CHALMERS

IMPRESS integrated data format

Deliverable from the IMPRESS project

Johan Tivander

Maria Erixon

Raul Carlson

Ann-Christin Pålsson

IMI - Industrial Environmental Informatics

for

CPM - Centre for Environmental Assessment of Product and Material Systems

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2006

CPM Report 2006:15

1 Summary

This report contains a description and definition of an integrated data format for environmental information named the *IMPRESS data format*. This is a result from sub-project 3 – Maintenance of integration - in the CPM project IMPRESS.

The IMPRESS data format is a combination of several formats developed for specific information domains in previous IMI and CPM projects including the SPINE data format, the RAVEL data format, the IA 98, OMNIITOX data format, etc. The original concept models and data structures of these formats are kept intact for the information domain they are intended. The integration has focused on interfaces between domains and on harmonization of data structures for concepts that are shared between the domains. For example the structures for storing numerical quantitative values, document references, and meta data differed slightly between the individual formats but for the IMPRESS data format a common structure is used.

The IMPRESS data format can be directly used for creating databases for environmental information. However due to the extensive information scope it may be practical only to implement a sub-set of tables may be used to match the scope of a specific application, e.g. for a DfE application it may not be necessary to implement tables for detailed environmental impact modelling.

The resulting IMPRESS data format can be regarded as a practical example that shows how primary data can be shared among applications. It can be used as an integration reference format when designing or redesigning existing environmental information system.

Table of Contents

2 Introduction

2.1 Background

During the past work within CPM and IMI many applications for industrial environmental information management have been developed. Each application is based on a concept model and a data format for the information relevant for the application. In 1995 the SPINE data format[1] was developed in CPM for managing Life Cycle Assessment (LCA) data. It was successfully implemented for the Swedish national Life Cycle Inventory (LCI) database named SPINE@CPM. In the CPM project Integrated Environmental Information Systems in 1998 the impact assessment part of the SPINE format was refined to match the concepts defined in the new ISO 14042 standard on Life Cycle Impact Assessment (LCIA). This specific format addition was named IA98[2] and was implemented and used for building up a transparent and generic impact assessment model database. A data format for Design for Environment (DfE) was developed in the RAVEL project in 2000[3]. The structures for process and impact assessment data was taken directly from SPINE and IA98 format which means the RAVEL

[1] Carlson R, Löfgren G, Steen B, "SPINE, A Relation Database Structure for Life Cycle Assessment"; Swedish Environmental Research Institute, IVL-Report B 1227, Göteborg, Sweden, 1995

[2] Carlson R, Steen B, "A Data Model for LCA Impact Assessment", Presented at 8th Annual Meeting of SETAC-Europe 1998 14-18 April, Bordeaux

[3] Carlson R, Forsberg P, "The RAVEL Information Platform Data Model" 2000 RAVEL project doc nr CPM-000919

format is an integrated data format for LCA and DfE data. The RAVEL format has been used with minor modifications in implementations of applications for DfE within, e.g. in IMPRESS sub-project 7[4]. CPM initiated the work on standardisation of LCI data which resulted in the ISO/TS 14048 LCA data documentation format[5]. As the overall conceptual structure of the 14048 format was built on the SPINE format the two formats are fairly similar and a strategy was drawn up for adapting the CPM databases based on SPINE at that time towards the world-wide consensus based format of ISO/TS 14048. A first implementation of a database format following the ISO/TS 14048 requirements was developed in 2001[6] and a mapping between SPINE and ISO/TS 14048 was finalised in 2003[7]. In the CPM project Policy controlled environmental management a high resolution process data format with the working name 2IMI 2003 was developed based on the ISO/TS 14048 concepts, specifically designed for EMS Enviornmental Management Systems. It has been implemented in the integrated application named the IMI portal[8]. Practical solutions for data exchange between formats are included in the IMI Portal including conversions between SPINE and ISO/TS 14048. In the EU funded project

---

[4] Erlandsson M, Flemström K, 2006, "Measurement and communication of environmental performance of products", Deliverable from IMPRESS sub project 7, CPM report 2006:2

[5] ISO, 2002, "ISO/TS 14048 Environmental management — Life cycle assessment — Data documentation format", ISO Technical specification

[6] Carlson R, Tivander J, "Data definition and file syntax for ISO/TS 14048 data exchange - including an implementation of a data storage format based on ISO/TS 14048", CPM Report 2001:9

[7] Carlson R, Erlandsson M, Flemström K, Pålsson A-C, Tidstrand U, Tivander J, "Data format mapping between SPINE and ISO/TS 14048", CPM report 2003:8

[8] IMI, 2006, "The IMI Portal", http://databases.imi.chalmers.se/nukes/index.html, 2006

OMNIITOX[9] the OMNIITOX data format[10] was developed to facilitate information management of detailed environmental impact models.

