and available without any impediments.

Permanence. As soon as a piece of data has been published, it should be available in this form forever. A newer version should not replace the older one but move it to an archive where it will be accessible forever.

Usage costs. No usage costs should ever be enforced for the access to Open Data.

The two principles added by the Sunlight Foundation are Permanence and Usage costs. The implementation model also includes some file formats which may be used for Open Data. The types used in this paper will be described in section 3.3.7. For the quality rating of the data the five star system is used which will be explained in section 2.8. As the first principle (completeness) states, metadata should always be included. In Austria there is a standard published by the working group on metadata [Habernig et al., 2015] which will be explained later on in section 3.3.9. As a next step an Open Government Data Portal has to be either set up, if it should be administrated within the organization, or chosen from already existing portals in Austria or the European Union. During the first stage the integration into standard processes within the organization has to be planned [Krabina et al., 2012, p.26]. IT-systems can be altered to automatically share updates to Open Data. If a new project is started in the organization, the data man-

agement strategies should be considered and implemented as well. In order to further improve published data, it is necessary to establish some kind of feedback possibility for the users. These steps conclude the first stage. It is recommended to repeat those steps regularly to improve on quality and number of published datasets.

2.5.2 Improving Open Participation

This second stage aims at establishing the rules and strategies for the organization to actively get in contact with the public, promote the released data and encourage them to take part in further development. According to [Krabina et al., 2012, p.30] it is important to make politicians and public administrators aware of the released data, its usage and development, as well as possibilities of participation. This should be achieved in the first step of this stage, called increasing media competence. Several reasons are listed why employees should be given access to social media [Krabina et al., 2012, p.30-31]. For this to work in the interest of the current organization training should be provided to employees and a social media guideline should be defined. There are already some public guidelines from the Austrian e-government platform as well as the City of Vienna. In the second step, creating and implementing social media strategy, a set of rules should be developed that determines the target groups, social networks and type of communications, that will be used. At the last step in this stage, called informative and consultative public involvement, a plan should be drawn up which includes all stakeholders and how they can be involved in the project. Their involvement can already start during stage one and consist of participation in community meetings, publishing and promoting the applications, that have been developed with released data, or the discussion of released data on social media. Two examples listed in [Krabina et al., 2012, p.34-35] are the campaign of identifying the most annoying traffic light in Vienna or the collaborative on-line creation of a mission statement for the redesign of Vienna's Schwedenplatz in 2012.

2.5.3 Enhancing Open Collaboration

In stage three the further integration of the public is targeted. Achievements and effects that are produced together with people outside of the organization, namely through the public, are combined under the term collaboration. In [Krabina et al., 2012, p.37] some examples are mentioned like the Bürger-Solarkraftwerk in Vienna, where people are financing a solar power plant together with fixed interest shares, or the Cooperation Open Government Data Austria that wants to unite all players of open government data on a

single platform. The governments should also encourage private companies to publish their data, which may be a direct use for citizens. This includes, for instance, locations of mailboxes and post offices, opening hours of pharmacies or ATMs [Krabina et al., 2012, p.37]. The applications resulting from Open Data are a valuable product and therefore their developers should be kept in contact and provided with updated data in order to ensure the viability of the applications. The second step in this stage, collaborative public participation, tries to outsource political decisions to the public, letting them decide and elaborate on future outcomes.

2.5.4 Realizing Ubiquitous Engagement

This stage aims for the engagement of all possible actors and stakeholders in the Open Data initiative, decision making and co-creating of various initiatives. It can be achieved by further increasing transparency and openness [Krabina et al., 2012, p.42] and securing long-term character of multi-channel strategy for ubiquitous engagement [Krabina et al., 2012, p.43], which means the integration into existing processes and new software solutions that can be used universally for all sorts of political demands.

2.6 Publishing data on the Web

There is another set of guidelines which also is a broad collection of considerations, that should be thought about upfront when publishing data on the Web. It is developed and continually updated by the World Wide Web Consortium (W3C). This organization consists of its own full-time staff and a lot of other people and organizations worldwide. Its "mission is to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web" [W3C, 2015b]. Two roles are participating in the exchange of electronic data over the Web, data publishers and data consumers. The publishers are the ones sharing the data, whereas the consumers are the ones that want to discover and use the published data. They also want the data to be up to date, accurate and available at all times [W3C, 2015b]. Most of the times the publishers and the consumers will not be known to each other and both parties have different expectations and understanding of data. When documents are put on the Web there are several ways of presenting as well as accessing them. Therefore a common method of publishing should be established [W3C, 2015b]. A list of Web challenges is presented that should be taken into account. The guidelines offer a large set of best practices covering these challenges. Most of these challenges will be covered in the methodology section 3 as well.

2.7 Linked Open Data

Publishing data as a concept of increasing knowledge and developing new combined values, also leads one step further, namely the creation of linked Open Data. Tim Berners-Lee, the founder of the World-Wide-Web, explains this idea as follows [Berners-Lee, 2006]: when several people and organizations provide data, this data can be vastly enriched by linking to others also published on the Web. This would lead the users to even more information and different data sets. Those links should be established by using Uniform Resource Identifiers, short URI. According to [Berners-Lee, 2006] they should be URIs referencing to HTTP resources so that everybody can look them up. For linked data the World Wide Web Consortium also has a working group which has set up some best practices [Hyland et al., 2014]. A document with data alone does not have to contain much value but can be increased by combining it with other resources [Hyland et al., 2014]. When modelling linked data, the goal is to make the data self-describing, so the origin or type of the value is defined within the same document. This approach is contrary to traditional data stores like relational databases, where data is described only in external documents or diagrams [Hyland et al., 2014]. The most common solution for creating linked data is the Resource Description Framework (RDF). Its structure is represented by a directed labelled graph consisting of triples of a subject, a predicate and an object [Polleres and Steyskal, 2014, p.2]. This construct makes it possible to describe values, for instance: "cat is of type mammal", which is shown below as an RDF graph.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://dbpedia.org/resource/>.
@prefix dbo: <http://dbpedia.org/ontology/>.

:Cat rdf:type dbo:Mammal.
:Dog rdf:type dbo:Mammal.
```

In order to remodel this triple to make use of URIs, which in turn are used as a "single global identification system" [Hyland et al., 2014], a set of already predefined vocabularies may be used. This makes it possible for data sets using the same vocabularies to be clearly defined, comparable and understandable by humans and machines alike. One of these vocabularies is the Dublin Core Metadata Innovation (DCMI), which will later be used in section 3.3.9, or DBpedia, which extracts structured information from Wikipedia articles. In the example of the cat, the word mammal can be written as an URI, referencing the DBpedia's "http://dbpedia.org/ontology/Mammal". When another person releases data with the triple "dog is of type mammal" and

also uses the DBpedia vocabulary, this information can be directly compared with the first example, leading the person to know, that cat is of exactly the same type as the dog. Although this is a very simple example it shows the general idea of linking and referencing data.

2.8 Measuring published Open Data

The most cited rating system for published Open Data is the five star Open Data rating chart by Tim Berners-Lee [Berners-Lee, 2006]. It consists of five steps, building upon each other, starting with a score of a single star:

One star. The data has to be on the Web in any file format, available under an open license.

Two stars. It should be structured data, readable by a machine. This means that data should be text or numbers instead of an image or PDF.

Three stars. No proprietary file format may be used, for example preferring CSV over an Excel file.

Four stars. Open standards like RDF or other linked data concepts from the World Wide Web Consortium should be used to describe the data.

Five stars. Data should be linked to other existing data to provide context.

Another method measuring the quality of Open Data is to evaluate the conformance with the Ten Principles for Opening Up Government Information (see section 2.5.1). The published data can be checked regarding the listed properties, if they get applied and to which extent. A more extensive classification system is provided by the Open Data Institute in London (<https://certificates.theodi.org>). They allow the self assessment of published data and grant certificates as means of evaluation for the quality of the data. There are four levels of available certificates: raw, pilot, standard and expert. The certificate can be put next to the data to give users a first look at the quality and freshness. On the other side, data providers get an insight on how they can improve the data itself, the describing metadata or other factors like user feedback in order to reach a higher level of certification. Currently they have 11.303 certificates in their database with the following levels: 2 expert, 4 standard, 77 pilot, 11.216 raw and 4 with no level specification. From the Vienna University of Economics and Business the Institute of Information Business maintains a platform called Open Data Portal Watch. The platform collects information and data sets from different open data portals and then assesses their quality. Sebastian Neumaier defined a set of quality metrics for this check [Neumaier, 2015]. They consist of retrievability, usage, completeness, accuracy, openness and contactability. As those metrics are applied to the single published sets of data to evaluate

the whole portal, they can also be used just to measure the quality of a single data set.

3 A practical use-case for publishing Open Data

This thesis presents a guideline on how to publish the data of a museum's art collection to an existing Open Data portal and the concrete deployment of these guidelines in a practical example. The different steps and the various possibilities are outlined and the decisions for the practical implementations of the project are explained.

3.1 Use-case: data of a museum's art collection

The museum in this case is the Museum of Modern Art in Vienna, Austria, further called "mumok". The institution was founded in September 1962 with the mission to create and exhibit a collection of modern art for Austria [Museum Moderner Kunst Stiftung Ludwig Wien, 2012]. The building was then situated in Vienna's 3rd district in the Schweizer Garten and moved to its current location in the Museumsquartier in 2001. Peter and Irene Ludwig, a married couple from Aachen, Germany, found interest in the new museum of modern art in Vienna and they lent a number of art works from their private collection for a current exhibition. Later on the Austrian Ludwig Foundation was born with the goal to administer a number of loaned works from the married couple into the permanent collection of the mumok [Museum Moderner Kunst Stiftung Ludwig Wien, 2012]. The legal form of enterprise is a scientific public-law institution as defined in the Austrian Bundesmuseen-Gesetz 2002 (Federal Museums Act). In 2010 a large-scale project was started to begin with digitization of the pieces of art in the museum's collection with a primary goal of creating digital images of the object and entering all sorts of textual information about it in a digital archive. This also led to the introduction of a new database system for the organization of the produced digital material and information. The software is called MuseumPlus and is developed by the German company Zetcom. Every piece of art has to be photographed and measured. Then the data is being keyed in to the database including a textual description, links to artists and exhibitions as well as the corresponding images. The links that get established here all refer to records that exist in the database as well. Every record is then revised and marked as ready to be published. Currently there are approximately 4.600 objects,

that can also be seen on the museum's website in the online collection section (<https://www.mumok.at/en/collection/online-collection>). The remaining pieces are still being worked on and will be published as soon as all information has been recorded and validated.

3.2 Process overview

This section provides an overview of the whole process of publishing data from an existing database to an Open Data portal. It consists of twelve steps that each follow upon the completion of its predecessor:

1. *Get Permission for publication* (section 3.3.1): As the very first step, permission has to be granted from the owner of the data on which parts and in what way they may be published at all.
2. *Acquire license for publishing* (section 3.3.2): The license under which the data will be released has to be discussed and verified in legal terms.
3. *Discover possible privacy concerns* (section 3.3.3): All the pieces of information have to be analysed for personal data, as these should never be available in an Open Data set.
4. *Analyse existing data* (section 3.3.4): The available source of data has to be investigated. The necessary tables have to be chosen and the relationships between them inspected. Afterwards those findings can be visually mapped in a diagram.
5. *Prepare data for export* (section 3.3.5): Using the above mentioned diagram one or more SQL views can be created as a preparation to facilitate the programming effort for exporting the data.
6. *Access data* (section 3.3.6): The work-space for extracting and transforming the information has to be set up, including the installation of necessary software. Using these tools a connection to the source database can be made to access the existing data.
7. *Choose publishing formats* (section 3.3.7): The file formats in which the Open Data will be made available has to be chosen in order to choose the right software for the transformation process.
8. *Transform data* (section 3.3.8): A script or program has to be created for exporting the data from the database server, transforming it and writing it to a number of different output files according to the chosen publishing formats.

9. *Create Metadata* (section 3.3.9): In order to create a description for the published data, several metadata files have to be produced, which explain the content to the users of the data.
10. *Publish to Open Data portal* (section 3.3.10): In this step the files can be made available at the Open Data portal to the public.
11. *Establish additional convenience settings* (section 3.3.11): Additional settings and helper scripts can be developed to fulfil the requirements for accessing Open Data resources via the Web.
12. *Configure automatic updates* (section 3.3.12): Here the process of automating the complete export is explained, which results in a new version of data every day.

A graphical overview of this process can be seen in figure 2 on page 23.

3.3 Publishing data

This section describes the proceedings of publishing data from the museum's collection of art as data sets of Open Data to an Open Data portal. As the author is currently employed with the museum, there already existed some knowledge about the circumstances and the database upfront, that may not be known to an outsider. Some steps will be easier to achieve if such a knowledge exists, but the process will be the same if it is carried out by an insider or an outsider who might need to consult internal experts additionally. The developments in this paper will be shown from the view of an external person, that gets confronted with the task, in order to keep an unbiased look at the matter. This person will further be called the publisher.

3.3.1 Get Permission for publication

At the very beginning the publisher will have to get permission for publication from the owner of the data. Mostly it will not be possible to just use all available data and release them to the public because of legal issues, or in case of private companies, because they do not want to show everything to their competitors. So it has to be clearly defined which information will be opened up and who is responsible for granting permission to the publication of said data.

When taking a closer look at the legal status of the museum (see section 3.1) it becomes obvious that the museum is financed with tax money and therefore everything being produced belongs to the Austrian citizens. This

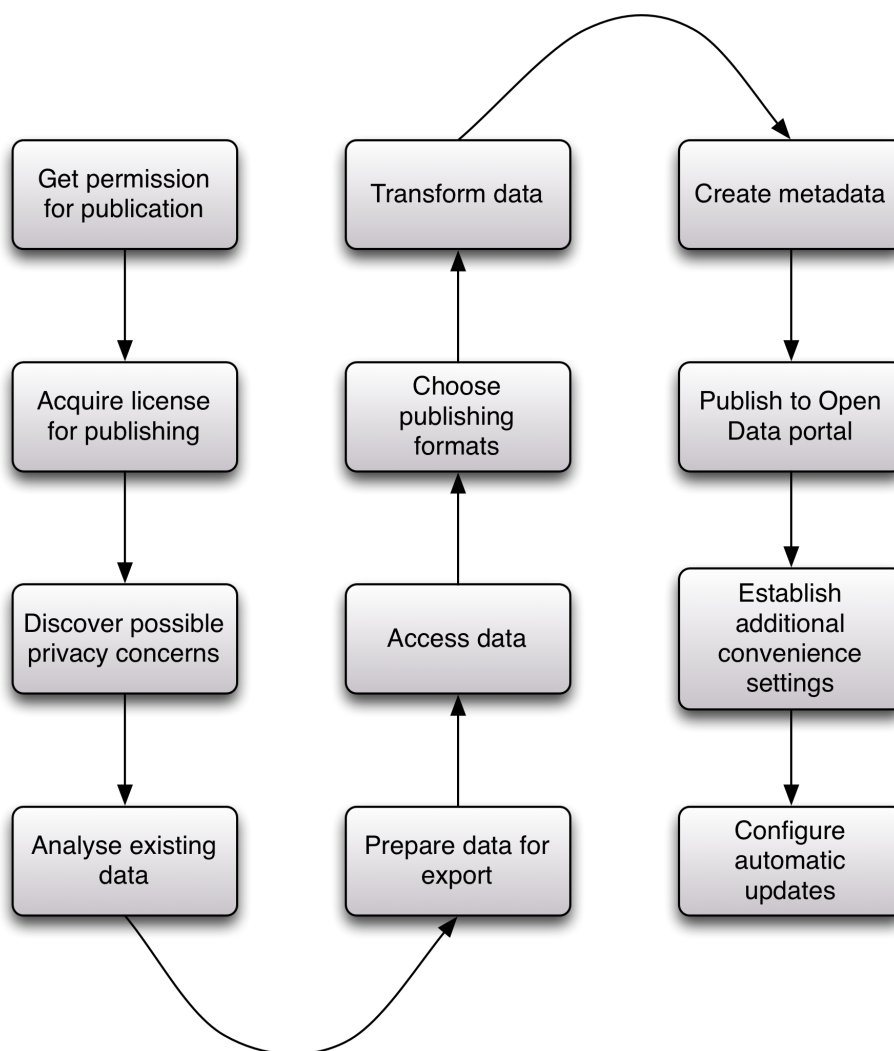


Figure 2: Overview of the publishing process.

applies to electronic data in the museum's database as well. But not all pieces of art in the database are owned by the museum. Some of them are on a so called permanent loan, for example from the Austrian Ludwig Foundation (<http://ludwigstiftung.at>), or they are just borrowed from another museum for the current exhibition. For those pieces not all exploitation rights have been transferred to the mumok. Therefore it has to be carefully considered what may be published and under which license. This matter also is the cause for the process of digitization to take a rather long time, as the permissions for publishing an item on the internet have to be granted from

the owners as well. There are two entities in the museum responsible for the collection and their digitization and publication. At first there is the department of collection, which is responsible for all pieces of art contained in the museum, currently being managed by Dr. Susanne Neuburger. On the other side there is Mag. Jörg Wolfert, head of the art education department, responsible to bring the public closer to modern art. Mag. Wolfert also coordinates the digitization project, finalizes the information in the database and marks the records as ready for publishing.

After a talk with Mag. Wolfert, an email has been sent to him by the author with a summary of the necessary permissions, that have to be granted, for publishing data of the mumok collection to an external Open Data portal. Those permissions include the fields of the records that will be published. These contain the same information that is already being presented on the mumok website. Also the sub-set of records applicable from the whole collection was defined. Each record is marked to be published with a value in specific field. This also makes it possible to automatically enlarge the exported data set as soon as a new record is released. Digital images will be represented by HTTP links to the images on the website. Those permissions have been granted by the museum as well as the possibility to automatically update the information with new data in the future as soon as it becomes available. For the permission of linking pieces of art to artists and exhibitions as well as additional information on both of them, another email to Dr. Susanne Neuburger has been sent by the author. The project was explained to her as well, including the supplementary list of necessary fields from the database.

Both requests produced a positive answer, thus the needed permissions for exporting data to an Open Data portal had been granted. The emails can be found in Appendix A.

3.3.2 Acquire License for publishing

The next step is to acquire an appropriate license for the published data. According to [Miller et al., 2008, p.2] it is necessary to deploy a license with Open Data, because such publications do not count as a creative work as defined by the international copyright laws. This protection only works for acts of creativity and therefore does not apply to databases or their contents. The legal situation for example in the United States of America in 2004 recommended to either completely lock down published data with security mechanisms or to open it up totally to the public [Miller et al., 2008, p.2]. Because of the increase of data that got published in the area of science and from governmental organizations, these challenges have been taken on

by several non-profit organizations. The goal is to explicitly define terms for using the data and encouraging others to work with them. The Open Knowledge Foundation has a list of conformant licenses on their Open Definition Website (<http://opendefinition.org/licenses/>), that originate mainly from two sources. Licenses from Creative Commons, "a nonprofit organization that enables the sharing and use of creativity and knowledge through free legal tools" [Creative Commons, 2001], and from the Website Open Data Commons, a project to "provide legal solutions for Open Data" [Open Knowledge Foundation, 2008]. In this list the available licenses are distinguished by three attributes:

- *Domain*: It defines for what scope of application the license may be used. The two available domain areas are *data*, meaning structure and format, and *content*, which stands for the actual data like text, images or documents, that are included in the data.
- *"BY"*: The second attribute is called "BY" and stands for *attribution*. The data can be used, changed or remixed even for commercial use, as long as the new document states that it has been originally created and licensed *by* the author.
- *"SA"*: This is an acronym for *share-alike*. This concept aims for further use of open licenses, as every work, that is based on the original one, has to use the same license.

When taking a closer look at the Creative Commons licenses, they are built in three layers, each of them expressing the same content to a different audience. There is the legal code written in formal jurisdictional language. Then there is the human readable version, which is an understandable summary for everybody. Finally it consists of a machine readable layer constructed with a proprietary language set, that enables computers to read what can be done with the accompanying data.

In case of the mumok, the data will be published on the Website of the Open Data Portal Austria (<http://www.opendataportal.at>). Their page about frequently asked questions states that there are two possible licences, that may be used for data resources [Open Data Portal Österreich, 2014]. The one with the most open permissions is the Creative Commons Public Domain licence CC 1.0 Universal, that transfers all rights of the publisher to anyone using the data. The second available choice is the Creative Commons Attribution 3.0 Austria licence (CC-BY-3.0-AT), that grants the permissions to share, redistribute and even change or remix the data for non-commercial as well as commercial use (<https://creativecommons.org/licenses/by/>

3.0/at/deed.en). For these rights it is necessary to credit the publisher, link to the licence and explicitly designate if the data has been changed from the original.

Here the second license has been chosen with the arguments, that in a case of data usage by a third person, it should always be known where the data originally came from and if it has been altered in any way, so that the end user knows what to expect.

3.3.3 Discover possible privacy concerns

When taking a look at the records that will be chosen, it is necessary to consider the right of privacy, that every individual human being has. This topic is also listed in the W3C's best practices for publishing data on the Web as sensitive data [W3C, 2015a]. Affected here is all data that may allow someone to "identify an individual person" [W3C, 2015a] including values like phone numbers, street addresses, finger prints or credit card numbers. A security plan should be set up, that takes care of explicitly scanning the data for such content and taking precautionary measures to ensure the protection of sensitive data or even remove them [W3C, 2015a]. It is possible to get permission from the different individuals, which may also result in the need to protect the data with security mechanisms, which in turn contradicts the open definition. In this case it is recommended to exclude those information from published data sets. The database from the mumok contains information on the artists like their personal address or other contact information. This content will be left out in the publication in order to ensure the right of privacy. For the pieces of art there also are entries of monetary concerns like the value of the object at time of procurement or the more accurate current value of insurance which can be seen as the value on them market. These entries will also be intentionally removed before the export. Also for making a difference between records, that can be published and others that still need to be reviewed and approved, a special field in the database is used (*Obj-DatenZ2.ObjC04S*) with a boolean content. The field gets set to a positive value as soon as the record is ready to be published. This also allows for a record to be excluded from the website's on-line collection as well as the Open Data export at any given time.

3.3.4 Analyse existing data

Following the process of getting the necessary permissions, choosing an appropriate licence and filtering out information that may result in a privacy issue, the data, that will be exported, has to be analysed for its structure.

Most commonly the data will be available in some sort of database. The type of database defines the way the data is stored and accessed, as well as the structure in which it is organized. In most cases it will not be possible to conduct the next steps at the live database, so it is recommended to get a copy or backup. The software for running the database can be set up in an external development environment with the backup mounted. This allows the publisher to experiment with the database and analyse the structure and relationships between the existing tables. In order to identify the necessary fields and their relations to each other it is convenient to create a sheet with all the values that should be present in the end, like for example the title of the art object, the name of the artist or the location of an exhibition, that the object has been presented at. This list will help in hunting down all of the fields in the database. Most of them can be straightforward, but others will have to be combined or can only be accessed through more complex queries. In a relational database the tables can be connected through relationships. There are one to one, one to many or many to many relationships. In order not to get lost in the database structure it is helpful to draw those relationships into an entity relationship diagram to get a visual overview of the connections. The list of fields and the diagram should be the outcome of this step.

Mumok's database runs on a Microsoft SQL Server 2008 R2, which is a relational database system. For the analysis of the data a backup of the live database has been exported. This backup has been imported on a more recent Microsoft SQL Server 2012 Express, that is freely available. As an addition the SQL Server Management Studio has been installed, which provides a graphical user interface. From this software the tables in the database can be reviewed and their relationships analysed. It is also possible to add the single tables into a composition, that also shows the relationships, which makes it easier to grasp them. The outcome of this analysis is an entity relationship diagram, which can be seen in figure 3 on page 29. The main table in the upper left corner contains information on the art objects and is called *ObjDaten*. All other information is linked to this table. There is an extension to this table called *ObjDatenZ2* which has a one to one relationship and contains even more information on the object. The table *ObjMultiple* is connected via a one to many relationship and contains fields like the title of the object or the used materials in different languages. In *ObjMass*, also a one to many relationship to *ObjDaten*, the different measures like height, width and depth with the corresponding unit of measurement are stored. The exploitation rights for the object are contained in the table *Recht*, connected via a one to many relationship. The remaining two tables, *Kuenstler* for the information on artists and *Ausstellung* for exhibitions, are

connected as a many to many relationship. In order to represent this in a relational database, an intermediate table needs to be introduced, that acts as a link. The table *Obj_Kuenstler* connects *ObjDaten* to *Kuenstler* and *Ausstellung_Obj* connects *ObjDaten* to *Ausstellung*.

3.3.5 Prepare data for export

For an optimal performance and less code in the script, that is needed to actually export the data from the database, a view can be constructed within the SQL server. This view is able to collect all information from the different tables, corresponding to their relationships, together into a newly created, temporary table which then contains all information on its own. This information can be accessed with an SQL select statement. The performance of the export is increased when using an SQL view, as the database uses its internal relational algebra to effectively collect the necessary data. Otherwise the script would either have to contain a very long select statement or would have to combine the data from the single tables itself. The database engine can already accomplish the same outcome more efficiently. Due to the fact that the file formats for the Open Data export may need a different data structure as well (see section 3.3.8), it may be necessary to create more than one SQL view.

In case of the mumok export two different SQL views have been created. The first one called *Open_Data_Full* contains inner joins via all related tables and will be used for the export to the JSON and JSON-LD file format. Because of the joins, multiple results are generated for objects that have more than one artist and/or exhibition. The second one called *Open_Data* only inner joins the table *ObjDaten* with *ObjDatenZ2*. Both SQL views are sorted by the ID of the art object from the field *ObjDaten.ObjId*, which is also used as the primary key. The other three tables, connected with a one to many relationship, are integrated into both views via sub-select queries. This allows for a specific assignment of values from the other tables to the same resulting row. Given the fact that the translations of the object's title are currently stored in the table *ObjMultiple* in German and in English, when joining the tables, we would get two resulting rows for the object. One with the full information on the art object including the German title and another one including the English title. When using sub-selects, the value of the German title can be inserted into the resulting field *TITLE.DE* and the English title into *TITLE.EN* of the same row. The title of the object in the different languages is stored in the field *OmuTypS* with the attribute "Titel (deutsch)" or "Titel (englisch)", which marks them as the value that should be used when generating an output. This should limit the possible number

of German or English titles for export, that may be attributed to one and the same object, to a single one. Because this limit does not get enforced within the database, in order to be absolutely sure, that only one value gets returned, the sub-select is limited to retrieve just the first result by using the command *TOP (1)*.