The intention with each format is primarily to support the specific application system for which they are developed. Each format has also been developed based on common principles and on previous concepts and formats which enables integration of the formats. Common concepts used between applications have been continuously identified as more experience has been gained by the practical work in projects. However, many differences occur between the individual formats, often on a detailed technical level such as different definitions between formats for data types of fields, but also by heterogeneous concept model including different names of the same concepts (synonyms) and the same name for different concepts (homonyms). In the IMPRESS project specific efforts have been put to fully harmonize the formats into one integrated format. The result is named the IMPRESS format and is presented in this report.

The IMPRESS format is intended to be a support when integrating information systems. It is a description of how to structure environmental information in a meaningful way so that it can be used in many user applications. In this way the IMPRESS format as a reference format that

[9] "Operational Models aNd Information tools for Industrial applications of eco/TOXicological impact assessments - OMNIITOX", http://omniitox.imi.chalmers.se/ 2006, EC contract G1RD-CT–2001–00501

[10] Tivander J, Carlson R, Erlandsson M, Erixon M, Geiron K, "Concept Model for the OMNIITOX Information System Including OMNIITOX Data Format Definition" OMNIITOX project report D20 and D26, 2004, EC contract G1RD-CT–2001–00501

serves as an example of how many applications can share a common data structure. It can also be used directly as a template to create databases.

2.2 The IMPRESS project

The IMPRESS data format has been developed within the IMPRESS project, in sub-project 3.

2.2.1 Goals and objectives

The IMPRESS project (acronym for IMPlementation of integRated Environmental information SystemS) 2004 – 2006 aimed at showing how information, methods and tools that supports environmentally related decisions within the industry, can be integrated with each other and with the corporate business processes and also implemented into the organizations.

The companies that participated in the project were Akzo Nobel, Bombardier Transportation, Duni, IKEA, ITT Flygt, SCA and Stora Enso. Research and development work was performed together with the research group Industrial Environmental Informatics (IMI) at Chalmers University of Technology. The project was funded by the Swedish competence Center for environmental assessment of Product and Material systems (CPM).

The overall task of IMPRESS was to implement method and tool integration with business processes in a number of industrial companies. The objectives were:

- Decrease the cost for industrial environmental management.
- Decrease the cost for developing, using and maintaining data, tools and methods for industrial environmental management.
- Facilitate acquisition of environmental information.
- Provide educational tools for industrial environmental management.

The project also aimed at investigating possibilities for exploitation and dissemination of previous and new CPM results to enhance the value and increase the usability of the results.

The specific methods and tools studied in this project are DfE, ERA, and LCA from a product perspective, EMS and LCA from a process perspective, and CO2-emission trading from a societal perspective. Six industrial application and implementation cases were included in the project:

- Emission trading
- Measurement and communication of environmental performance of products
- Environmental management at site and group level
- Risk management adapted to REACH
- Three tools for IPP
- Integration of experiences and new information

These six cases were studied in detail in close cooperation between IMI and the companies in different sub projects, including e.g. market analyses, specific method development, implementation etc. A general integration methodology was regarded in a separate sub-project. Similarly, technical maintenance for integration, commercialization work, and knowledge exchange were performed in three different sub-projects.

2.3 Sub project 3 Maintenance of integration

The goal of sub-project 3 was to maintain the integrating quality of the prototype tools, in terms of software and manuals, developed in the industrial application sub-projects.
This is to take responsibility for a CPM base-line model that is strategic, applicable in a long term perspective and aimed at sustainable development as it is perceived today.

State of the art of the methods and already existing tools would be demonstrated to raise the level of understanding within the companies to manage environmental information in a transparent, quality assured, and integrated way that decreases costs and increases the benefits. This is a prerequisite for an effective collaboration in the industrial application sub-projects.

The data administration and technical administration of on-line CPM data services (LCI@CPM) were carried out in this sub-project as well.