```
SELECT TOP (1) OmuInhalt01M FROM ObjMultiple
WHERE (OmuObjId = ObjDaten.ObjId)
AND (OmuGenFeldS = 'ObjTitelOriginalS')
AND (OmuTypS = 'Titel (deutsch)')
```

In other tables with one to many relationships, this is done more or less the same way. When entering information on the object's measurements, it is possible to register several different dimensions, for example the frame or the packaging. In this case each entry gets a value for specifying its importance regarding the output in the field *ObjMass.ObjSortL*, where the number one comes first. The number one in this list will also be used for the output in the SQL view.

```
SELECT TOP (1) ObmMasseHF FROM ObjMass
WHERE (ObmObjId = ObjDatenZ2.ObjZ2Id)
AND (ObmTypMasseS = 'Objektmaß')
AND (ObmSortL = 1)
```

3.3.6 Access data

For the step of exporting data from the database and transforming them to to desired output formats, a development environment should be chosen that fulfils the following needs. It has to be possible to access the live database either directly or through an on demand VPN connection. The access will most likely be achieved by using the correct software libraries corresponding to the database or by using the correct ODBC driver. The programming or scripting language used will then be able to access the data by doing a select query on the SQL views, that have been created in section 3.3.5. An account to the source database will also be needed for the publisher with SELECT permissions on the necessary tables.

On one of the museum's servers a virtual machine running CentOS 7 on OpenVZ was created and provided to the publisher with full access rights. It was possible to login to this machine from an external location through a VPN connection. The software needed to get things up and running was the ODBC driver for SQL Server on Linux provided directly from Microsoft. For the purpose of querying data from an external application a user in the

museum's database has been created, allowing SELECT statements on the necessary tables. As a scripting language, python has been selected, which only needs the ODBC package pyodbc to be able to communicate with the database. The script can then directly query content from the database.

3.3.7 Choose publishing formats

When it comes to the file format that will be used for the process of exporting to Open Data, there are several possibilities available. Every file type that conforms to the Open Definition (see section 2.1) can be used, although there are some popular ones that are more suitable than others. These will be explained in the next paragraphs.

3.3.7.1 JSON

JSON stands for JavaScript Object Notation and derives from the ECMAScript, which is a standard for object literals in JavaScript. The ECMA itself is an industry association that develops standards for Information Technology and Consumer Electronics [Ecma International, 2011]. JSON was presented in 2001 and is a sub-set of ECMAScript that shares textual representation of objects with many other programming languages [Ecma International, 2013, p.4]. Due to its simplicity its grammar will stay the same for a long time, making it a very stable notation to use. Nearly every commonly used programming language today has its own implementation of JSON. This makes it a very convenient format for interchanging data objects between different languages. It is a simple representation of key/value pairs, that allows to build objects or even ordered lists like arrays. Strings are represented as unicode characters, numbers only follow the representation that humans use, namely a sequence of digits [Ecma International, 2013, p.4]. The interpretation of the number value is then left to the programming language that uses the object. More complex structures can also be created by nesting objects and arrays. Figure 4 shows the most basic object structure, that is used in JSON, where object values are in the simplest case strings or numeric values. More complex structures can also be created by nesting objects and arrays in value positions; Figure 5 shows the corresponding notation of an array.

3.3.7.2 CSV

Another format, that has been mostly used for data interchange between spreadsheet programs for years, is called Comma Separated Values (CSV). It is a very convenient file format for depicting tabular data models, as they also exist in relational databases. The World Wide Web Consortium has a

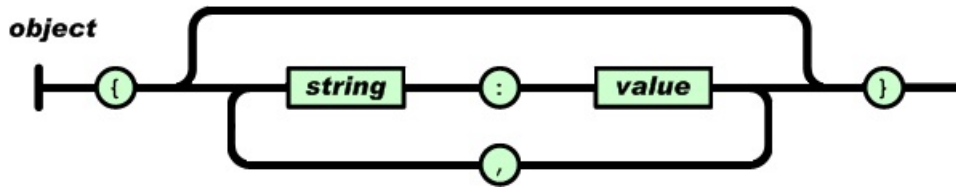


Figure 4: JSON notation of an object. Source: [Ecma International, 2001].

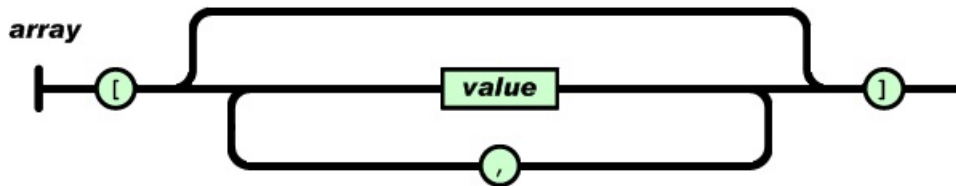


Figure 5: JSON notation of an array. Source: [Ecma International, 2001].

Working Group on CSV with the goal to improve the format for a higher interoperability when it comes to Open Data [Herman et al., 2013]. They appoint the advantages of CSV as being easy to manipulate and generate. There also are a lot of tools available that can work with this format [Herman et al., 2013]. No formal documentation exists for CSV but there is an Request for Comment (RFC) 4180 from the Internet Engineering Task Force (IETF) that can be seen as a de facto standard. The author also explains, that there are many different specifications and he tries to summarize the most important aspects [Shafranovich, 2005]: A CSV file is a text file with one or more records. Each record uses one line and the lines are delimited by line breaks (CRLF). The first line can optionally be a header line containing the names for represented fields in the lines below. In order for the reader to interpret a header row it is necessary to appoint to its existence in the MIME types header parameter. The fields are separated with commas and their quantity should be the same on the header line as well as in every other line that follows. A field may be enclosed in double quotes, but fields that contain line breaks, commas or double quotes, have to be marked with them. A double quote inside a field value has to be escaped by another double quote. These are rather simple rules that have to be followed but in order to be interpreted correctly, the usage of some of the rules have to be documented in an accompanying file. This example is a short CSV file containing an optional header row.

objid,inventorynr,originaltitle
660,MG 37/0,Zeitungsfoto 064
451,AV 25/0,Mutterschaft
3300,AV 210/0,Scape-mates

3.3.7.3 RDF

For describing resources on the Web, the Resource Description Framework (RDF) can be used. RDF is a data description concept and not primarily a file format of its own, although this concept is used in a number of formats like Turtle, JSON-LD or RDF/XML. The World Wide Web Consortium also has a primer on this topic, which suggests RDF appropriate when "information on the Web needs to be processed by applications" [Schreiber and Raimond, 2014]. The meaning of the data does not get lost when data in RDF format is exchanged between applications. When publishing data on the Web its information can be linked to even more, already existing information. These links can be followed by human beings manually as well as automatically by machines and applications. According to [Schreiber and Raimond, 2014] the main advantages are:

- The possibility to add information to data on the Web, that can be read by machines.
- Enriching data by making links to other data sets containing more information.
- Using it for exchanging data between different applications in a standards-compliant way.

The described resources in RDF consist of a triple of subject, predicate and object, where subject and object are two resources related to each other. This relationship is built by the predicate. These triples can also be visualized as a labelled direct graph that is directed from subject to object. The textual example from [Schreiber and Raimond, 2014] looks as follows:

```
<subject> <predicate> <object>  
<Bob> <is a> <person>  
<Bob> <is a friend of> <Alice>  
<Bob> <is born on> <the 4th [sic] of July 1990>  
<Bob> <is interested in> <the Mona Lisa>  
<the Mona Lisa> <was created by> <Leonardo da Vinci>  
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>
```

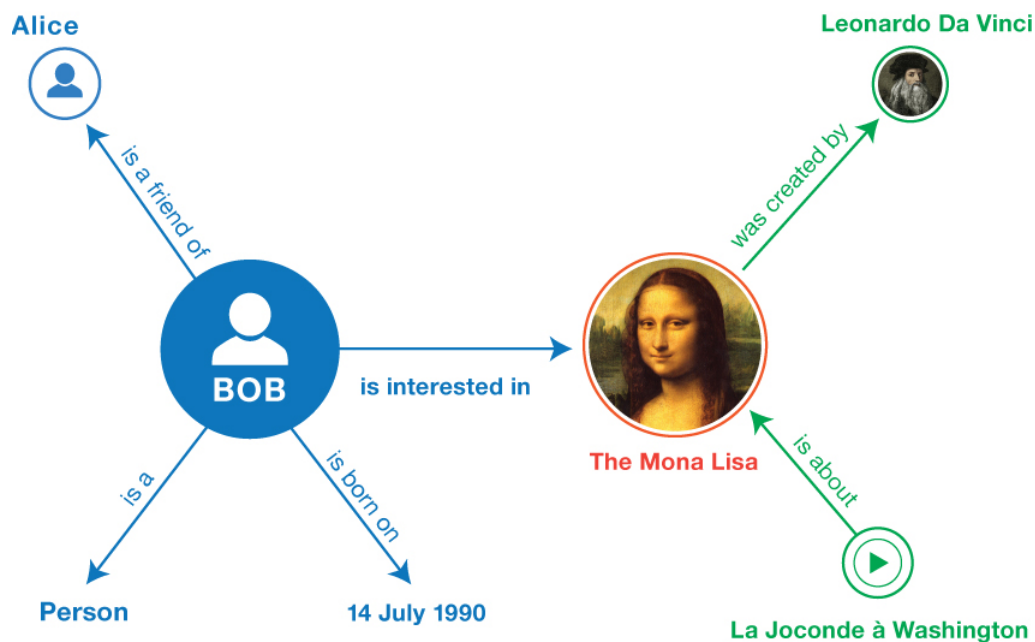


Figure 6: RDF graph of textual representation example. Source: [Schreiber and Raimond, 2014].

The same information can then be visualized in a graph, as can be seen in figure 6.

There are three types of data that can be used in RDF [Schreiber and Raimond, 2014]. The first one is the International Resource Identifier (IRI), which can be either an URL or any other address in this form. It does not have to be known how to access it or where it is located. In this case the IRI is used as a global identifier and can be utilized in the subject, the predicate and the object. A data type only available for the object type are literals, which means a representation of string, a number or a date, accompanied by a data-type like integer or float, so that the value can be interpreted correctly. As a third option there are blank nodes, that can be put into the subject or the object position, explicitly describing a resource that does not necessarily need an IRI as an identifier. A blank node can be used when describing an unknown resource if it is necessary to link to or from this resource as well.

3.3.7.4 JSON-LD

When looking at Tim Berners-Lee's Open Data five star rating, it is unavoidable to produce linked data when the highest rating shall be achieved.

There is a document format called JSON-LD, which stands for JavaScript Object Notation Linked Data, that allows a transition from pure JSON to linked data in the RDF format [Sporny et al., 2014]. The World Wide Web Consortium also has a community group on this topic which finished the first released specification in March 2013. This format allows already existing systems, that are used to working with JSON, to be able to transform plain data into linked data, as the whole syntax is conforming with the JSON specification. On the other side, such data can then be used as a RDF data format, that can be handled the same way as a Turtle or RDF/XML document. It extends the existing specification with the possibilities to use a universal identifier by using IRIs, refer to other JSON objects on the Web, determine the datatype of the values and to designate strings with their corresponding language [Sporny et al., 2014]. This format makes high use of standard vocabularies like the Data Catalog Vocabulary (DCAT) or the community vocabulary from schema.org. According to [Hyland et al., 2014] the usage of vocabularies is necessary to support the expansion of data on the Web. JSON-LD defines a context of used vocabularies which makes it clear to everybody in what environment the data is placed. For example on schema.org there is the definition of an object type VisualArtwork (<https://schema.org/VisualArtwork>). It defines the parent objects, from which properties may be inherited, the properties of the object itself as well as more specific types available of that object. These attributes can then be facilitated in the context to describe the values used in the JSON object. The example here shows the field *default_image* being of the type <https://schema.org/thumbnailUrl>:

```
{
  "@context":
  {
    "default_image": "http://schema.org/thumbnailUrl"
  },
  "default_image": {
    "@id": "http://www.mumok.at/test.jpg"
  }
}
```

When the field *default_image* is then used in a JSON object, it is known that the value is predefined on the schema.org website and can also be automatically transformed to this:

```

{
  "@context": "http://schema.org/",
  "thumbnailUrl": "http://www.mumok.at/test.jpg"
}

```

As only one vocabulary is used, the main URL defining the context is written at the top and the descriptor of the field is transformed to the used property from schema.org, namely *thumbnailUrl*. The key *@id* in the first example states that the following IRI is a unique identifier. The context may also contain multiple sets of vocabularies which can then be used in form of a prefix in front of the key. When a JSON document has to be enhanced to JSON-LD the context can also be included as a reference to an external file, in which it is defined.

3.3.8 Transform data

The publisher will have to transform the data from the database into the desired output formats by using a programming or scripting language. Therefore it is necessary to integrate the libraries for CSV, JSON and RDF into the already set up programming environment. In a simple case, when querying the data from its source, it would be possible to take every resulting row from the database and directly transform it into a JSON object or a line in a CSV file. In most cases the data structure in the database will be a little more complicated and so the strategies of handling the data have to be planned. These complications come from the already mentioned many to many relationships in the database (see section 3.3.5) and have to be processed differently for each output format. With JSON it is possible to add redundant data to every object, so the joined output of the tables can be used. A mechanism has to be programmed that attaches the additional information from the linked tables correctly to the main object. When CSV should be the output format, each table from the entity relationship diagram (see figure 3) can be produced as its own CSV file, including primary and foreign keys, as CSV also uses the tabular data format. For JSON-LD it is required to provide unique identifiers through the use of IRIs. Objects in a database have a primary key that differentiates the objects uniquely. When publishing linked data, this is not enough, as the primary key usually is just an integer value, that is used in many other databases as well. The key should at least be used in a meaningful context. For example the primary key "8931" of an exhibition could be used as an IRI in the following form: *http://www.somemuseum.info/exhibition/8931*. This IRI does not have to link to content at the moment, it only has to uniquely identify the object and can eventually be provided with its on Website later.

As the entity relationship diagram (see figure 3) for the mumok shows, the main data object is the piece of art, surrounded by the linked tables, that contain more information on it. The first output format for the publication of mumok's data was JSON. The piece of art will also be the main object in the JSON file, with information on its artists and exhibitions. All data exported will be represented in UTF-8. The URL for the thumbnail image, that is added to every object, has to be queried from the database of the mumok website. For this case the thumbnail URL will always be of the format `https://www.mumok.at/imageobject.php?objid=` followed by the ID of the object. This small helper page looks up the image for the corresponding object in the database and presents it to the user. A python script (`sql_to_json.py`) has been created, which is able to access the Microsoft SQL Server database and query the contents of the SQL view `Open_Data_Full`. The result gets sorted by the unique ID `ObjId` of the object in the database. Due to joining the linked tables in the SQL view, multiple rows starting with the same object ID will be returned if an object contains more than one artist or exhibition. These can then be walked through, creating the main object at first and then subsequently attaching additional informations from the other resulting rows to the object. As soon as the object ID changes to another number, the first object is completed. Artists and exhibitions are added with their primary database key as identifier. Datetime objects from the Microsoft SQL Server have to be converted to strings. In the end the data gets written out to a JSON file (`data.json`). This file can be sent through a JSON validator like `http://jsonlint.com/` to be sure that the structure is valid. It is advisable to start with just one row from the SQL view and see if the JSON output is valid. Then the corresponding exhibitions and artists can be added to the object and finally all objects can be processed at once.

For the creation of the CSV format the resulting data structure will be more or less the same as it is in the database, because both sides rely on the tabular data model. The SQL view `Open_Data` combines the table for the piece of art (`ObjDaten`) with all other tables, that have a one to many relationship. The output is also sorted by the ID of the object. This information will be written to the first (`obj.csv`) of the resulting five CSV files. At the very beginning of the CSV file a byte order mark has to be placed to allow other programs to identify this text-file as encoded in UTF-8, followed by the first line as an optional header line. This header line contains the values of the columns returned by the database server. It is possible to change the rules like the usage of double quotes or the field separators when transforming the data into a CSV file in the python script. The default settings in python although conform to the rules in RFC 4180 and will therefore not be changed. Then the tables for linking objects to exhibitions

(*Ausstellung_Obj*) are queried from the SQL server and written to a CSV file *obj_exhibition.csv*, followed by the table *Ausstellung* that gets exported to a file called *exhibition.csv*. This process continues until the information from all the tables is written to separate CSV files. In order to limit the output of exhibitions and artists to the ones that are linked to exported objects, it is necessary to join those tables with the table *ObjDatenZ2*. It contains the field *ObjC04S* which defines the object as ready to be published. The CSV files can be checked for validity as well, which has been done on the website <http://csvlint.io/>.

The third output format JSON-LD has the requirement of a context, explaining which vocabularies have been used, and unique identifiers through IRIs. The website schema.org has a set of vocabularies for pieces of art, in this case the used object from there is called *VisualArtwork*. Another vocabulary is also needed to describe other data types, that is defined in the Dublin Core vocabulary (<http://dublincore.org/>). The context part will be included at the top of the JSON file as its own object (*@context*) followed by another object called *@set*, which contains the rest of the data. The key *@vocab* defines the main vocabulary used and the key *dc* defines the secondary vocabulary from Dublin Core.

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "dc": "http://purl.org/dc/elements/1.1/",
    ...
    "url": "@id",
    ...
    "name": "familyName",
    "firstname": "givenName",
    ...
  },
  "@set": [
    {
      ...
    },
    ...
  ]
}
```

The field *url* gets defined as *@id*, which makes the URL for the art object the identifier for the JSON object. The other used fields also get described

by their corresponding terms from the vocabulary. So their meaning, data type or language is strictly defined and more information can be found on the linked URL. Unique identifiers have to be created for objects, artists and exhibitions. As the objects and artists are published on the mumok website as well, they already have their own URL there, that can be used as an IRI in the context of JSON-LD. Unfortunately the URL on the website does not contain the same ID as the object does in the database. But the objects do get imported to the website from the database as well, so the database of the website has a mapping of database ids to website ids. A user for the website's MySQL database has been set up to allow select queries on the two tables *mumok_collection_import_objects* and *mumok_collection_import_artists*. It is possible to query the ids of the object's website as well as the artist's website, from which an URL can be constructed, that can be used as the IRI and that also links to the website. Exhibitions don't work with this schema and are also not linked to objects on the respective website. The IRIs for exhibitions are created as blank nodes containing the exhibition's ID from the database prefixed with *_e:*. When an object or an artist cannot be found in the database of the web-server it also gets a blank node identifier *_o:* or *_a:*. The remaining steps of the script are more or less the same as in the JSON export script. At the end the data gets written to a file called *data.jsonld*. This concludes the extraction of data from the database and their transformation to the desired output formats JSON, CSV and JSON-LD.

3.3.9 Create metadata

Regarding to the Ten Principles for Opening Up Government Information [Sunlight Foundation, 2010] the first principle demands for completeness, which also includes the provision of metadata. This can serve as a description for humans and machines alike to be able to decipher the meaning of the values in the different Open Data formats. The working group for Open Government Data in Austria developed a set of metadata [Habernig et al., 2015, p.4], that should be used on the data portal to describe the available resources, when publishing data as a governmental institution. There is a set of twelve mandatory fields like metadata identifier, title, the license used, resource URL or maintainer. For even more information another set of 21 optional fields exists. The documents themselves should also be accompanied by metadata. The World Wide Web Consortium defines a model for metadata concerning tabular data models on the Web [Tennison et al., 2015]. This can be used for CSV files. The schema comes in the form of a JSON document describing the various fields and their data types and also the pri-

mary and foreign keys that connect the data to other CSV files. Also the formatting rules that have been used for creating the structure of the CSV file will be written in the metadata. Another definition of the JSON Table Schema from the Open Knowledge Foundation describes the process with an example on how to transform from a tabular data model to the JSON table schema [Open Knowledge Foundation, 2015]. In order to find the correct metadata for a CSV file, the HTTP response should include a link header, that points to the description file, when accessing the resource. For JSON files the same JSON Table Schema can be attached at the end of the file as a description for all available fields. JSON-LD solves the issue of the need to include metadata by linking the field's values to its corresponding data type descriptions in existing vocabularies.

In mumok's data export there are five different CSV files for their corresponding tables in the database. For each one of them an extra file containing the metadata has been created. The files have been created manually as their content stays the same until the underlying data structure changes. The files are named like the CSV file postfixed by the string *-metadata.json*. They include a number of informations like the URL to the CSV file, the publisher or the table schema. This schema further contains all fields used along with a description and a data type. Finally the field which contains the primary key is appointed at the end. The following shows an overview of the most important attributes:

Name: @context
 Description: This attribute defines the namespace and the language that is used for the attributes in the metadata document.
 Example:

```
"context": [
    "http://www.w3.org/ns/csvw",
    {"@language": "en"}
  ]
```

Name: url
 Description: The URL of the CSV file that this metadata document describes.
 Example:

```
"url": "http://data.mumok.at/data.csv"
```

Name: dc:title
 Description: A Title for the data in the CSV file. The prefixed value *dc* is described in the CSV on the Web namespace as a reference to another namespace, namely <http://purl.org/dc/terms/>.
 Example:

```
"dc:title": "Objects from the mumok collection"
```

Name: dc:publisher
Description: Contains information on the publisher of the CSV file, e.g. including a name and an URL.

Example:

```
"dc:publisher": {  
  "schema:name": "mumok - museum moderner  
  kunst stiftung ludwig wien",  
  "schema:url": {"@id": "http://www.mumok.at"}  
}
```

Name: dc:license
Description: The license under which the document is released. The value in this attribute is prefixed with the keyword *@id* which is used when the value is an URL through which it can be uniquely identified.

Example:

```
{"@id": "https://creativecommons.org/licenses  
/by/3.0/at/legalcode"}
```

Name: dialect
Description: This attribute describes the formal structure of the CSV file including the encoding, the existence of a header, the character that acts as a field delimiter or the symbol used for quoting text.

Example:

```
"dialect": {  
  "encoding": "utf-8",  
  "quoteChar": "\"",  
  "header": true,  
  "headerRowCount": 1,  
  "delimiter": ",",  
}
```

Name: tableSchema
Description: This attribute is a JSON object by itself and includes an Array named *columns* that describes the different columns in the CSV file.

Example:

```
"tableSchema": {  
  "columns": [  
    ...  
  ]  
}
```

Name: primaryKey
Description: The column in the CSV file that is used as the primary key, which has a unique value throughout the file and can be referenced from another CSV file.
Example: "primaryKey": "objid"

Name: columns
Description: This is an array of all the columns that exist in the CSV file describing them with the attributes like *name*, *dc:description*, the used *datatype* and if the field has to be filled out.
Example: "columns": [{
 "name": "objid",
 "dc:description": "An object identifier.",
 "datatype": "integer",
 "required": true
}]

An exemplary metadata file can be seen in section Appendix D. The JSON file (*data.json*) has been extended by transferring the object's data in its own object called *@data* and the description in the new object *@metadata*. The structure is the same as it has been used for the CSV's metadata, but it combines the information for all available fields into a single object. As the JSON file is created from scratch every time the script runs, the metadata information is added by a function in the script itself, but also has to be changed when the data structure gets altered. No changes have to be made to the JSON-LD file, as the language already integrates existing schemas from vocabularies into the created data.

3.3.10 Publish to Open Data portal

After the creation of the data files in different file formats, the publisher has to find a way to host the files on the Web. The most straightforward path is to publish them on the organization's website where they can be accessed by users. A more flexible approach would be to host the files on an additional server or virtual host with its own sub-domain. The files could then be linked to from the organization's website as well as from other sources. It is also possible to set up a platform that is dedicated to the management of many Open Data sets. There are open source solutions like the Comprehensive Knowledge Archive Network (<http://ckan.org/>), short CKAN, that is also used in Austria for the portal data.gv.at and data.opendataportal.at

for such purposes. DKAN (<https://www.drupal.org/project/dkan>) is another open source software. It is released as a CKAN module for the content management system Drupal, that can be integrated in existing websites running on top of this CMS. Commercially hosted platforms are also available like the Socrata Open Data Portal (<http://www.socrata.com/products/open-data-portal/>). All of these software solutions have a tool for user feedback integrated as well as an Application Programming Interface (API) that helps to manage changes and updates through a program.

An existing Open Data platform has been chosen for mumok's files. The Open Data Portal Austria (<https://www.opendataportal.at/>) is set up with CKAN and offers its services for free, so non-governmental organizations or private persons can publish their data at a central location. An account has been created on their site. Then the portal's administrators have been contacted to explain that data will be published in the name of an organization. They have then changed the account type accordingly. To publish the files a data record can be created which provides a textual description and other mandatory metadata fields. The data export of the mumok needs six different data records: one containing the whole information on pieces of art combined with their artists and exhibitions in the file formats JSON and JSON-LD, and five others for every different CSV file. To each data record one or more resources can be added, in this case the corresponding files. CSV resources get accompanied by the metadata file in form of a JSON table schema. The files can either be uploaded to the portal or hosted on an external site, in which case only the URL to the file has to be entered. As the mumok will provide a daily update on the data, the resources are linked externally. For this to work an additional virtual host *data.mumok.at* has been created on the web-server, that also hosts the mumok website.

3.3.11 Establish additional convenience settings

Some more configuration has to be done on the server side where the files are actually hosted. Here the Apache Webserver 2.2 has been used to host and deliver the data. The default configuration already delivers the right MIME content type for JSON files but is missing the one for JSON-LD as well as some additional settings required for CSV. For JSON-LD files the following configuration has to be set:

```
<Files ~ "\.jsonld?$">  
    Header set Content-Type "application/ld+json"  
</Files>
```

The guidelines for the tabular data model from the World Wide Web Consortium state that a link header should be provided when accessing the CSV file via HTTP, that helps to locate the necessary metadata [Tennison et al., 2015]. The content type should also get advertised correctly including the file's encoding [Tennison et al., 2015] and specifying the existence of a header row. For these two settings to be handled correctly by the web-server, these configuration lines have been added:

```
<Files ~ "\.csv?$">
  Header set Content-Type "text/csv; charset=utf-8; \
  header=present"
  SetEnvIf Request_URI "(.*)" Request_URI=$1
  Header add Link "<{%Request_URI}e-metadata.json>; \
  rel=describedBy;type=application/csvm+json"
</Files>
```

3.3.12 Configure automatic updates

The principle of timeliness says, that in an optimal situation new data should be made available to the public immediately [Sunlight Foundation, 2010, p.1]. This requirement can be fulfilled by producing a new data export automatically in a set interval of time. The frequency of those updates depends on the criticality of the data and can vary from update upon changes to once a week. So some kind of mechanisms have to be installed by the publisher to automate the whole process of fetching data from the database, transforming it to the desired output formats and finally publishing the new content to a location on the Web. For automatic updates of mumok's Open Data files a bash script has been created that takes care of calling the python scripts, that connect to the database containing the source data. The data gets extracted from the Microsoft SQL Server and transformed into the three desired output formats, CSV, JSON and JSON-LD. This process is described in detail in section 3.3.8. The resulting files have to be transferred to their final location, in this case the directory from which the web-server for data.mumok.at provides its files. As this will happen with secure copy (scp) an ssh-key with an empty password is needed to allow authorization via the bash script. The files can then be copied without the need of any passwords. As a last step the data portal has to be notified of the changes. The CKAN installation provides an API that can be used for such a purpose. A python script has been used for this task. In order to allow an external application to change metadata values on the portal, an API key is needed, which can be copied from the account's profile page. Then the ID's of the available data record's

resources have to be noted. This can be done either by querying the API or by extracting the ID from the URL in a web-browser. This example shows a full URL with the ID *27ff4b2b-2d2b-4db6-897e-4e4ae927d933*, that is needed to manipulate this record:

```
http://data.opendataportal.at/dataset/  
ausstellungen-der-sammlung-des-mumok/resource/  
27ff4b2b-2d2b-4db6-897e-4e4ae927d933
```

For every newly uploaded file the change date and the size of the file have to be updated by issuing a HTTP request to the API. For an automatic execution of the bash script once a day, the cron daemon has been used, that sends an email with the results to the publisher's email-address. This message also serves as a reminder that such a script is running every day and to encourage the publisher to check on the correct outcome at regular intervals. It can also be helpful to catch up on errors if any of the underlying external mechanisms change in the future and are then not compatible with the script any more. Changes could either be introduced in the CKAN API or in any of the used software (Microsoft SQL-Server, Python, etc.) and the process or the programming code will have to be adapted accordingly. If the structure of the used tables in the database are changed for instance by an update of the MuseumPlus application, the created SQL views will throw an error, which will also be reported in the same email message. As means of discovering problems with the automation process as quickly as possible, mumok's monitoring system checks for the file age of the published files directly on the web-server and informs the staff as soon as those files are older than 25 hours.

4 Evaluation

The resulting documents can be evaluated for quality and openness by applying three different methods of measuring, as already presented in section 2.8. When using Tim Berners-Lee's five star rating all documents in CSV or JSON format will receive three stars, whereas the JSON-LD file can be attributed with four stars as an open standard for linked data concepts is used. In regard to the Ten Principles for Opening Up Government Information, as explained in section 2.5.1, the documents can fulfil most of the ten requirements. As those principles are just a collection of attributes that have to be taken into account when producing Open Data, they are no real instrument for measurement. So the author created a scale where every principle can add

ten points to a total score and a maximum of 100 points is achievable. This grants the possibility to evaluate and compare the quality and usefulness of the documents with others.

Completeness (Score: 8/10). The data is complete only within the borders of its predefined area. Not all pieces of art from the museum's collection are published to the Open Data platform. This is a restriction set by the organization itself, as they only want to present data, where they can vouch for correctness. Metadata is included as well, so a score of eight can be counted.

Primacy (Score: 10/10). The museum is the primary source of the originating data, as the information gets researched there and is published directly from their database. This results in a score of ten.

Timeliness (Score: 9/10). A new update to the released information is published in the three available file formats to the Open Data portal on a daily basis. Real time publication is currently not implemented, but due to the fact that this is not time critical data, this item is worth nine points.

Ease of Physical and Electronic Access (Score: 10/10). The documents can either be downloaded directly from mumok's web-server or accessed via the Open Data portal. The only needed resource for access is an internet connection, which is publicly available in libraries or other governmental buildings by now. A score of ten points can be achieved.

Machine readability (Score: 10/10). All available document formats can be read by a machine, which adds ten points to the total score.

Non-discrimination (Score: 10/10). A membership or identification is not needed for access. The data is also available at any given time. This results in a score of ten.

Use of Commonly Owned Standards (Score: 10/10). For the production of the documents only open-source software is used, which is freely available to everybody. The open-source components include the operating system for the programming environment, the ODBC driver from Microsoft for accessing the database, the python scripting language for transforming the data and the Apache web-server making the data available on the Web. A full score of ten points can be achieved here.

Licensing (Score: 10/10). The licence under which the documents are published is the Creative Commons 3.0 Attribution licence, that allows commercial and non-commercial use of all data. The source and author have to be named when the data is incorporated into another work or publication. As the data is marked from where it originates and is available without any impediments, a full score of ten is allocated.

Permanence (Score: 0/10). Whenever a document is published it should be available forever. This permanent availability is not applicable with the Open Data export of mumok, as every new update overwrites its predecessor.

A score of zero is reached. In order to improve this score a mechanism for keeping each daily copy has to be installed. Every day a new export of data is created, the files on the web-server have to be altered to include the creation-date and postfix it to the file-name. All of the files can then still be accessed with a consistent naming scheme. This also involves the need of more storage space on the web-server, as the number of files will grow every day.

Usage costs (Score: 10/10). Accessing the data is not tied to any monetary values or other costs. This gives a score of ten.

As a total score a value of 87 out of 100 is reached for the conformance to these ten principles. This is a good score as 87 percent of the goals have been reached. The results can also be evaluated with another method, which provides a different outcome.

The Open Data Institute in London issues certificates for Open Data releases based on information provided in a dynamic questionnaire. This process is described in section 2.8. After filling out the necessary forms on their website, the data gets sorted in one of the available categories. Currently the first level of certification named *raw* can be achieved. This is due to some requirements missing for the next level, which are shown below with a short explanation on how to improve:

- An URL scheme should be used for different release versions. Currently the most up to date files can be retrieved with a static URL, e.g. `http://data.mumok.at/data.json`. When the files get archived permanently, every day a new version gets published, they will be renamed including their creation date. The same JSON file from October 15, 2015 could then be accessed by using the URL `http://data.mumok.at/data.json?2015-10-15`.
- No technical documentation is provided at the moment. It will be provided and linked from mumok's website as well as the Austrian Open Data Portal in the future.
- A rights statement should be published with the data. This should be created by the publishing organization. The mumok does not have a legal department and therefore the creation of such a statement has been postponed to a date where the resources will be available.
- Instructions on how to suggest improvements have to be published.
- Extra contact information for people having questions about privacy should be made available.

- The risk of disclosing personal data has to be formally assessed. This means that a report has to be written on the analysis of the data, why some information has been excluded and how the permissions for published private information have been granted from the owner.

5 Related Works

Within the Austrian Open Data portals no data sets from other Austrian museums can be found as of today. The Wien Museum, which is dedicated to conserving the history of Vienna, including architecture, art and fashion, has published its collection of currently 10.079 Viennese fashion objects to the Europeana Fashion portal. This data set can then be queried via Europeana's SPARQL endpoint. The Amsterdam Museum created a set of linked Open Data from their object collection that conforms with the object metadata collection from Europeana [de Boer et al., 2013]. In the United States the Smithsonian American Art Museum also published the data of their collection of objects in form of linked Open Data [Szekely et al., 2013].

5.1 Amsterdam Museum

The Amsterdam Museum's mission is to collect and preserve the history and objects of the city of Amsterdam and its citizens. Currently they have around 70.000 objects in their permanent collection. The digital metadata and information is handled by the Adlib Museum Software [de Boer et al., 2013, p.1]. Like the mumok this museum published their collection to the internet by 2010. They also provide an XML REST API for accessing the data. In order to provide linked data they worked together with the Europeana project that aggregates data from several European cultural institutions into a single web-service and make them available as five star linked data. When using this approach the metadata has to be tailored to their specifications [de Boer et al., 2013, p.2], which provide interoperability by the different data sets from multiple institutions. The Amsterdam Museum wanted to keep their own structure of metadata and therefore developed means of preserving this while also conforming to Europeana's rules. As an outcome they also produced five star linked data that introduces an interoperability layer in order to work with the Europeana Data Model. The museums' data consists of three parts, the metadata for the pieces of art, a number of thesaurus concepts, that can be used for a uniform description, and a set of persons that can be linked to the other two [de Boer et al., 2013, p.2]. URLs have been assigned to all of those records by using purl.org, which provides

Persistent Uniform Resource Locators, as the developers had no access to the web-servers URL schema at that moment. The Europeana Data Model (EDM) uses all sorts of existing vocabulary sets like Dublin Core (DC) or Visual Resources association (VRA). The Amsterdam Museum uses its own set of vocabulary. In order to conform with the EDM they introduced an interoperability layer of RDF triples, that define their own properties as sub-properties of their corresponding value in the EDM [de Boer et al., 2013, p.4]. This resulted in 90 properties being sub-properties of Dublin Core and seven mapped to EDM-specific properties. The 28.000 terms used in the thesaurus are compatible with the Simple Knowledge Organization System (SKOS) and get externally linked with two services. Non-geographical terms were pointed to the Dutch Art and Architecture thesaurus (AATNed) and geographical terms were bound to GeoNames [de Boer et al., 2013, p.5]. The records in the data set of persons was mapped to the Getty Union List of Artist Names (ULAN) and DBpedia.

This process of transforming their digital collection to a five star linked data set resulted in the creation of 6.603.333 RDF triples describing a number of 168.413 resources with 134 different predicates [de Boer et al., 2013, p.4].

5.2 Smithsonian American Art Museum (SAAM)

The museum maintains a collection of American art and is situated in the heart of Washington. Several kinds of art are contained in its catalogue like photography, visual culture or video games over a time period of three centuries [Smithsonian American Art Museum, 2014]. In 2013 the museum opened up their data and published them as linked Open Data. The relational database contains a total number of 41.267 objects with their corresponding 8.261 artists [Szekely et al., 2013, p.3]. As the very first step the team carrying out the project had to focus on legal aspects. Their solution was to only use data that has already been published on the museum's website, which means that those information had already received a positive outcome in the verification process of the legal department. Further on an ontology was defined that extends the Europeana Data Model (EDM) with SAAM's own subclasses and sub properties. In addition the project also used classes and properties from schema.org to map geographical data as well as RDA Group 2 Elements for biographical information [Szekely et al., 2013, p.4]. The museum decided to extend the EDM because that would maximise the compatibility with similar datasets from other museums [Szekely et al., 2013, p.3]. In order to map the values from the relational database to the created schema, their own tool called KARMA has been used. KARMA works by taking structured data as an input and transforming it to RDF output con-

forming to an ontology the user provides. It takes care of some steps in the process through automation and also provides a visual interface to the users, where they can actually see the mappings and fine-tune the resulting outcome [Szekely et al., 2013, p.5]. For this to work the structured data had to be prepared in the relational database by creating SQL views of the data and gets additional support from stored procedures. This step can be carried out by the system administrators and greatly simplifies the data structure available to the people responsible for the mapping process. Classes and properties can then be connected between the museum's data schema and the schema of the extended EDM model [Szekely et al., 2013, p.7]. The museum also linked the resulting data of artists to DBpedia, the Getty Union List of Artist Names and the Amsterdam Rijksmuseum's dataset [Szekely et al., 2013, p.10]. As a way to ensure the factor of being up to date and providing the correct information, the KARMA interface also allows employees of the museum to review the mapped results and also gives them the possibility to edit those matches [Szekely et al., 2013, p.12].

6 Future Works and Conclusion

The concept of Open Data not only returns values generated by tax money back to the citizens who paid for it, but also allows the creation of new applications based on the public data. This then produces even more value for the public in form of combined data from different sources. It also opens new ways of involvement for people by being able to see what is produced in public or private organizations and contributes feedback or new information to them. Governments are producing Open Data for some years now, but the knowledge of its existence and what can be done with it is still not widely spread. Some standards are defined for the different file formats as well as for the concept of Open Data. These include rules that should be followed when putting data on the Web and what licenses can and should be used for which purpose. This paper also is an overview of a step-by-step process of publishing Open Data from an existing database. Some examples, decisions and programming code are provided that may help even more people to do the same with data in other organizations.

The offered services surrounding this Open Data publication should be improved by following the steps listed by the Open Data Institute in order to reach the subsequent levels of certification. This includes presenting more information like technical documentations, how the rights to the published data are claimed, how the personal information for the published artists are protected or the possibility of asking questions or providing feedback. This

could be realized by transferring documentation and programming code, including examples, to a platform like Github, where their development can be followed or improved on by everybody. The Ten Principles should also be followed more closely. For every change in the content of the published documents a new version and/or the differences should be published, keeping the older ones and providing access to them via a consistent URL naming scheme containing the date of the desired resource. Also a publication in real time could be established by using triggers in the SQL database, that start the extraction and transformation process on demand, only when changes to the underlying data occur. The part that lies in the responsibility of the museum is the further publication of additional records in order to be able to publish the whole art-collection to the Web. For the improvement of the quality and value of the documents itself it would be necessary to implement more linked data concepts. Locations and people could be associated with other vocabularies or information in other datasets like DBpedia or Europeana. Also a data representation in RDF, as a result of processing and refining the JSON-LD document, could result in providing another kind of access to the data in form of a SPARQL endpoint.

References

- Berners-Lee, T. (2006). Linked Data - Design Issues. <http://www.w3.org/DesignIssues/LinkedData.html>. Accessed: 2015-10-04, Article edited 2010.
- Chignard, S. (2013). A Brief History of Open Data. *Paris Tech review*, <http://www.paristechreview.com/2013/03/29/brief-history-open-data/>, March 29th, 2013, Accessed: 2015-08-16.
- Creative Commons (2001). About - Creative Commons. <https://creativecommons.org/about>. Accessed: 2015-10-11.
- de Boer, V., Wielemaker, J., van Gent, J., Oosterbroek, M., Hildebrand, M., Isaac, A., van Ossenbruggen, J., and Schreiber, G. (2013). Amsterdam Museum Linked Open Data. *Semantic Web*, 4(3):237–243.
- Dietrich, D., Gray, J., McNamara, T., Poikola, A., Pollock, R., Tait, J., and ZijlstraTon (2014). Open Data Handbook. <http://opendatahandbook.org/guide/en/why-open-data/>. Accessed. 2015-08-14.
- Ecma International (2001). JSON. <http://json.org>. Accessed: 2015-10-18.
- Ecma International (2011). What is Ecma International. <http://www.ecma-international.org/memento/index.html>. Accessed: 2015-10-18.
- Ecma International (2013). ECMA-404: The JSON Data Interchange Format. Technical Report October, ECMA International, <http://ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>, Geneva.
- Habernig, C., Harm, R., Höchtel, J., Jörg, W., Kaltenböck, M., Lutz, B., Mittlböck, M., and Pawel, S. (2015). OGD Metadata – 2.3. Technical report, E-Government Bund-Länder-Gemeinden, http://reference.e-government.gv.at/uploads/media/OGD-Metadaten_2_3_2015_02_19_EN.pdf.
- Herman, I., Brickley, D., and Tennison, J. (2013). CSV on the Web Working Group Charter. <http://www.w3.org/2013/05/lcsv-charter.html>. Accessed: 2015-10-18.
- Herzberg, B. (2014). The Next Frontier for Open Data: An Open Private Sector — Voices. <http://blogs.worldbank.org/voices/next-frontier-open-data-open-private-sector>. Accessed: 2015-08-30.

- Höchtel, J., Parycek, P., and Straße, K. D. (2014). Open Government Data Austria - Organisation, Procedures and Uptake. Technical report, Danube University Krems, http://www.w3.org/2013/share-psi/wiki/images/f/fc/Samos_SharePSI_Austria_UptakeandImpact_fin.pdf.
- Huijboom, N. and Broek, T. V. D. (2011). Open data : an international comparison of strategies. *European Journal of ePractice*, 12(March/ April 2011):1–13.
- Hyland, B., Atemezing, G., and Villazón-Terrazas, B. (2014). Best Practices for Publishing Linked Data. <http://www.w3.org/TR/ld-bp/>. Accessed: 2015-09-25.
- Janssen, M., Charalabidis, Y., and Zuiderwijk, A. (2012). Benefits, Adoption Barriers and Myths of Open Data and Open Government. *Information Systems Management*, 29(4):258–268.
- Klessmann, J., Denker, P., Schieferdecker, I., and Schulz, S. E. (2012). Open Government Data. Technical report, Federal Ministry of the Interior, Division O1, Alt-Moabit 101 D, 10559 Berlin, http://opendataforum.info/files/OECD_Barbara_Ubaldi.pdf.
- Krabina, B., Prorok, T., and Lutz, B. (2012). Open Government Implementation Model. Technical report, KDZ - Centre for Public Administration Research, <http://www.kdz.or.at/sites/default/files/documents/kdz/news/Open%20Government%20Implementation%20Model%20KDZ%20V2.0-final.pdf>, Vienna.
- Miller, P., Styles, R., and Heath, T. (2008). Open Data Commons, a License for Open Data. *Proceedings of the WWW2008 Workshop on Linked Data on the Web*, 369.
- Möller, E. (2006). Definition of Free Cultural Works. <http://freedomdefined.org/Definition>. Accessed: 2015-08-14.
- Museum Moderner Kunst Stiftung Ludwig Wien (2012). History — mumok. <https://www.mumok.at/en/history>. Accessed: 2015-10-10.
- National Human Genome Research Institute (1999). Availability of Genomic DNA Sequence (1999). <http://www.genome.gov/10001802>. Accessed: 2015-08-16.
- National Human Genome Research Institute (2012). Data Release and Access Principles and Policy. <http://www.genome.gov/10001801>. Accessed: 2015-08-16.

- Neumaier, S. (2015). *Open Data Quality Assessment and Evolution of (Meta-) Data Quality in the Open Data Landscape*. Diploma thesis, Vienna University of Technology.
- Open Data Portal Österreich (2014). FAQs - Häufig gestellte Fragen — OpenDataPortal Österreich. <https://www.opendataportal.at/faqs/>. Accessed: 2015-10-17.
- Open Knowledge Foundation (2008). About — Open Data Commons. <http://opendatacommons.org/about/>. Accessed: 2015-10-11.
- Open Knowledge Foundation (2015). JSON Table Schema - Data Protocols - Open Knowledge Foundation. <http://dataprotocols.org/json-table-schema/>. Accessed: 2015-07-12.
- Open Knowledge Foundation Austria (2015). Die neuen Regeln zum Weiterverwenden öffentlicher Dokumente — Open Knowledge Foundaiton Austria. <http://okfn.at/2015/07/17/die-neuen-regeln-zum-weiterverwenden-offentlicher-dokumente/>. Accessed: 2015-11-01.
- Opensource.org (2004). The Open Source Definition. <http://opensource.org/docs/osd>. Accessed: 2015-08-14.
- Polleres, A. and Steyskal, S. (2014). Semantic Web Standards for Publishing and Integrating Open Data. *Handbook of Research on Advanced ICT Integration for Governance and Policy Modeling*, pages 28–47.
- Pollock, R. (2014). Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge. <http://opendefinition.org/od/index.html>. Accessed: 2015-08-14.
- Schreiber, G. and Raimond, Y. (2014). RDF 1.1 Primer. <http://www.w3.org/TR/rdf11-primer/>. Accessed: 2015-10-18.
- Shafranovich, Y. (2005). Common Format and MIME Type for Comma-Separated Values (CSV) Files. <http://tools.ietf.org/html/rfc4180>. Accessed: 2015-10-18.
- Smithsonian American Art Museum (2014). About the Museum / American Art. <http://americanart.si.edu/visit/about/>. Accessed: 2016-01-04.
- Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., and Lindström, N. (2014). JSON-LD 1.0. <http://www.w3.org/TR/json-ld/>. Accessed: 2015-10-24.