3 Work procedure

The development of the IMPRESS data format took a starting point in the existing data formats described in chapter *2.1 Introduction – Background*. Documentation of the formats and implemented databases based on the formats was assembled. The concept models and tables and fields for all formats were nearly fully covered in available reports from IMI and CPM. The existing databases provided examples on actual data in the fields for further understanding of the concepts.

The overall high level concept of SPINE was used as a reference model that sets the scope of what is included in domain of environmental information and also the main internal structure of environmental information, see chapter *4.1 Information scope – Overview*. The SPINE concept model has also been used as a main reference model in the General method for integration[11] developed in IMPRESS sub-project 2 in parallel to the development of the IMPRESS data format.

The formats was first analysed in terms sorting out common concepts from the concepts exclusive for one format. Common concepts that were identically defined in the formats could be directly integrated. However different definitions in terms of concept name, data type, or relations to other concepts existed. These issues was assembled into “issue areas”, e.g. issues regarding meta data structure, issues regarding resolution of environmental impact model data

[11] Erixon M, Tivander J, Pålsson A-C, Carlson R, 2006, “General method for integration of industrial environmental information systems” IMPRESS project report, Industrial Environmental Informatics, CPM Report 2006:14

and environmental impact assessment data, or issues on consistent name spaces and nomenclatures.

A series of discussion seminars was held by within research group IMI on one issue at a time. The discussions started by reaching a common conceptual description of the issue area based on experiences on methods and applications using the information and followed by decisions on the final naming and relations of concepts and also on data format definition for the concept. All decisions were documented in minutes and the resulting format was specified as presented in chapter *5. The data format* in this report. A database was also built up as a working tool for direct implementation of the decided format.

4 Information scope

4.1 Overview

The high level concept model of SPINE is a model that describes the scope of environmental information and the main internal structure of environmental information, see figure 1. For all environmental decision support for industrial applications information is needed on the controlled technical system, on the physical impact on the environmental system, and on the importance of the environmental impact in relation to the useful goods generated by the controlled technical system. All existing data formats developed within CPM and IMI can be related to the SPINE concept model. For example the SPINE data format used for documenting LCI data is data about technical systems and the OMNIITOX data format is designed to store data for environmental impact models with interfaces in terms of aspects and indicators to the technical and social system. Figure 2 shows an overview of the main concepts of the IMPRESS data format. It can be regarded as a more highly resolved representation of the SPINE model. Blue colours represent technical system concepts, orange represents social system concepts, green represents nature system concepts. The grey coloured substance concept is common to both technical and nature system.

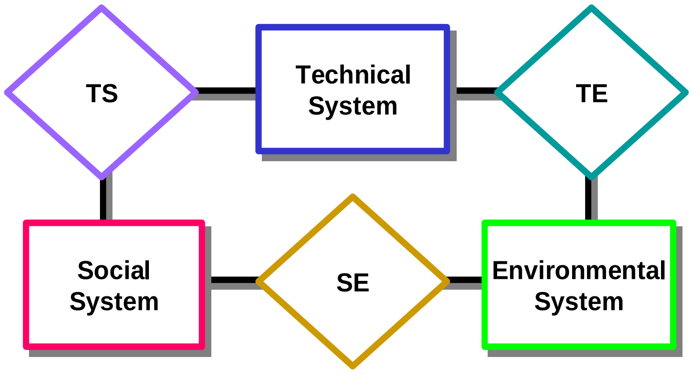

Figure 1 The high level concept model of SPINE defining the scope of environmental information for industrial applications