- Sunlight Foundation (2010). Ten Principles for Opening Up Government Information. Technical report, Sunlight Foundation, <https://sunlightfoundation.com/policy/documents/ten-open-data-principles/>, Washington, DC.
- Szekely, P., Knoblock, C., and Yang, F. (2013). Connecting the Smithsonian American Art Museum to the Linked Data Cloud. *The Semantic Web: Semantics and Big Data*, 7882:593–607.
- Tauberer, J. (2014). Open Government Data Definition: The 8 Principles of Open Government Data - Open Government Data: The Book. <https://opengovdata.io/2014/8-principles/>. Accessed: 2015-11-21.
- Tennison, J., Kellogg, G., and Herman, I. (2015). Model for Tabular Data and Metadata on the Web. <http://www.w3.org/TR/tabular-data-model/>. Accessed: 2015-07-12.
- W3C (2015a). Data on the Web Best Practices. <http://w3c.github.io/dwbp/bp.html>. Accessed: 2015-09-25.
- W3C (2015b). W3C Mission. <http://www.w3.org/Consortium/mission>. Accessed: 2015-09-26.

Appendix A - Email conversation

From: Wolfert, Jörg joerg.wolfert@mumok.at
Subject: AW: Daten für Veröffentlichung als Open Data
Date: 5. Juni 2015 um 11:45
To: Winkler, Florian florian.winkler@mumok.at



Lieber Florian, ich denke das ist absolut in Ordnung, schließlich handelt es sich um öffentliche Daten, die, bis auf die Rechte der Abbildungen, jederzeit weitergegeben werden können,
lg
Jörg

Jörg Wolfert
Leitung Kunstvermittlung

mumok
museum moderner kunst stiftung ludwig wien
Museumsplatz 1
1070 Wien
T: +43-1-52500-1317
F: +43-1-52500-1300
E: joerg.wolfert@mumok.at
I: www.mumok.at

Rechtsform: Wissenschaftliche Anstalt öffentlichen Rechts, Sitz: Wien, FN: 225579i, HG Wien, UID: AT U16286908, DVR: 4009306
Bank: Erste Bank, BLZ 20111 Konto: 28742269802 IBAN: AT64 2011 1287 4226 9802 BIC: GIBAATWWXXX

-----Ursprüngliche Nachricht-----

Von: Winkler, Florian
Gesendet: Freitag, 05. Juni 2015 11:42
An: Wolfert, Jörg
Betreff: Daten für Veröffentlichung als Open Data

Hi,

Ich würde gerne im Rahmen meiner Bachelorarbeit an der WU Wien zum Thema "Open Cultural Data", die Daten der mumok Sammlung auf der öffentlichen Plattform opendataportal.at veröffentlichen.

Es werden die Daten nur zu jene Kunstwerken veröffentlicht, die für die Veröffentlichung in der Datenbank gekennzeichnet sind. Die Datenfelder für die Veröffentlichung wären die gleichen, die auch für die Online Sammlung verwendet werden (siehe Anhang) und Bilder werden nur per URL auf die bereits veröffentlichten Bilder auf der mumok Webseite <http://www.mumok.at> verlinkt.

Für die Verknüpfung Kunstwerke <-> Ausstellung werde ich mich noch an Susanne Neuburger wenden.

Die Daten sollen dann auch regelmäßig automatisch aktualisiert werden.

Wäre das für das mumok so in Ordnung?

Lg, Florian

Figure 7: Email conversation with Mag. Jörg Wolfert

From: Neuburger, Susanne susanne.neuburger@mumok.at
Subject: AW: Daten für Veröffentlichung als Open Data
Date: 10. Juni 2015 um 10:36
To: Winkler Florian florian.winkler@s.wu.ac.at

NS

Lieber Florian,

ist in Ordnung!
Liebe Grüße

Susi

Dr. Susanne Neuburger
Sammlungsleitung

mumok
museum moderner kunst stiftung ludwig wien
Museumsplatz 1
1070 Wien
T: +43-1-52500-1354
F: +43-1-52500-1300
E: susanne.neuburger@mumok.at
I: www.mumok.at

Rechtsform: Wissenschaftliche Anstalt öffentlichen Rechts, Sitz: Wien, FN: 225579i, HG Wien, UID: AT U16286908, DVR: 4009306
Bank: Erste Bank, BLZ 20111 Konto: 28742269802 IBAN: AT64 2011 1287 4226 9802 BIC: GIBAATWWXXX

-----Ursprüngliche Nachricht-----

Von: Winkler Florian [mailto:florian.winkler@s.wu.ac.at]
Gesendet: Mittwoch, 10. Juni 2015 10:24
An: Neuburger, Susanne
Betreff: Daten für Veröffentlichung als Open Data

Hallo,

Ich würde gerne im Rahmen meiner Bachelorarbeit an der WU Wien zum Thema "Open Cultural Data", die Daten der mumok Sammlung auf der öffentlichen Plattform opendataportal.at veröffentlichen.

Es werden die Daten nur zu jene Kunstwerken veröffentlicht, die für die Veröffentlichung in der Datenbank gekennzeichnet sind. Die Datenfelder für die Veröffentlichung wären die gleichen, die auch für die Online Sammlung verwendet werden (siehe Anhang) und Bilder werden nur per URL auf die bereits veröffentlichten Bilder auf der mumok Webseite <http://www.mumok.at> <<http://www.mumok.at>> verlinkt.

Von Jörg Wolfert habe ich dafür bereits das ok bekommen. Er hat gemeint ich solle mich noch melden wegen der Verknüpfung Kunstwerke <-> Ausstellung. Hier würde ich gerne zu jedem Werk die in der Datenbank vermerkten Stationen (Ausstellungsort, Titel, Datum Von-Bis) dazu veröffentlichen.

Die Daten sollen dann auch regelmäßig automatisch aktualisiert werden.

Wäre das für das mumok so in Ordnung?

Vielen Dank und Lg, Florian

Figure 8: Email conversation with Dr. Susanne Neuburger

Appendix B - Excel file with field mappings

Fieldname for export	Example value	Description	Field in database
OBJID	366	Object ID in database	ObjDaten.ObjID
INVENTORYNR	MAV 20/0	Inventory number	ObjDaten.ObjInventarNrS
INVENTORYNR_SORTED	MAV 00020/000	Inventory number sortable	ObjDaten.ObjInventarNrSortiertS
ORIGINALTITLE	Fusion	Original title of object	ObjDaten.ObjTitelOriginalS
TITLE_DE	Fusion	Title of object in German	(SELECT TOP (1) Omulnhalt01M FROM dbo.ObjMultiple AS ObjMultiple_1 WHERE (OmuObjId = dbo.ObjDaten.ObjId) AND (OmuGenFeldS = 'ObjTitelOriginalS') AND (OmuTypS = 'Titel (deutsch)'))
TITLE_EN	Fusion	Title of object in English	(SELECT TOP (1) Omulnhalt01M FROM dbo.ObjMultiple AS ObjMultiple_1 WHERE (OmuObjId = dbo.ObjDaten.ObjId) AND (OmuGenFeldS = 'ObjTitelOriginalS') AND (OmuTypS = 'Titel (englisch)'))
PRODUCTION_DATE	2001	Production date	ObjDaten.ObjDatierungS
IN_EXHIBITION	nein	Currently in exhibition?	(SELECT CASE WHEN dbo.ObjDaten.ObjStandortS LIKE '%Ebene%' THEN 'ja' ELSE 'nein' END AS Expr1)
MATERIAL_DE	Video auf DVD übertragen, Farbe, Ton, 15'15"	Used materials in German	(SELECT TOP (1) Omulnhalt01M FROM dbo.ObjMultiple AS ObjMultiple_1 WHERE (OmuObjId = dbo.ObjDaten.ObjId) AND (OmuGenFeldS = 'ObjTechMatM') AND (OmuTypS = 'Ausgabe'))
MATERIAL_EN	Video transferred to DVD, color, sound, 15'15"	Used materials in English	(SELECT TOP (1) Omulnhalt01M FROM dbo.ObjMultiple AS ObjMultiple_1 WHERE (OmuObjId = dbo.ObjDaten.ObjId) AND (OmuGenFeldS = 'ObjTechMatM') AND (OmuTypS = 'Ausgabe EN'))
MEASUREMENT_HEIGHT	80,5	Measurement of object - height	(SELECT TOP (1) ObmMasseHF FROM dbo.ObjMass AS ObjMass_1 WHERE (ObmObjId = dbo.ObjDatenZ2.ObjZ2Id) AND (ObmTypMasseS = 'Objektmaß') AND (ObmSortL = 1))
MEASUREMENT_WIDTH	81	Measurement of object - width	(SELECT TOP (1) ObmMasseBF FROM dbo.ObjMass AS ObjMass_1 WHERE (ObmObjId = dbo.ObjDatenZ2.ObjZ2Id) AND (ObmTypMasseS = 'Objektmaß') AND (ObmSortL = 1))
MEASUREMENT_DEPTH	85	Measurement of object - depth	(SELECT TOP (1) ObmMasseTF FROM dbo.ObjMass AS ObjMass_1 WHERE (ObmObjId = dbo.ObjDatenZ2.ObjZ2Id) AND (ObmTypMasseS = 'Objektmaß') AND (ObmSortL = 1))
MEASUREMENT_UNIT	cm	Unit of measurement (cm, qm2, ...)	(SELECT TOP (1) ObmMasseMS FROM dbo.ObjMass AS ObjMass_1 WHERE (ObmObjId = dbo.ObjDatenZ2.ObjZ2Id) AND (ObmTypMasseS = 'Objektmaß') AND (ObmSortL = 1))
DESCRIPTION_DE	Foto: Ludwig Hoffenreich Atelier Muehl, Perinetgasse 1, 1200 Wien	Textual description of object in German	(SELECT CAST(REPLACE(CAST(dbo.ObjDatenZ2.ObjC01M AS NVARCHAR(MAX)), CHAR(13) + CHAR(10), '') AS NTEXT) AS Expr1)
DESCRIPTION_EN	Photography: Ludwig Hoffenreich Muehl's Studio, Perinetgasse 1, 1200 Vienna	Textual description of object in English	(SELECT CAST(REPLACE(CAST(dbo.ObjDatenZ2.ObjC02M AS NVARCHAR(MAX)), CHAR(13) + CHAR(10), '') AS NTEXT) AS Expr1)
CREDITLINE	Geschenk Edelbert Köb/donation by Edelbert Köb	Creditline	ObjDaten.ObjFeld03M
ACQUIRY_DATE	2007	Acquiry date of the object	ObjDaten.ObjFeld04M
EXPLOITATION_RIGHTS	VBK Wien	Exploitation rights	(SELECT TOP (1) RchBeschriftungS FROM dbo.Recht AS Recht_1 WHERE (RchRefId = dbo.ObjDaten.ObjId) AND (RchArtS = 'Ausgabe Verwertungsrecht'))
DEFAULT_IMAGE	https://www.mumok.at/sites/default/files/styles/adaptive_gallery/public/cms/onlinesammlung/1382/P_687_0_Det_Web.jpg	HTTP link to the image on the mumok online collection website	'https://www.mumok.at/imageobject.php?objid=' + CAST(dbo.ObjDaten.ObjId AS varchar)

TAGS	Tschechien, Aktion, Performance, Gewalt, Politik, Tschechische Republik	Tags describing the object	ObjDatenZ2.ObjC03M
ARTIST_ID	365	Artist ID in database	Obj_Kuenstler.OkuKueId WHERE dbo.Obj_Kuenstler.OkuObjId = dbo.ObjDatenZ2.ObjZ2Id
ARTIST_FIRSTNAME	Siegfried	Firstname of the artist	Kuenstler.KueVorNameS
ARTIST_NAME	Anzinger	Lastname of the artist	Kuenstler.KueNameS
ARTIST_BIRTHYEAR	1953	Year of birth of the artist	Kuenstler.KueJahrVonL
ARTIST_DEATHYEAR		Year of death of the artist	Kuenstler.KueJahrBisL
ARTIST_CITYOFBIRTH_DE	Weyer, Österreich	Location of birth of the artist in German	Kuenstler.KueGeburtLandS
ARTIST_CITYOFBIRTH_EN	Weyer, Austria	Location of birth of the artist in English	Kuenstler.KueGeburtOrtS
ARTIST_CITYOFDEATH_DE		Location of death of the artist in German	Kuenstler.KueTodesLandS
ARTIST_CITYOFDEATH_EN		Location of the death of the artist in English	Kuenstler.KueTodesOrtS
EXHIBITION_ID	3645	Exhibition ID in database	Ausstellung_Obj.AobAusId WHERE Ausstellung_Obj.AobObjId = ObjDatenZ2.ObjZ2Id
EXHIBITION_TITLE	Eva Aeppli und Daniel Spoerri-Erst letzt das Erste	Title of the exhibition	Ausstellung.AusTitelS
EXHIBITION_CITY	Hadersdorf am Kamp	City of the exhibition	Ausstellung.AusArtS
EXHIBITION_LOCATION	Kunststaulager Spoerri GmbH & Co. KG	Location of the exhibition	Ausstellung.AusOrtS
EXHIBITION_DATEFROM	20.03.2010	Starting date of the exhibition	Ausstellung.AusDatumVonD
EXHIBITION_DATEUNTIL	15.10.2010	End date of the exhibition	Ausstellung.AusDatumBisD
		The record may be published on the web.	ObjC04S = 'ja'

Appendix C - Python scripts for data transformation

Script 'sql_to_csv.py' for converting SQL data to CSV files.

```
# -*- coding: utf-8 -*-
import sys
import pyodbc
import datetime
import io
import csv
import codecs

def convert_datetime_to_string(row):
    for i in range(len(row)):
        if type(row[i]) == datetime.datetime:
            row[i] = row[i].strftime('%Y-%m-%d')
        elif type(row[i]) == unicode:
            row[i] = row[i].encode("utf-8")
    return row

def write_csv(out_file, sql_select):

    cursor.execute(sql_select)
    column_names = []
    for column in cursor.description:
        column_names.append(column[0].lower())

    data_out = open(out_file, "w")
    data_out.write(codecs.BOM_UTF8)
    data_out.close()
    data_out = open(out_file, 'ab')
    mywriter = csv.writer(data_out)
    mywriter.writerow(column_names)
    for row in cursor:
        row_clean = convert_datetime_to_string(row)
        mywriter.writerow(row)
    data_out.close()

conn = pyodbc.connect(r'DSN=mpdb;uid=opendata;pwd=*****;APP=OpenDataExporter')
cursor = conn.cursor()

write_csv('~/opendata/obj.csv', 'select * from Open_Data')
write_csv('~/opendata/obj_exhibition.csv',
'SELECT Ausstellung_Obj.AobId AS id, '
'Ausstellung_Obj.AobObjId AS objid, '
'Ausstellung_Obj.AobAusId AS exhibition_id '
'FROM Ausstellung_Obj '
'INNER JOIN '
'Ausstellung ON Ausstellung_Obj.AobAusId = Ausstellung.AusId '
'INNER JOIN '
'ObjDatenZ2 ON Ausstellung_Obj.AobObjId = ObjDatenZ2.ObjZ2Id '
'WHERE (ObjDatenZ2.ObjC04S = N\'ja\') '
'ORDER BY Ausstellung_Obj.AobId'
)

write_csv('~/opendata/exhibition.csv',
'SELECT Ausstellung.AusId AS exhibition_id, '
'Ausstellung.AusArtS AS city, '
'Ausstellung.AusTitelS AS title, '
'Ausstellung.AusOrtS AS location, '
'Ausstellung.AusDatumVonD AS datefrom, '
'Ausstellung.AusDatumBisD AS dateuntil '
'FROM Ausstellung_Obj '
'INNER JOIN '
'Ausstellung ON Ausstellung_Obj.AobAusId = Ausstellung.AusId '
'INNER JOIN '
'ObjDatenZ2 ON Ausstellung_Obj.AobObjId = ObjDatenZ2.ObjZ2Id '
'WHERE (ObjDatenZ2.ObjC04S = N\'ja\') '
'GROUP BY Ausstellung.AusId, '
'Ausstellung.AusArtS, '
'Ausstellung.AusTitelS, '
```

```

        'Ausstellung.AusOrtS, '
        'Ausstellung.AusDatumVond, '
        'Ausstellung.AusDatumBisD '
    'ORDER BY exhibition_id'
)
write_csv('~/opendata/obj_artist.csv',
'SELECT Obj_Kuenstler.OkuId AS id, '
'Obj_Kuenstler.OkuObjId AS objid, '
'Obj_Kuenstler.OkuKueId AS artist_id '
'FROM Obj_Kuenstler '
'INNER JOIN '
'Kuenstler ON Obj_Kuenstler.OkuKueId = Kuenstler.KueId '
'INNER JOIN '
'ObjDatenZ2 ON Obj_Kuenstler.OkuObjId = ObjDatenZ2.ObjZ2Id '
'WHERE (ObjDatenZ2.ObjC04S = N\'ja\') '
'ORDER BY Obj_Kuenstler.OkuId'
)
write_csv('~/opendata/artist.csv',
'SELECT Kuenstler.KueId AS artist_id, '
'Kuenstler.KueVorNameS AS firstname, '
'Kuenstler.KueNameS AS name, '
'Kuenstler.KueGeburtLandS AS cityofbirth_de, '
'Kuenstler.KueGeburtOrtS AS cityofbirth_en, '
'Kuenstler.KueTodesLandS AS cityofdeath_de, '
'Kuenstler.KueTodesOrtS AS cityofdeath_en, '
'Kuenstler.KueJahrVonL AS birthyear, '
'Kuenstler.KueJahrBisL AS deathyear '
'FROM Obj_Kuenstler '
'INNER JOIN '
'Kuenstler ON Obj_Kuenstler.OkuKueId = Kuenstler.KueId '
'INNER JOIN '
'ObjDatenZ2 ON Obj_Kuenstler.OkuObjId = ObjDatenZ2.ObjZ2Id '
'WHERE (ObjDatenZ2.ObjC04S = N\'ja\') '
'GROUP BY Kuenstler.KueId, '
'Kuenstler.KueVorNameS, '
'Kuenstler.KueNameS, '
'Kuenstler.KueGeburtLandS, '
'Kuenstler.KueGeburtOrtS, '
'Kuenstler.KueTodesLandS, '
'Kuenstler.KueTodesOrtS, '
'Kuenstler.KueJahrVonL, '
'Kuenstler.KueJahrBisL '
'ORDER BY Kuenstler.KueId'
)