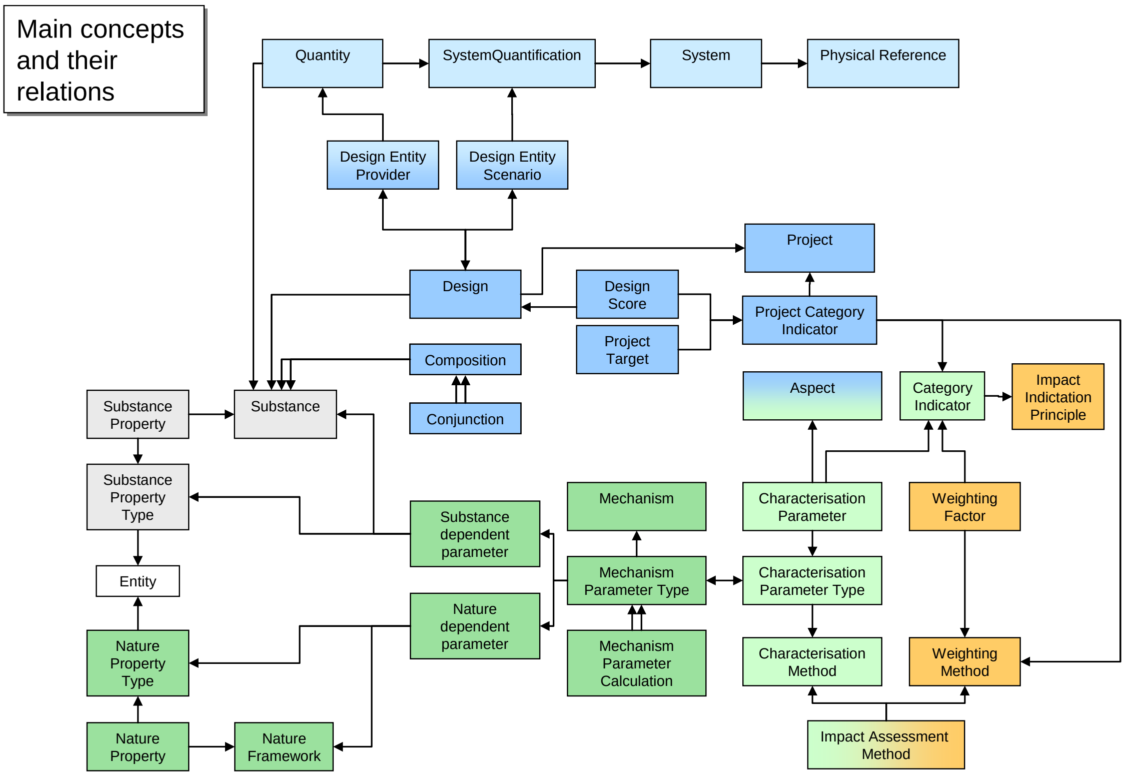

Figure 2 Overview of the integrated IMPRESS data format. Blue colours represent technical system concepts, orange represents social system concepts, green represents nature system concepts. The substance concept is used in all systems.

4.2 User applications

The data format supports information management for a wide range of environmental applications:

- Selection and definition of indicators
- Setting targets and environmental performance follow-up
- Documentation of CO2 emissions from a site in according to an emission trading scheme
- Environmental Management System
- Design for Environment
- Green procurement/purchasing
- Communication of environmental requirements to suppliers
- Communication of environmental performance to e.g. customers in terms of environmental product declaration
- Environmental Risk Assessment
- Generation of Safety Data Sheets
- LCA
- Material and substance property management

Different application use different parts of the format but many parts are shared. For example, the system concept is used for LCI process model data as well as EMS site model data, in DfE the tables related to substance are used for information about materials and products, and in Environmental impact modeling the substance tables are used to store property data values required to quantify the environmental impact.

5 The IMPRESS data format

The IMPRESS data format is presented in sections based on main concepts. In each section the tables related to the concepts are described according to the table below.

Fields:	**[Fieldname] (primary key in bold)**	[Field content description]
	…	…
Unique constraint:	[Fieldname(s) with unique constraint]	[Reason for the constraint]
Foreign keys:	[Fieldname(s) referencing a foreign key]	[Name of table with referenced key]
	…	…

Since unique constraints are only scarcely occurring this row is omitted if no constraint is added on the fields.

5.1 Project

A project represents an organizational task. Information about a project concerns connecting users, applications, and information. Project members with various roles in the project can be specified and additional users can be invited to participate. Data that is treated within the project is specified through the project context, see also chapter 5.13.5 Context. For example, within an LCA study project, the inventory data of physical references and systems are pointed out. The specific software or other method application used is pointed out as a project property. Data access and manipulation restrictions depending on user role can be handled through application specific solutions by using additional project properties.

Environmental impact indicators and weighting method can be associated with the project and target values for the environmental performance can be set for the system being studied.

Specific nomenclatures may be chosen for the project work. Together these nomenclatures constitute a project vocabulary. Figure 3 shows the tables of the IMPRESS data format used for handling information from a project perspective.