```


Script 'sql_to_json.py' for converting SQL data to JSON file.

```
# -*- coding: utf-8 -*-
import sys
import pyodbc
import json
import datetime
import io

def convert_datetime_to_string(row):
    for i in range(len(row)):
        if type(row[i]) == datetime.datetime:
            row[i] = row[i].strftime('%Y-%m-%d')
    return row

def insertMetadata():
    metadata = {
        "@context": ["http://www.w3.org/ns/csvw", {"@language": "en"}],
        "url": "http://data.mumok.at/obj.json",
        "dc:title": "Objects from the mumok collection together with their artists and
        exhibitions",
        "dcat:keyword": ["art", "museum"],
        "dc:publisher": {
            "schema:name": "mumok - museum moderner kunst stiftung ludwig wien",
            "schema:url": {"@id": "http://www.mumok.at"}
        },
        "dc:license": {"@id": "https://creativecommons.org/licenses/by/3.0/at/legalcode"},
        "dc:modified": {"@value": "2015-07-04", "@type": "xsd:date"},
        "tableSchema": {
            "columns": [{
                "name": "objid",
                "titles": ["OBJID", "Object Identifier"],
                "dc:description": "An identifier for the object.",
                "datatype": "integer",
                "required": "true"
            }, {
                "name": "inventorynr",
                "titles": "Inventory number",
                "dc:description": "The inventory number of the object.",
                "datatype": "string"
            }, {
                "name": "inventorynr_sorted",
                "titles": "sortable inventory number",
                "dc:description": "The inventory number of the object, sortable by prefixed zeroes
                to get a fixed length.",
                "datatype": "string"
            }, {
                "name": "originaltitle",
                "titles": "Originaltitle",
                "dc:description": "The original title of the object as given by the creator.",
                "datatype": "string"
            }, {
                "name": "title_de",
                "titles": "Title in German",
                "dc:description": "The title of the object translated to German.",
                "datatype": "string",
                "lang": "de"
            }, {
                "name": "title_en",
                "titles": "Title in English",
                "dc:description": "The title of the object translated to English.",
                "datatype": "string",
                "lang": "en"
            }, {
                "name": "production_date",
                "titles": "Date of production",
                "dc:description": "The date on which the object has been produced (informal value:
                year or range of years)",
                "datatype": "string"
            }, {
                "name": "in_exhibition",
                "titles": "Object in exhibition",
                "dc:description": "States if the object is currently on display in an exhibition (
                possible values: 'yes'/'no').",
                "datatype": "boolean"
            }
        ]
    }
```

```

}, {
  "name": "material_de",
  "titles": "Used materials in German",
  "dc:description": "The materials the object is made of in German.",
  "datatype": "string",
  "lang": "de"
}, {
  "name": "material_en",
  "titles": "Used materials in English",
  "dc:description": "The materials the object is made of in English.",
  "datatype": "string",
  "lang": "en"
}, {
  "name": "measurement_height",
  "titles": "Measurement of height",
  "dc:description": "The height measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_width",
  "titles": "Measurement of width",
  "dc:description": "The width measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_depth",
  "titles": "Measurement of depth",
  "dc:description": "The depth measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_unit",
  "titles": "Measurement unit",
  "dc:description": "The unit of measurement for the given measurements ('
    measurement_height', 'measurement_width', measurement_depth').",
  "datatype": "string"
}, {
  "name": "description_de",
  "titles": "Description in German",
  "dc:description": "The description of the object in German.",
  "datatype": "string",
  "lang": "de"
}, {
  "name": "description_en",
  "titles": "Description in English",
  "dc:description": "The description of the object in English.",
  "datatype": "string",
  "lang": "en"
}, {
  "name": "creditline",
  "titles": "Creditline",
  "dc:description": "The creditline states the owner or loaning party of the object
    (usually in format: 'German creditline'/'English creditline').",
  "datatype": "string"
}, {
  "name": "acquiry_date",
  "titles": "Date of acquiry",
  "dc:description": "The year in which the object has been acquired by the museum.",
  "datatype": {"base": "date", "format": "yyyy"}
}, {
  "name": "exploitation_rights",
  "titles": "Exploitation rights",
  "dc:description": "The owner of the object's exploitation rights.",
  "datatype": "string"
}, {
  "name": "default_image",
  "titles": "Default Image",
  "dc:description": "The URL for the default image being set by the data operator.",
  "datatype": "string"
}, {
  "name": "tags",
  "titles": "Tags",
  "dc:description": "The tags that were used to classify the object.",
  "datatype": "string"
}, {

```

```

"name": "exhibitions",
"titles": "Exhibitions",
"dc:description": "Exhibitions in which the object has been presented.",
"datatype": "object",
"columns": [
  {
    "name": "exhibition_id",
    "titles": "Exhibition Identifier",
    "dc:description": "A unique identifier for the exhibition.",
    "datatype": "integer",
    "required": "true"
  }, {
    "name": "city",
    "titles": "City",
    "dc:description": "The city in which the exhibition took place (no language defined, mostly in German).",
    "datatype": "string"
  }, {
    "name": "title",
    "titles": "Exhibition Title",
    "dc:description": "The title of the exhibition.",
    "datatype": "string"
  }, {
    "name": "location",
    "titles": "Location of exhibition",
    "dc:description": "The location at which the exhibition took place (no language defined, mostly in German).",
    "datatype": "string"
  }, {
    "name": "datefrom",
    "titles": "Exhibition starting date",
    "dc:description": "The date on which the exhibition has started.",
    "datatype": "string"
  }, {
    "name": "dateuntil",
    "titles": "Exhibition ending date",
    "dc:description": "The date on which the exhibition has ended.",
    "datatype": "string"
  }
],
"primary_key": "exhibition_id"
}, {
  "name": "artist",
  "titles": "Artists",
  "dc:description": "Artists that have been involved with the creation of the object",
  "datatype": "object",
  "columns": [
    {
      "name": "artist_id",
      "titles": "Artist Identifier",
      "dc:description": "A unique identifier for the artist.",
      "datatype": "integer",
      "required": "true"
    }, {
      "name": "firstname",
      "titles": "Firstname",
      "dc:description": "The firstname of the artist.",
      "datatype": "string"
    }, {
      "name": "name",
      "titles": "Name (Lastname)",
      "dc:description": "The lastname of the artist.",
      "datatype": "string"
    }, {
      "name": "cityofbirth_de",
      "titles": "City of birth in German",
      "dc:description": "The city in which the artist has been born, in German.",
      "datatype": "string",
      "lang": "de"
    }, {
      "name": "cityofbirth_en",
      "titles": "City of birth in English",
      "dc:description": "The city in which the artist has been born, in English.",
      "datatype": "string",
      "lang": "en"
    }
  ]
}

```

```

    }, {
        "name": "cityofdeath_de",
        "titles": "City of death in German",
        "dc:description": "The city in which the artist has died, in German.",
        "datatype": "string",
        "lang": "de"
    }, {
        "name": "cityofbirth_en",
        "titles": "City of death in English",
        "dc:description": "The city in which the artist has died, in English",
        "datatype": "string",
        "lang": "en"
    }, {
        "name": "birthyear",
        "titles": "Artist's birthyear",
        "dc:description": "The year in which the artist has been born.",
        "datatype": {"base": "date", "format": "yyyy"}
    }, {
        "name": "deathyear",
        "titles": "Artist's deathyear",
        "dc:description": "The year in which the artist has died.",
        "datatype": {"base": "date", "format": "yyyy"}
    }
    ],
    "primaryKey": "artist_id"
}
],
"primaryKey": "objid"
}
}

return metadata

def append_to_existing_json_object(row):
    results[row.OBJID]['exhibitions'][str(row.EXHIBITION_ID)] = {
        'city' : row.EXHIBITION_CITY,
        'datefrom' : row.EXHIBITION_DATEFROM,
        'dateuntil' : row.EXHIBITION_DATEUNTIL,
        'location' : row.EXHIBITION_LOCATION,
        'title' : row.EXHIBITION_TITLE,
    }
    results[row.OBJID]['artists'][str(row.ARTIST_ID)] = {
        'birthyear' : row.ARTIST_BIRTHYEAR,
        'cityofbirth_de' : row.ARTIST_CITYOFBIRTH_DE,
        'cityofbirth_en' : row.ARTIST_CITYOFBIRTH_EN,
        'cityofdeath_de' : row.ARTIST_CITYOFDEATH_DE,
        'cityofdeath_en' : row.ARTIST_CITYOFDEATH_EN,
        'deathyear' : row.ARTIST_DEATHYEAR,
        'firstname' : row.ARTIST_FIRSTNAME,
        'name' : row.ARTIST_NAME,
    }
}

def create_dict_from_row(row):
    row_dict = {
        'acquiry_date' : row.ACQUIRY_DATE,
        'creditline' : row.CREDITLINE,
        'default_image' : row.DEFAULT_IMAGE,
        'description_de' : row.DESCRPTION_DE,
        'description_en' : row.DESCRPTION_EN,
        'exploitation_rights' : row.EXPLOITATION_RIGHTS,
        'inventorynr' : row.INVENTORYNR,
        'inventorynr_sorted' : row.INVENTORYNR_SORTED,
        'in_exhibition' : row.IN_EXHIBITION,
        'material_de' : row.MATERIAL_DE,
        'material_en' : row.MATERIAL_EN,
        'measurement_depth' : row.MEASUREMENT_DEPTH,
        'measurement_height' : row.MEASUREMENT_HEIGHT,
        'measurement_unit' : row.MEASUREMENT_UNIT,
        'measurement_width' : row.MEASUREMENT_WIDTH,
        'originaltitle' : row.ORIGINALTITLE,
        'production_date' : row.PRODUCTION_DATE,
        'tags' : row.TAGS,
        'title_de' : row.TITLE_DE,
        'title_en' : row.TITLE_EN,
        'artists' : {
            str(row.ARTIST_ID): {

```

```

        'birthyear' : row.ARTIST_BIRTHYEAR,
        'cityofbirth_de' : row.ARTIST_CITYOFBIRTH_DE,
        'cityofbirth_en' : row.ARTIST_CITYOFBIRTH_EN,
        'cityofdeath_de' : row.ARTIST_CITYOFDEATH_DE,
        'cityofdeath_en' : row.ARTIST_CITYOFDEATH_EN,
        'deathyear' : row.ARTIST_DEATHYEAR,
        'firstname' : row.ARTIST_FIRSTNAME,
        'name': row.ARTIST_NAME,
    },
},
'exhibitions' : {
    str(row.EXHIBITION_ID): {
        'city' : row.EXHIBITION_CITY,
        'datefrom' : row.EXHIBITION_DATEFROM,
        'dateuntil' : row.EXHIBITION_DATEUNTIL,
        'location' : row.EXHIBITION_LOCATION,
        'title' : row.EXHIBITION_TITLE,
    },
},
}
return row_dict

conn = pyodbc.connect(r'DSN=mpdb;uid=opendata;pwd=*****;APP=OpenDataExporter')
cursor = conn.cursor()
cursor.execute('SELECT * FROM Open_Data_Full ORDER BY OBJID')
#column_names = []
#for column in cursor.description:
# column_names.append(column[0])
results = {}
last_objid = int()
for row in cursor:
    row_cleaned = convert_datetime_to_string(row)
    #if it comes from a joined row on the same objectid - append it to the object with same
    #objid
    if row.OBJID == last_objid:
        append_to_existing_json_object(row_cleaned)
    #else - create a new json object
    else:
        results[row.OBJID] = (create_dict_from_row(row_cleaned))
        last_objid = row.OBJID

output={}
output['@data'] = results
output['@metadata'] = insertMetadata()

with io.open('~\opendata\data.json', 'w', encoding='utf-8') as f:
    f.write(unicode(json.dumps(output, sort_keys=True, indent=4, ensure_ascii=False)))

```

Script 'sql_to_json-ld.py' for converting SQL data to JSON-LD file.

```
# -*- coding: utf-8 -*-
import sys
import pyodbc
import MySQLdb as mdb
import json
import datetime
import io

def queryUrls(OBJID):
    url = {}
    cursor = mdbcon.cursor()
    cursor.execute("SELECT CONCAT ('http://www.mumok.at/node/', entity_id_de) FROM
        mumok_collection_import_objects WHERE objid = '"+str(OBJID)+"'")
    if cursor.rowcount != 1:
        url['object'] = '_o:'+str(OBJID)
    else:
        url['object'] = cursor.fetchone()[0]

    cursor.execute("SELECT CONCAT ('http://www.mumok.at/node/', entity_id_de) FROM
        mumok_collection_import_artists WHERE objid = '"+str(OBJID)+"'")
    if cursor.rowcount != 1:
        if cursor.rowcount > 1:
            cursor.execute("SELECT CONCAT ('http://www.mumok.at/node/', entity_id_de) FROM
                mumok_collection_import_artists \
                WHERE objid = '"+str(OBJID)+"' AND vorname = '"+('' if row.ARTIST_FIRSTNAME
                    is None else row.ARTIST_FIRSTNAME)+"' AND nachname = '"+('' if row.
                    ARTIST_NAME is None else row.ARTIST_NAME)+"'")
            if cursor.rowcount != 1:
                url['artist'] = '_a:'+str(row.ARTIST_ID)
            elif cursor.rowcount == 1:
                url['artist'] = cursor.fetchone()[0]
            else:
                url['artist'] = '_a:'+str(row.ARTIST_ID)
        elif cursor.rowcount == 1:
            url['artist'] = cursor.fetchone()[0]
        else:
            url['artist'] = '_a:'+str(row.ARTIST_ID)
    return url

def convert_datetime_to_string(row):
    for i in range(len(row)):
        if type(row[i]) == datetime.datetime:
            row[i] = row[i].strftime('%Y-%m-%d')
    return row

def append_to_existing_json_object(row, urls):
    exhibition_exists = False
    for exhibition in tmpRecord['exhibitions']:
        if exhibition['@id'] == '_e:'+str(row.EXHIBITION_ID):
            exhibition_exists = True

    if not exhibition_exists:
        tmpRecord['exhibitions'].append({
            '@id': '_e:'+str(row.EXHIBITION_ID),
            '@type': 'dc:Event',
            'city': row.EXHIBITION_CITY,
            'datefrom': row.EXHIBITION_DATEFROM,
            'dateuntil': row.EXHIBITION_DATEUNTIL,
            'location': row.EXHIBITION_LOCATION,
            'title': row.EXHIBITION_TITLE,
        })

    artist_exists = False
    for artist in tmpRecord['artists']:
        if artist['@id'] == urls['artist']:
            artist_exists = True

    if not artist_exists:
        tmpRecord['artists'].append({
            '@id': urls['artist'],
            '@type': 'Person',
            'birthyear': row.ARTIST_BIRTHYEAR,
            'cityofbirth_de': row.ARTIST_CITYOFBIRTH_DE,
            'cityofbirth_en': row.ARTIST_CITYOFBIRTH_EN,
            'cityofdeath_de': row.ARTIST_CITYOFDEATH_DE,
```

```

        'cityofdeath_en' : row.ARTIST_CITYOFDEATH_EN,
        'deathyear' : row.ARTIST_DEATHYEAR,
        'firstname' : row.ARTIST_FIRSTNAME,
        'familyname': row.ARTIST_NAME,
    })
def create_dict_from_row(row, urls):
    row_dict = {
        '@id' : urls['object'],
        'acquiry_date' : row.ACQUIRY_DATE,
        'creditline' : row.CREDITLINE,
        'default_image' : row.DEFAULT_IMAGE,
        'description_de' : row.DESCRPTION_DE,
        'description_en' : row.DESCRPTION_EN,
        'exploitation_rights' : row.EXPLOITATION_RIGHTS,
        'inventorynr' : row.INVENTORYNR,
        'inventorynr_sorted' : row.INVENTORYNR_SORTED,
        'in_exhibition' : row.IN_EXHIBITION,
        'material_de' : row.MATERIAL_DE,
        'material_en' : row.MATERIAL_EN,
        'measurement_depth' : row.MEASUREMENT_DEPTH,
        'measurement_height' : row.MEASUREMENT_HEIGHT,
        'measurement_unit' : row.MEASUREMENT_UNIT,
        'measurement_width' : row.MEASUREMENT_WIDTH,
        'originaltitle' : row.ORIGINALTITLE,
        'production_date' : row.PRODUCTION_DATE,
        'tags' : row.TAGS,
        'title_de' : row.TITLE_DE,
        'title_en' : row.TITLE_EN,
        'artists' : [{
            '@id' : urls['artist'],
            '@type' : 'Person',
            'birthyear' : row.ARTIST_BIRTHYEAR,
            'cityofbirth_de' : row.ARTIST_CITYOFBIRTH_DE,
            'cityofbirth_en' : row.ARTIST_CITYOFBIRTH_EN,
            'cityofdeath_de' : row.ARTIST_CITYOFDEATH_DE,
            'cityofdeath_en' : row.ARTIST_CITYOFDEATH_EN,
            'deathyear' : row.ARTIST_DEATHYEAR,
            'firstname' : row.ARTIST_FIRSTNAME,
            'familyname': row.ARTIST_NAME,
        }],
        'exhibitions' : [{
            '@id' : '_e:'+str(row.EXHIBITION_ID),
            '@type' : 'dc:Event',
            'city' : row.EXHIBITION_CITY,
            'datefrom' : row.EXHIBITION_DATEFROM,
            'dateuntil' : row.EXHIBITION_DATEUNTIL,
            'location' : row.EXHIBITION_LOCATION,
            'title' : row.EXHIBITION_TITLE,
        }],
    }
    return row_dict

#open connection to Museumplus database
conn = pyodbc.connect(r'DSN=mpdb;uid=opendata;pwd=*****;APP=OpenDataExporter')
cursor = conn.cursor()
cursor.execute('select * from Open_Data_Full ORDER BY OBJID')
column_names = []
for column in cursor.description:
    column_names.append(column[0])

#open connection to webserver database to get the urls for objects and artists
try:
    mdbcon = mdb.connect('transfer.mumok.at', 'dbuser', '*****', 'dbname', charset='utf8')
    cur = mdbcon.cursor()
except mdb.Error, e:
    print "Error %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)

results = {}
results['@context'] = {

```

```

    "@vocab": "http://schema.org/",
    "dc": "http://purl.org/dc/elements/1.1/",
    "dc:license": {"@id": "https://creativecommons.org/licenses/by/3.0/at/legalcode"},
    "url": "@id",
    "acquiry_date": "https://schema.org/copyrightYear",
    "creditline": "dc:Provenance",
    "default_image": "thumbnailUrl",
    "description_de": { "@id": "description", "@language": "de" },
    "description_en": { "@id": "description", "@language": "en" },
    "exploitation_rights": "dc:RightsHolder",
    "inventorynr": "dc:identifier",
    "inventorynr_sorted": "dc:identifier",
    "material_de": { "@id": "artMedium", "@language": "de" },
    "material_en": { "@id": "artMedium", "@language": "en" },
    "measurement_depth": { "@id": "depth", "@type": "QuantitativeValue" },
    "measurement_height": { "@id": "height", "@type": "QuantitativeValue" },
    "measurement_width": { "@id": "width", "@type": "QuantitativeValue" },
    "originaltitle": "name",
    "production_date": "dateCreated",
    "tags": "keywords",
    "title_de": { "@id": "alternateName", "@language": "de" },
    "title_en": { "@id": "alternateName", "@language": "en" },
    "birthyear": "birthDate",
    "cityofbirth_de": { "@id": "birthPlace", "@language": "de" },
    "cityofbirth_en": { "@id": "birthPlace", "@language": "en" },
    "cityofdeath_de": { "@id": "deathPlace", "@language": "de" },
    "cityofdeath_en": { "@id": "deathPlace", "@language": "en" },
    "deathyear": "deathDate",
    "firstname": "givenName",
    "name": "familyName",
    "city": {"@type": "Place"}
}
results['@set'] = []
last_objid = 0
for row in cursor:
    row_cleaned = convert_datetime_to_string(row)
    urls = queryUrls(row.OBJID)
    #if it comes from a joined row on the same objectid - append it to the object with same
    objid
    if row.OBJID == last_objid:
        append_to_existing_json_object(row_cleaned, urls)
    #else - create a new json object
    else:
        if last_objid == 0:
            tmpRecord = (create_dict_from_row(row_cleaned, urls))
        else:
            results['@set'].append(tmpRecord)
            tmpRecord = (create_dict_from_row(row_cleaned, urls))
        last_objid = row.OBJID

#for the last record to be added
results['@set'].append(tmpRecord)

with io.open('~/.opendata/data.jsonld', 'w', encoding='utf-8') as f:
    f.write(unicode(json.dumps(results, sort_keys=False, indent=4, ensure_ascii=False)))

```


Appendix D - Metadata file for CSV

An example of a metadata file describing the content of the CSV file 'obj.csv' containing the art objects in JSON format.

```
{
  "@context": ["http://www.w3.org/ns/csvw", {"@language": "en"}],
  "url": "http://data.mumok.at/data.csv",
  "dc:title": "Objects from the mumok collection",
  "dcat:keyword": ["art", "museum"],
  "dc:publisher": {
    "schema:name": "mumok - museum moderner kunst stiftung ludwig wien",
    "schema:url": {"@id": "http://www.mumok.at"}
  },
  "dc:license": {"@id": "https://creativecommons.org/licenses/by/3.0/at/legalcode"},
  "dc:modified": {"@value": "2015-07-04", "@type": "xsd:date"},
  "dialect": {
    "encoding": "utf-8",
    "lineTerminators": ["\r\n", "\n"],
    "quoteChar": "\"",
    "doubleQuote": true,
    "skipRows": 0,
    "commentPrefix": "#",
    "header": true,
    "headerRowCount": 1,
    "delimiter": ",",
    "skipColumns": 0,
    "skipBlankRows": false,
    "skipInitialSpace": false,
    "trim": false
  },
  "tableSchema": {
    "columns": [{
      "name": "objid",
      "titles": ["OBJID", "Object Identifier"],
      "dc:description": "An identifier for the object.",
      "datatype": "integer",
      "required": true
    }, {
      "name": "inventorynr",
      "titles": "Inventory number",
      "dc:description": "The inventory number of the object.",
      "datatype": "string"
    }, {
      "name": "inventorynr_sorted",
      "titles": "sortable inventory number",
      "dc:description": "The inventory number of the object, sortable by prefixed zeroes to get a fixed length.",
      "datatype": "string"
    }, {
      "name": "originaltitle",
      "titles": "Originaltitle",
      "dc:description": "The original title of the object as given by the creator.",
      "datatype": "string"
    }, {
      "name": "title_de",
      "titles": "Title in German",
      "dc:description": "The title of the object translated to German.",
      "datatype": "string",
      "lang": "de"
    }, {
      "name": "title_en",
      "titles": "Title in English",
      "dc:description": "The title of the object translated to English.",
      "datatype": "string",
      "lang": "en"
    }, {
      "name": "production_date",
      "titles": "Date of production",
      "dc:description": "The date on which the object has been produced (informal value: year or range of years)",
      "datatype": "string"
    }
  ]
}
```

```

}, {
  "name": "in_exhibition",
  "titles": "Object in exhibition",
  "dc:description": "States if the object is currently on display in an exhibition (
    possible values: 'yes'/'no').",
  "datatype": "boolean"
}, {
  "name": "material_de",
  "titles": "Used materials in German",
  "dc:description": "The materials the object is made of in German.",
  "datatype": "string",
  "lang": "de"
}, {
  "name": "material_en",
  "titles": "Used materials in English",
  "dc:description": "The materials the object is made of in English.",
  "datatype": "string",
  "lang": "en"
}, {
  "name": "measurement_height",
  "titles": "Measurement of height",
  "dc:description": "The height measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_width",
  "titles": "Measurement of width",
  "dc:description": "The width measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_depth",
  "titles": "Measurement of depth",
  "dc:description": "The depth measurement of the object (unit can be deducted from
    column 'measurement_unit').",
  "datatype": "double"
}, {
  "name": "measurement_unit",
  "titles": "Measurement unit",
  "dc:description": "The unit of measurement for the given measurements ('
    measurement_height', 'measurement_width', measurement_depth').",
  "datatype": "string"
}, {
  "name": "description_de",
  "titles": "Description in German",
  "dc:description": "The description of the object in German.",
  "datatype": "string",
  "lang": "de"
}, {
  "name": "description_en",
  "titles": "Description in English",
  "dc:description": "The description of the object in English.",
  "datatype": "string",
  "lang": "en"
}, {
  "name": "creditline",
  "titles": "Creditline",
  "dc:description": "The creditline states the owner or loaning party of the object (
    usually in format: 'German creditline'/'English creditline').",
  "datatype": "string"
}, {
  "name": "acquiry_date",
  "titles": "Date of acquiry",
  "dc:description": "The year in which the object has been acquired by the museum.",
  "datatype": {"base": "date", "format": "yyyy"}
}, {
  "name": "exploitation_rights",
  "titles": "Exploitation rights",
  "dc:description": "The owner of the object's exploitation rights.",
  "datatype": "string"
}, {
  "name": "default_image",
  "titles": "Default Image",
  "dc:description": "The URL for the default image being set by the data operator.",
  "datatype": "string"

```

```
}, {  
  "name": "tags",  
  "titles": "Tags",  
  "dc:description": "The tags that were used to classify the object.",  
  "datatype": "string"  
}],  
  "primaryKey": "objid"  
}  
}
```

Appendix E - Python script: CKAN API access

A python script 'ckan_api_update.py' that updates the modification time and file size of the published files on the Open Data portal.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import urllib2
import urllib
import json
import pprint
import datetime
import os

last_modified = datetime.datetime.now().isoformat()

# Put the details of the dataset we're going to create into a dict.
dataset_dict = {
    'ausstellungen-der-sammlung-des-mumok_csv': {
        'id': '27ff4b2b-2d2b-4db6-897e-4e4ae927d933',
        'url': 'http://data.mumok.at/exhibition.csv',
        'last_modified': last_modified },
    'ausstellungen-der-sammlung-des-mumok_csv-metadata': {
        'id': '9f951616-fef7-44aa-8b13-27e7fba090fd',
        'url': 'http://data.mumok.at/exhibition.csv-metadata.json',
        'last_modified': last_modified },
    'künstler-der-sammlung-des-mumok_csv': {
        'id': 'db43a903-0e58-41b6-af4b-adb9710ac4d8',
        'url': 'http://data.mumok.at/artist.csv',
        'last_modified': last_modified },
    'künstler-der-sammlung-des-mumok_csv-metadata': {
        'id': 'e25640f8-a3e4-46d2-8a4f-9be471b115d2',
        'url': 'http://data.mumok.at/artist.csv-metadata.json',
        'last_modified': last_modified },
    'kunstwerke-der-sammlung-des-mumok_json': {
        'id': '98f5e536-02f4-4471-8f2f-3881171f13ee',
        'url': 'http://data.mumok.at/data.json',
        'last_modified': last_modified },
    'kunstwerke-der-sammlung-des-mumok_json-ld': {
        'id': '7428a2fd-df14-4ca3-bf6f-f7ae36674f8d',
        'url': 'http://data.mumok.at/data.jsonld',
        'last_modified': last_modified },
    'kunstwerke-der-sammlung-mumok_csv': {
        'id': 'cc5f337f-1f79-4b82-88e3-cdaaa7c4440e',
        'url': 'http://data.mumok.at/obj.csv',
        'last_modified': last_modified },
    'kunstwerke-der-sammlung-mumok_csv-metadata': {
        'id': 'f81dbfcc-c7da-471b-8f4e-dc5d5609b424',
        'url': 'http://data.mumok.at/obj.csv-metadata.json',
        'last_modified': last_modified },
    'mapping-zwischen-kunstwerk-und-ausstellung-der-sammlung-des-mumok_csv': {
        'id': '4bd77383-d870-4493-97bf-69eb95da8a9f',
        'url': 'http://data.mumok.at/obj_exhibition.csv',
        'last_modified': last_modified },
    'mapping-zwischen-kunstwerk-und-ausstellung-der-sammlung-des-mumok_csv-metadata': {
        'id': '18f8c02f-f266-43f7-813f-47820d38dedd',
        'url': 'http://data.mumok.at/obj_exhibition.csv-metadata.json',
        'last_modified': last_modified },
    'mapping-zwischen-kunstwerk-und-kunstler-der-sammlung-des-mumok_csv': {
        'id': '86257835-66ec-4f5c-a9cc-8721faf57dc6',
        'url': 'http://data.mumok.at/obj_artist.csv',
        'last_modified': last_modified },
    'mapping-zwischen-kunstwerk-und-kunstler-der-sammlung-des-mumok_csv-metadata': {
        'id': 'ae5044fa-2a39-405f-bcce-b2d3709a62c5',
        'url': 'http://data.mumok.at/obj_artist.csv-metadata.json',
        'last_modified': last_modified }
}

# get the file size of the csv and json files and put it in the dataset_dict
```

```

for dataset in dataset_dict:
    file = os.path.basename(dataset_dict[dataset]['url'])
    dataset_dict[dataset]['size'] = os.stat('~/opendata/'+file).st_size
    #print dataset_dict[dataset]

#loop through the dataset_dict and make the API call to CKAN to set the updated information
#on the resouces
for dataset in dataset_dict:
    # Use the json module to dump the dictionary to a string for posting.
    data_string = urllib.quote(json.dumps(dataset_dict[dataset]))

    #print data_string
    # We'll use the package_create function to create a new dataset.
    request = urllib2.Request('http://data.opendataportal.at/api/3/action/resource_update')

    # Creating a dataset requires an authorization header.
    # Replace *** with your API key, from your user account on the CKAN site
    # that you're creating the dataset on.
    request.add_header('Authorization', '***')

    # Make the HTTP request.
    response = urllib2.urlopen(request, data_string)
    assert response.code == 200

    # Use the json module to load CKAN's response into a dictionary.
    response_dict = json.loads(response.read())
    assert response_dict['success'] is True

    # package_create returns the created package as its result.
    created_package = response_dict['result']
    pprint.pprint(created_package)

```

Appendix F - Resulting files: CSV, JSON, JSON-LD

The first few lines of the resulting CSV file 'artist.csv'.

```
artist_id,firstname,name,cityofbirth_de,cityofbirth_en,cityofdeath_de,cityofdeath_en,
  birthyear,deathyear
2,,gelatin,,,,,
3,Markus,Muntean,"Graz, Österreich","Graz, Austria",,,1962,
4,Adi,Rosenblum,"Haifa, Israel","Haifa, Israel",,,1962,
5,Vito,Acconci,"New York, NY, USA","New York, NY, USA",,,1940,
6,Max,Ackermann,"Berlin, Deutschland","Berlin, Germany","Unterlengenhardt, Deutschland","
  Unterlengenhardt, Germany",1887,1975
8,,Anouj,"Poissy, Frankreich","Poissy, France",,,1941,
9,Siegfried,Anzinger,"Weyer, Österreich","Weyer, Austria",,,1953,
10,Karel,Appel,"Amsterdam, Niederlande","Amsterdam, The Netherlands","Zürich, Schweiz","
  Zurich, Switzerland",1921,2006
11,,Arman,"Nizza, Frankreich","Nice, France","New York, NY, USA","New York, NY, USA"
  ,1928,2005
12,,Armando,"Amsterdam, Niederlande","Amsterdam, The Netherlands",,,1929,
13,Hans (Jean),Arp,"Straßburg, Frankreich (damals Deutschland)","Straßburg, Frankreich (
  damals Deutschland)","Basel, Schweiz","Basel, Switzerland",1886,1966
14,Sophie,Taeuber-Arp,"Davos, Schweiz","Davos, Switzerland","Zürich, Schweiz","Zurich,
  Switzerland",1889,1943
15,,Ay-O,"Tamatsukuri, Japan","Tamatsukuri, Japan",,,1931,
16,,Art & Language,,,,,
20,Christian Ludwig,Attersee,"Bratislava, Slowakische Republik","Bratislava, Slovak
  Republic",,,1940,
21,Joannis,Avramidis,"Batum, Georgien","Batum, Georgian Republic",,,1922,
24,Marina,Abramovic,"Belgrad, Serbien","Belgrade, Serbia",,,1946,
25,Kurt,Absolon,"Wien, Österreich","Vienna, Austria","Wulkaprodersdorf, Österreich","
  Wulkaprodersdorf, Austria",1925,1958
26,Friedrich,Achleitner,"Schalchen, Österreich","Schalchen, Austria",,,1930,
27,René,Acht,"Basel, Schweiz","Basel, Switzerland","Herbolzheim, Deutschland","Herbolzheim,
  Germany",1920,1998
28,Marc,Adrian,"Wien, Österreich","Vienna, Austria","Wien, Österreich","Vienna, Austria"
  ,1930,2008
29,Eva,Aeppli,"Zofingen, Schweiz","Zofingen, Switzerland",,,1925,
30,Billy,Al Bengston,"Dodge City, KS, USA","Dodge City, KS, USA",,,1934,
31,Josef,Albers,"Bottrop, Deutschland","Bottrop, Germany","New Haven, CT, USA","New Haven,
  CT, USA",1888,1976
32,Pierre,Alechinsky,"Brüssel, Belgien","Brussels, Belgium",,,1927,
33,Mel,Alexenberg,"New York, NY, USA","New York, NY, USA",,,1937,
35,Stephen,Althouse,"Washington D.C., USA","Washington D.C., USA",,,1948,
36,Eric,Andersen,"Kopenhagen, Dänemark","Copenhagen, Denmark",,,1942,
39,Getulio,Alviani,"Udine, Italien","Udine, Italy",,,1939,
40,John,de Andrea,"Denver, CO, USA","Denver, CO, USA",,,1941,
41,Nicola,De Maria,"Foglianise, bei Benevento, Italien","Foglianise near Benevento, Italy"
  ,,,1954,
42,Walter,De Maria,"Albany, CA, USA","Albany, CA, USA","New York, NY, USA","New York, NY,
  USA",1935,2013
43,Nikolaij,Andronow,"Moskau, Russland","Moscow, Russia",,,1929,1998
44,Eduard,Angeli,"Wien, Österreich","Vienna, Austria",,,1942,
45,Horst,Antes,"Heppenheim an der Bergstraße, Deutschland","Heppenheim an der Bergstraße,
  Germany",,,1936,
46,Siegrun,Appelt,"Bludenz, Österreich","Bludenz, Austria",,,1965,
48,Alice,Aycock,"Harrisburg, PA, USA","Harrisburg, PA, USA",,,1946,
49,Francis,Bacon,"Dublin, Großbritannien","Dublin, Great Britain","Madrid, Spanien","Madrid,
  Spain",1909,1992
50,Txomin,Badiola,"Bilbao, Spanien","Bilbao, Spain",,,1957,
51,Jo,Baer,"Seattle, WA, USA","Seattle, WA, USA",,,1929,
52,Olle,Baertling,"Halmstad, Schweden","Halmstad, Sweden","Stockholm, Schweden","Stockholm,
  Sweden",1911,1981
53,Jean,Baier,"Genf, Schweiz","Geneva, Switzerland",,,1932,1999
54,Enrico,Baj,"Mailand, Italien","Milan, Italy","Vergiate, Italien","Vergiate, Italy"
  ,1924,2003
55,Mirosław,Balka,"Warschau, Polen","Warsaw, Poland",,,1958,
56,Giacomo,Balla,"Turin, Italien","Torino, Italy","Rom, Italien","Rome, Italy",1871,1958
57,Alain,Balzac,"Agen, Haute-Garonne, Frankreich","Agen, Haute-Garonne, France",,,1958,
58,Ernst,Barlach,"Wedel, Deutschland","Wedel, Germany","Rostock, Deutschland","Rostock,
  Germany",1870,1938
```

The first object of the resulting JSON file 'data.json'.