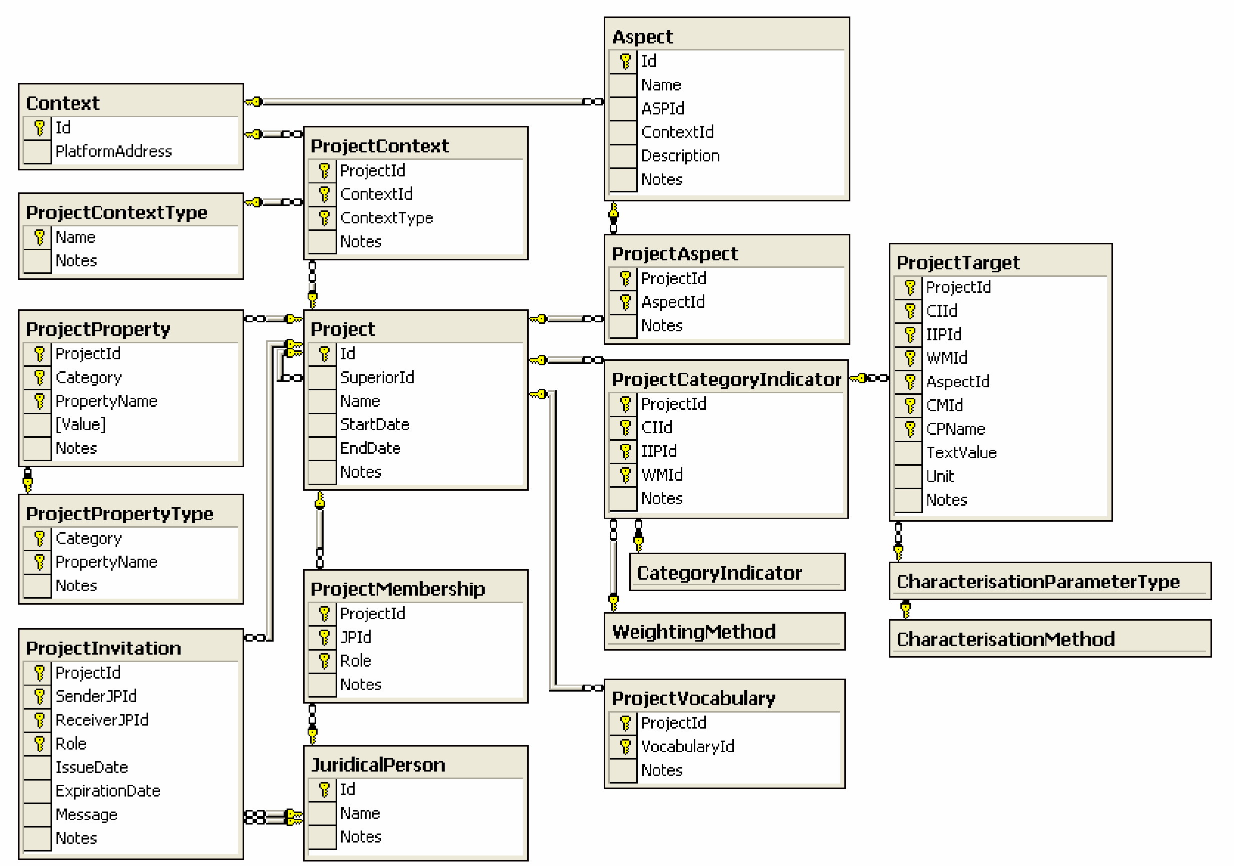

Figure 3 The tables in the IMPRESS data format related to the Project concept

5.1.1 Project

A project connects users with an application and with information. By relating a project to a superior project, project structures are created. The project structure handles the actual building process of a product. It describes how the task of building a product is organizationally divided into pieces. The top node project has the task of building the product. From this project several subprojects are started, each with a more specified and detailed description. This division into subprojects continues until easy comprehensible projects have been reached.

Fields:	**Id**	Artificial identification number to uniquely identify the project.
	SuperiorId	Projects can be subprojects of other projects, meaning that references can be made to the project which a specific project is part of.
	Name	A project shall be given a name.
	StartDate	The start date of a project can be described.
	EndDate	The end date of a project can be described.
	Notes	Additional relevant information about the project.
Foreign keys:	SuperiorId	Referencing the table Project.

5.1.2 ProjectTarget

The selectable category indicators represent quantitative elements of concern in the environment. The table ProjectTarget keeps a record of the set of indicators that are evaluated in the project and also the weight of the indicator. Finally this table enables to express quantitative and clear targets that can be compared with the assessed environmental impact of the system being studied.