```
{
  "@data": {
    "1": {
      "acquiry_date": "1962",
      "artists": {
        "892": {
          "birthyear": 1886,
          "cityofbirth_de": "Orel, Russland",
          "cityofbirth_en": "Orel, Russia",
          "cityofdeath_de": "Paris, Frankreich",
          "cityofdeath_en": "Paris, France",
          "deathyear": 1962,
          "firstname": "Antoine",
          "name": "Pevsner"
        }
      }
    },
    "creditline": "erworben/acquired in",
    "default_image": "https://www.mumok.at/imageobject.php?objid=1",
    "description_de": null,
    "description_en": null,
    "exhibitions": {
      "107": {
        "city": null,
        "datefrom": null,
        "dateuntil": null,
        "location": "U3 B Kubismus und Konstruktivismus",
        "title": "U3.0002"
      },
      "395": {
        "city": "Wien, Österreich",
        "datefrom": "2007-03-23",
        "dateuntil": "2007-10-07",
        "location": "Museum moderner Kunst Stiftung Ludwig Wien\r\nAusstellung in Ebene 8",
        "title": "Laboratorium Moderne.Bildende Kunst, Fotografie und Film im Aufbruch"
      },
      "434": {
        "city": "Wien, Österreich",
        "datefrom": "2008-02-29",
        "dateuntil": "2008-05-18",
        "location": "Mumok, Wien",
        "title": "Genau und anders/Mathematik in der Kunst von Dürer bis Sol LeWitt"
      },
      "451": {
        "city": "Wien, Österreich",
        "datefrom": "2008-06-06",
        "dateuntil": "2008-11-02",
        "location": "MUMOK Wien\r\nin den Ausstellungsebenen 6,7 und 8",
        "title": "Mehransichtigkeiten-Derain,Kupka,Magritte,Klee...../Werner Hofmann zu Ehren"
      },
      "546": {
        "city": "Wien, Österreich",
        "datefrom": "2010-03-26",
        "dateuntil": "2010-06-27",
        "location": "Ausstellung Ebene 3 (Manzoni)\r\nAusstellung Ebene 1 ( Pevsner, Mondrian, Moholy-Nagy)",
        "title": "Bilder über Bilder.Diskursive Malerei von Albers bis Zobernig. Aus der Daimler Kunst Sammlung"
      },
      "557": {
        "city": "Wien, Österreich",
        "datefrom": "2010-06-25",
        "dateuntil": "2011-01-03",
        "location": "Ausstellung MUMOK\r\nEbene 7 bzw. 8",
        "title": "The Moderns.Revolution in Art and Science 1890-1935"
      },
      "979": {
        "city": null,
        "datefrom": "2014-03-14",

```

```

        "dateuntil": "2015-02-08",
        "location": "mumok wien, Ebene -3",
        "title": "Die Gegenwart der Moderne"
    }
},
"exploitation_rights": "VBK Wien",
"in_exhibition": "no",
"inventorynr": "P 1/0",
"inventorynr_sorted": "P 00001/000",
"material_de": "Bronze, Glas",
"material_en": "Bronze, glass",
"measurement_depth": 85.0,
"measurement_height": 80.5,
"measurement_unit": "cm",
"measurement_width": 81.0,
"originaltitle": "Konstruktion für einen Flughafen",
"production_date": "1934 - 1935",
"tags": "Russland, Frankreich, Konstruktivismus, Kinetismus, Abstraction-Création
, Abstraktion, Modell, Klassische Moderne,",
"title_de": "Konstruktion für einen Flughafen",
"title_en": "Construction Design for an Airport"
}
},
"@metadata": {
  "@context": [
    "http://www.w3.org/ns/csvw",
    {
      "@language": "en"
    }
  ],
  "dc:license": {
    "@id": "https://creativecommons.org/licenses/by/3.0/at/legalcode"
  },
  "dc:modified": {
    "@type": "xsd:date",
    "@value": "2015-07-04"
  },
  "dc:publisher": {
    "schema:name": "mumok - museum moderner kunst stiftung ludwig wien",
    "schema:url": {
      "@id": "http://www.mumok.at"
    }
  },
  "dc:title": "Objects from the mumok collection together with their artists and
  exhibitions",
  "dcat:keyword": [
    "art",
    "museum"
  ],
  "tableSchema": {
    "columns": [
      {
        "datatype": "integer",
        "dc:description": "An identifier for the object.",
        "name": "objid",
        "required": "true",
        "titles": [
          "OBJID",
          "Object Identifier"
        ]
      },
      {
        "datatype": "string",
        "dc:description": "The inventory number of the object.",
        "name": "inventorynr",
        "titles": "Inventory number"
      },
      {
        "datatype": "string",
        "dc:description": "The inventory number of the object, sortable by
        prefixed zeroes to get a fixed length.",
        "name": "inventorynr_sorted",
        "titles": "sortable inventory number"
      }
    ]
  }
}

```



```

    "datatype": "string",
    "dc:description": "The original title of the object as given by the
        creator.",
    "name": "originaltitle",
    "titles": "Originaltitle"
},
{
    "datatype": "string",
    "dc:description": "The title of the object translated to German.",
    "lang": "de",
    "name": "title_de",
    "titles": "Title in German"
},
{
    "datatype": "string",
    "dc:description": "The title of the object translated to English.",
    "lang": "en",
    "name": "title_en",
    "titles": "Title in English"
},
{
    "datatype": "string",
    "dc:description": "The date on which the object has been produced (
        informal value: year or range of years)",
    "name": "production_date",
    "titles": "Date of production"
},
{
    "datatype": "boolean",
    "dc:description": "States if the object is currently on display in an
        exhibition (possible values: 'yes'/'no').",
    "name": "in_exhibition",
    "titles": "Object in exhibition"
},
{
    "datatype": "string",
    "dc:description": "The materials the object is made of in German.",
    "lang": "de",
    "name": "material_de",
    "titles": "Used materials in German"
},
{
    "datatype": "string",
    "dc:description": "The materials the object is made of in English.",
    "lang": "en",
    "name": "material_en",
    "titles": "Used materials in English"
},
{
    "datatype": "double",
    "dc:description": "The height measurement of the object (unit can be
        deducted from column 'measurement_unit').",
    "name": "measurement_height",
    "titles": "Measurement of height"
},
{
    "datatype": "double",
    "dc:description": "The width measurement of the object (unit can be
        deducted from column 'measurement_unit').",
    "name": "measurement_width",
    "titles": "Measurement of width"
},
{
    "datatype": "double",
    "dc:description": "The depth measurement of the object (unit can be
        deducted from column 'measurement_unit').",
    "name": "measurement_depth",
    "titles": "Measurement of depth"
},
{
    "datatype": "string",
    "dc:description": "The unit of measurement for the given measurements ('
        measurement_height', 'measurement_width', 'measurement_depth').",
    "name": "measurement_unit",
    "titles": "Measurement unit"
}

```

```

},
{
  "datatype": "string",
  "dc:description": "The description of the object in German.",
  "lang": "de",
  "name": "description_de",
  "titles": "Description in German"
},
{
  "datatype": "string",
  "dc:description": "The description of the object in English.",
  "lang": "en",
  "name": "description_en",
  "titles": "Description in English"
},
{
  "datatype": "string",
  "dc:description": "The creditline states the owner or loaning party of
    the object (usually in format: 'German creditline'/'English
    creditline').",
  "name": "creditline",
  "titles": "Creditline"
},
{
  "datatype": {
    "base": "date",
    "format": "yyyy"
  },
  "dc:description": "The year in which the object has been acquired by the
    museum.",
  "name": "acquiry_date",
  "titles": "Date of acquiry"
},
{
  "datatype": "string",
  "dc:description": "The owner of the object's exploitation rights.",
  "name": "exploitation_rights",
  "titles": "Exploitation rights"
},
{
  "datatype": "string",
  "dc:description": "The URL for the default image being set by the data
    operator.",
  "name": "default_image",
  "titles": "Default Image"
},
{
  "datatype": "string",
  "dc:description": "The tags that were used to classify the object.",
  "name": "tags",
  "titles": "Tags"
},
{
  "columns": [
    {
      "datatype": "integer",
      "dc:description": "A unique identifier for the exhibition.",
      "name": "exhibition_id",
      "required": "true",
      "titles": "Exhibition Identifier"
    },
    {
      "datatype": "string",
      "dc:description": "The city in which the exhibition took place (no
        language defined, mostly in German).",
      "name": "city",
      "titles": "City"
    },
    {
      "datatype": "string",
      "dc:description": "The title of the exhibition.",
      "name": "title",
      "titles": "Exhibition Title"
    }
  ]
}

```

```

        "datatype": "string",
        "dc:description": "The location at which the exhibition took place
        (no language defined, mostly in German).",
        "name": "location",
        "titles": "Location of exhibition"
    },
    {
        "datatype": "string",
        "dc:description": "The date on which the exhibition has started.",
        "name": "datefrom",
        "titles": "Exhibition starting date"
    },
    {
        "datatype": "string",
        "dc:description": "The date on which the exhibition has ended.",
        "name": "dateuntil",
        "titles": "Exhibition ending date"
    }
],
"datatype": "object",
"dc:description": "Exhibitions in which the object has been presented.",
"name": "exhibitions",
"primary_key": "exhibition_id",
"titles": "Exhibitions"
},
{
    "columns": [
        {
            "datatype": "integer",
            "dc:description": "A unique identifier for the artist.",
            "name": "artist_id",
            "required": "true",
            "titles": "Artist Identifier"
        },
        {
            "datatype": "string",
            "dc:description": "The firstname of the artist.",
            "name": "firstname",
            "titles": "Firstname"
        },
        {
            "datatype": "string",
            "dc:description": "The lastname of the artist.",
            "name": "name",
            "titles": "Name (Lastname)"
        },
        {
            "datatype": "string",
            "dc:description": "The city in which the artist has been born, in
            German.",
            "lang": "de",
            "name": "cityofbirth_de",
            "titles": "City of birth in German"
        },
        {
            "datatype": "string",
            "dc:description": "The city in which the artist has been born, in
            English.",
            "lang": "en",
            "name": "cityofbirth_en",
            "titles": "City of birth in English"
        },
        {
            "datatype": "string",
            "dc:description": "The city in which the artist has died, in
            German.",
            "lang": "de",
            "name": "cityofdeath_de",
            "titles": "City of death in German"
        },
        {
            "datatype": "string",
            "dc:description": "The city in which the artist has died, in
            English.",
            "lang": "en",
            "name": "cityofbirth_en",

```

```

        "titles": "City of death in English"
    },
    {
        "datatype": {
            "base": "date",
            "format": "yyyy"
        },
        "dc:description": "The year in which the artist has been born.",
        "name": "birthyear",
        "titles": "Artist's birthyear"
    },
    {
        "datatype": {
            "base": "date",
            "format": "yyyy"
        },
        "dc:description": "The year in which the artist has died.",
        "name": "deathyear",
        "titles": "Artist's deathyear"
    }
],
"datatype": "object",
"dc:description": "Artists that have been involved with the creation of
the object.",
"name": "artist",
"primaryKey": "artist_id",
"titles": "Artists"
}
],
"primaryKey": "objid"
},
"url": "http://data.mumok.at/obj.json"
}
}

```

The metadata context and the first object of the resulting JSON-LD file
'data.jsonld'.

```
{
  "@context": {
    "title_de": {
      "@id": "alternateName",
      "@language": "de"
    },
    "acquiry_date": "https://schema.org/copyrightYear",
    "material_en": {
      "@id": "artMedium",
      "@language": "en"
    },
    "birthyear": "birthDate",
    "cityofbirth_de": {
      "@id": "birthPlace",
      "@language": "de"
    },
    "description_de": {
      "@id": "description",
      "@language": "de"
    },
    "cityofdeath_de": {
      "@id": "deathPlace",
      "@language": "de"
    },
    "@vocab": "http://schema.org/",
    "production_date": "dateCreated",
    "measurement_width": {
      "@id": "width",
      "@type": "QuantitativeValue"
    },
    "tags": "keywords",
    "measurement_height": {
      "@id": "height",
      "@type": "QuantitativeValue"
    },
    "city": {
      "@type": "Place"
    },
    "inventorynr_sorted": "dc:identifier",
    "firstname": "givenName",
    "creditline": "dc:Provenance",
    "title_en": {
      "@id": "alternateName",
      "@language": "en"
    },
    "dc": "http://purl.org/dc/elements/1.1/",
    "description_en": {
      "@id": "description",
      "@language": "en"
    },
    "cityofbirth_en": {
      "@id": "birthPlace",
      "@language": "en"
    },
    "dc:license": {
      "@id": "https://creativecommons.org/licenses/by/3.0/at/legalcode"
    },
    "exploitation_rights": "dc:RightsHolder",
    "material_de": {
      "@id": "artMedium",
      "@language": "de"
    },
    "measurement_depth": {
      "@id": "depth",
      "@type": "QuantitativeValue"
    },
    "name": "familyName",
    "url": "@id",
    "originaltitle": "name",
    "default_image": "thumbnailUrl",
    "inventorynr": "dc:identifier",
  }
}
```

```

"cityofdeath_en": {
  "@id": "deathPlace",
  "@language": "en"
},
"deathyear": "deathDate"
},
"@set": [
  {
    "title_de": "Konstruktion für einen Flughafen",
    "exhibitions": [
      {
        "city": null,
        "title": "U3.0002",
        "datefrom": null,
        "location": "U3 B Kubismus und Konstruktivismus",
        "@id": "_e:107",
        "@type": "dc:Event",
        "dateuntil": null
      },
      {
        "city": "Wien, Österreich",
        "title": "Laboratorium Moderne.Bildende Kunst, Fotografie und Film im
        Aufbruch",
        "datefrom": "2007-03-23",
        "location": "Museum moderner Kunst Stiftung Ludwig Wien\r\nAusstellung in
        Ebene 8",
        "@id": "_e:395",
        "@type": "dc:Event",
        "dateuntil": "2007-10-07"
      },
      {
        "city": "Wien, Österreich",
        "title": "Genau und anders/Mathematik in der Kunst von Dürer bis Sol
        LeWitt",
        "datefrom": "2008-02-29",
        "location": "Mumok, Wien",
        "@id": "_e:434",
        "@type": "dc:Event",
        "dateuntil": "2008-05-18"
      },
      {
        "city": "Wien, Österreich",
        "title": "Mehransichtigkeiten-Derain,Kupka,Magritte,Klee...../Werner
        Hofmann zu Ehren",
        "datefrom": "2008-06-06",
        "location": "MUMOK Wien\r\nin den Ausstellungsebenen 6,7 und 8",
        "@id": "_e:451",
        "@type": "dc:Event",
        "dateuntil": "2008-11-02"
      },
      {
        "city": "Wien, Österreich",
        "title": "Bilder über Bilder.Diskursive Malerei von Albers bis Zobernig.
        Aus der Daimler Kunst Sammlung",
        "datefrom": "2010-03-26",
        "location": "Ausstellung Ebene 3 (Manzoni)\r\nAusstellung Ebene 1 (
        Pevsner, Mondrian, Moholy-Nagy)",
        "@id": "_e:546",
        "@type": "dc:Event",
        "dateuntil": "2010-06-27"
      },
      {
        "city": "Wien, Österreich",
        "title": "The Moderns.Revolution in Art and Science 1890-1935",
        "datefrom": "2010-06-25",
        "location": "Ausstellung MUMOK\r\nEbene 7 bzw. 8",
        "@id": "_e:557",
        "@type": "dc:Event",
        "dateuntil": "2011-01-03"
      },
      {
        "city": null,
        "title": "Die Gegenwart der Moderne",
        "datefrom": "2014-03-14",
        "location": "mumok wien, Ebene -3",

```

```

        "@id": "_e:979",
        "@type": "dc:Event",
        "dateuntil": "2015-02-08"
    }
],
"acquiry_date": "1962",
"material_en": "Bronze, glass",
"artists": [
    {
        "cityofdeath_de": "Paris, Frankreich",
        "birthyear": 1886,
        "firstname": "Antoine",
        "cityofbirth_de": "Orel, Russland",
        "cityofdeath_en": "Paris, France",
        "deathyear": 1962,
        "familyname": "Pevsner",
        "@id": "http://www.mumok.at/node/13519",
        "@type": "Person",
        "cityofbirth_en": "Orel, Russia"
    }
],
"description_de": null,
"production_date": "1934 - 1935",
"measurement_width": 81.0,
"@id": "http://www.mumok.at/node/15011",
"tags": "Russland, Frankreich, Konstruktivismus, Kinetismus, Abstraction-Création
, Abstraktion, Modell, Klassische Moderne,",
"measurement_height": 80.5,
"inventorynr_sorted": "P 00001/000",
"creditline": "erworben/acquired in",
"title_en": "Construction Design for an Airport",
"description_en": null,
"in_exhibition": "no",
"exploitation_rights": "VBK Wien",
"material_de": "Bronze, Glas",
"measurement_depth": 85.0,
"originaltitle": "Konstruktion für einen Flughafen",
"default_image": "https://www.mumok.at/imageobject.php?objid=1",
"inventorynr": "P 1/0",
"measurement_unit": "cm"
}
]
}

```