Fields:	**ProjectId**	Reference to the project to which the indicator is to be related.
	CIid	Reference to the category indicator.
	IIPId	Reference to the impact indication principle used when defining the indicator.
	WMId	Reference to the weighting method that makes the indicators used in the project comparable.
	TextValue	Holds the data for the specific parameter if the amount structure is not used for this parameter, (see description above). Examples: ”Blue", "Yes", "Allowed”
	Notes	Additional relevant information.
Foreign	ProjectId	Referencing the table Project.

keys:		
	CIId, IIPId	Referencing the table CategoryIndicator.
	WMId	Referencing the table WeightingMethod.

5.1.3 ProjectCharacterisation

The table also set the characterisation parameters that shall be used to calculate the environmental impacts of an aspect.

Fields:	**ProjectId**	Reference to the project to which the indicator is to be related.
	CIid	Reference to the category indicator.
	IIPId	Reference to the impact indication principle used when defining the indicator.
	WMId	Reference to the weighting method that makes the indicators used in the project comparable.
	AspectId	Reference to the aspect, i.e. the element in the technical system that is causing the impact on the category indicator.
	CMId	Reference to the characterisation method, i.e. environmental impact assessment method.
	CPName	Reference to the characterisation parameter, i.e. ther specific quantitative expression of the environmental impact assessment method.
	Unit	The unit of the project target.
	Notes	Any additional relevant information.
Foreign keys:	ProjectId, SubstanceId	Referencing the table Design.
	ProjectId, CIId, IIPId, WMId	Referencing the table ProjectCategoryIndicator.
	CMId, CPName	Referencing the table CharacterisationParameterType.
	ValueType	Referencing the table ValueType.

	Unit	Referencing the table Unit.

5.1.4 ProjectAspect

A project can have a number of important aspects, e.g. data categories in an LCA study. This table enables to connect aspects to a project without specifying which category indicators the aspects have an impact on. Compare with the table ProjectTarget.

Fields:	**ProjectId**	Identifier of the project.
	AspectId	Reference to the aspect, i.e. the element in the technical system that is monitored.
	Notes	
Foreign keys:	ProjectId	Referencing the table Project.
	AspectId	Referencing the table Aspect.

5.1.5 ProjectProperty

A project can have a number of project specific properties, such as a target weight of the designed product etc. These properties lock the descriptions of the necessarily qualities of the project, and they can also be used for storage of knowledge.

Fields:	**ProjectId**	Reference to the project to which the property applies.
	Category	The category of the property.
	PropertyName	The name for the project property.
	Value	A value for the project property. The amount-structure is not used since no results from measurements are supposed to be stored here, only numerical scalars or strings.
	Notes	Additional relevant information about the project property.
Foreign keys:	ProjectId	Referencing the table Project.

	Category, PropertyName	Referencing the table ProjectPropertyType.

5.1.6 ProjectPropertyType

Project properties shall be selected within a predefined range of types. This table specifies the namespace of valid properties.

Fields:	**Category**	The category of the property. Examples: “Basic”, “Name”
	PropertyName	The name for the project property. Examples: “System mass”, “Number of seats”, “Long project name”
	Notes	Additional relevant information about the type of the project property type.
Foreign keys:	N/A	

5.1.7 ProjectContext

This table connects the project to the datasets managed within the project through the use of the context structure, see Context chapter 5.13.5.

Fields:	**ProjectId**	Identifier of the project.
	ContextId	Identifier of the context.
	ContextType	Reference to the type of context, e.g. “SystemPointer”, “IIPPointer”, “CMPointer”
	Notes	Additional relevant information.
Foreign keys:	ProjectId	Referencing the table Project.
	ContextId	Referencing the table Context.
	ContextType	Referencing the table ContextType.

5.1.8 ProjectContextType

Additional specification to facilitate correct interpretation of the data pointed out by the context.

Fields:	Name	Type of context. Examples: “SystemPointer”, “IIPPointer”, “CMPointer”
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.1.9 ProjectMembership

To enable e.g. customised user settings, access restrictions, design approval etc. within the information system, a project may be associated with a number of users with different roles within the project.

Fields:	**ProjectId**	Identifier of the project.
	JPId	Identifier of the juridical person.
	Role	The role of the person within the project. Any person can have different roles within the same project.
	Notes	Additional relevant information.
Foreign keys:	ProjectId	Referencing the table Project.
	JPId	Referencing the table JuridicalPerson.

5.1.10 ProjectInvitation

This table stores data on invitations of new members to a project.

Fields:	**ProjectId**	Identifier of the project.
	SenderJPId	Identifier of the juridical person sending the invitation.
	ReceiverJPId	Identifier of the juridical person receiving the invitation.
	ReceiverRole	The designated role of the receiver.
	Notes	Additional relevant information.
Foreign keys:	ProjectId	Referencing the table Project.
	SenderJPId	Referencing the table JuridicalPerson.

	ReceiverJPId	Referencing the table JuridicalPerson.

5.2 Substance and substance property

The substance concept represents physical matter. This includes what can be referred to as substances, chemicals, mixtures, items, materials, components, etc, depending on the application. Note that the substance concept does not include a specification of a particular physical instance of the substance. For example the substance named Carbon dioxide refers to any existing carbon dioxide. Specific instances of substances may be defined in applications, e.g. in a Process inventory the particular Carbon dioxide emitted from a specific plant is considered.

A substance may have many synonymous names, which are applied depending on the context. The use of CAS-numbers is one way of naming substances, but since CAS-numbers are not available for all substances, it is not sufficient to identify any substance. In product development it may be more efficient to use a company specific name of a substance. In addition, substances may be categorised in substance categories, in order to be easily dealt with, e.g. many specific types of steel can be put in one steel category. Since no nomenclature has names for each conceivable substance an artificial id is given each substance for unique identification.

A substance may have many various properties – substance properties - such as molecular weight, tensile strength, recyclability, labelling requirements, restrictions on usage, ranking values, etc. The value may be relative or absolute, be quantitative or qualitative, and based on measurements, calculations and/or value choices. Some substance properties values are dependent on ambient conditions at the measurement, e.g. the boiling point is dependent on the atmospheric pressure. It may hence be necessary to specify more than one substance property parameter to sufficiently document the substance property. The description of the method applied to obtain the value of the property and other relevant information for transparent documentation is included in the substance property documentation concept. A substance property connects a substance with a substance property type.

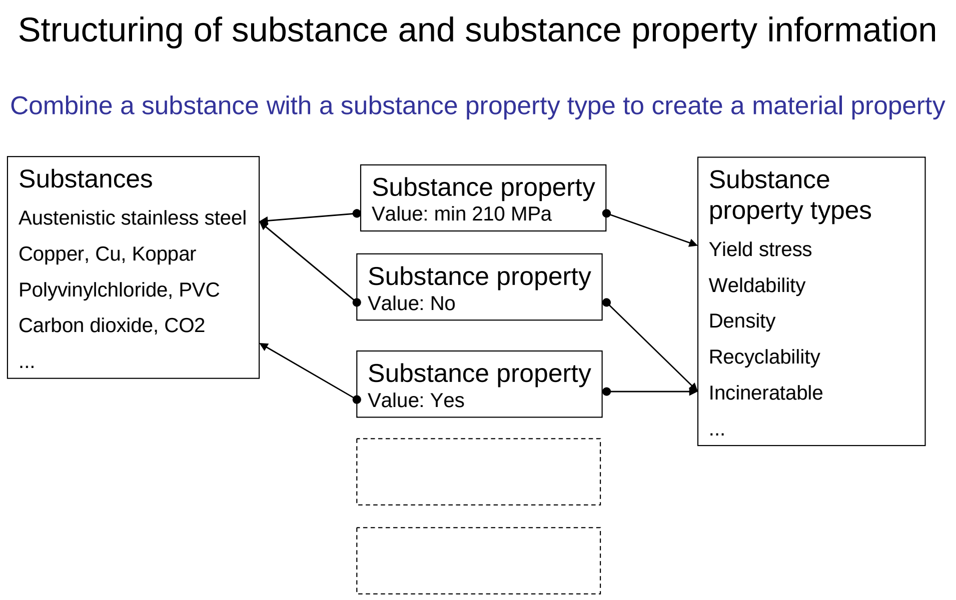

Figure 4 Example data and structure of substance property data linking a substance with a substance property type.

A substance property type is a definition of a property. It consists mainly of a name of the property, a description of its physical representation, the acceptable observation method(s) to obtain values of the property, and the valid value range of the property. In addition properties may be categorised in property categories such as physical, chemical, and toxicological properties to facilitate efficient information management.

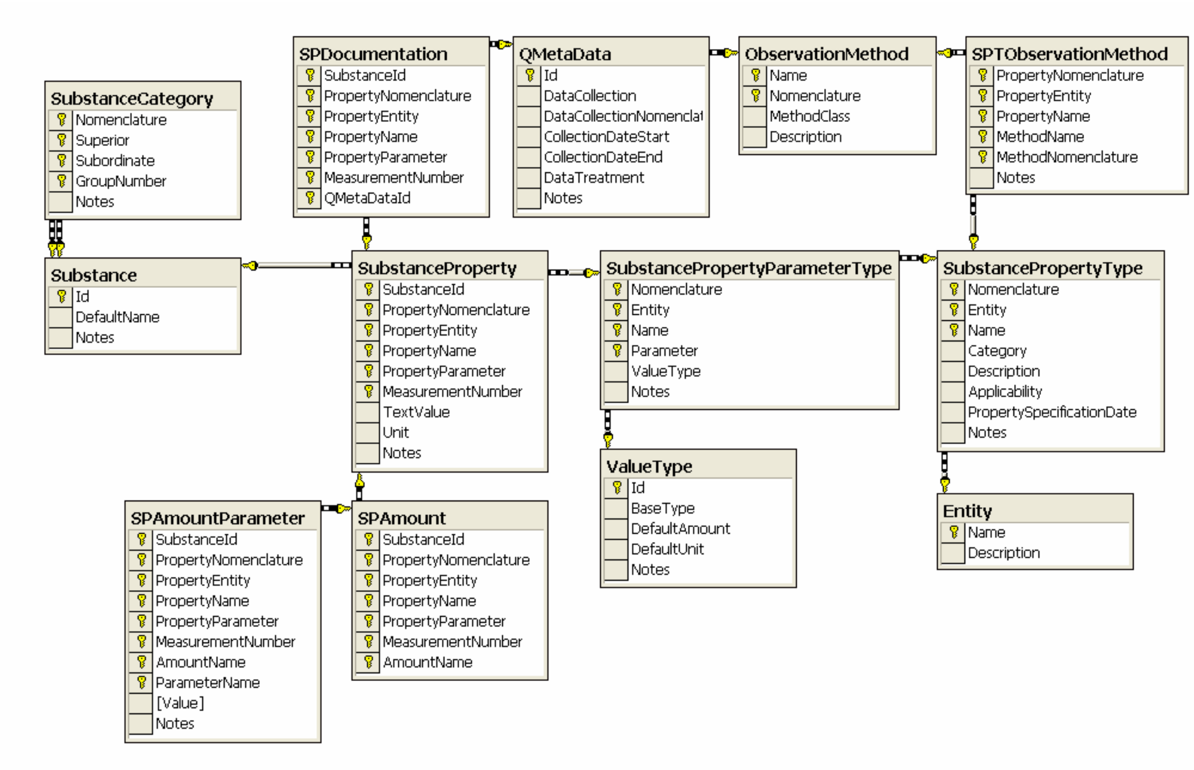

Figure 5 The tables in the IMPRESS data format for storage of substance and substance property data.

Figure 5 show the main tables representing the concepts substance and substance properties. This format is applicable to any user application involving substance information, e.g. as in materials within DfE, as elementary flows of a process in an LCI, as a chemical interacting with the environment an environmental impact assessment model, etc.

5.2.1 Substance

A table for storing unique identifiers of substances. A default name may be given the substance in e.g. smaller applications where only one nomenclature is needed with only one name per substance

Fields:	**Id**	Artificial identification number to uniquely identify the substance.
	DefaultName	Intended for storing the most commonly used name of the substance or specifically for storage of the substance

		name for applications where only one substance nomenclature is used.
	Notes	Additional relevant information about the identity of the substance.
Foreign keys:	N/A	

5.2.2 SubstanceCategory

Substances may be logically categorised into hierarchical categories. This is for example useful when grouping the materials.

Fields:	**Nomenclature**	The name of the nomenclature used to name and categorise the substance.
	Superior	Reference to the superior substance in a binary relation. The top of the category is identified by superior ID = subordinate ID.
	Subordinate	Reference to the subordinate substance in a binary relation.
	GroupNumber	Sequential number that enables a substance to participate in several different groups within a nomenclature. Without this group number the sub-trees of a component represented in different branches would be merged.
	Notes	Additional relevant information.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.
	Superior	Referencing the table Substance.
	Subordinate	Referencing the table Substance.

5.2.3 SubstancePropertyType

Substance properties shall like all other properties be selected within a predefined range of types. The available substance property types shall be defined in this table. In OMNIITOX a substance property type is referred to as a *substance property specification*.

Fields:	**Nomenclature**	Name of nomenclature in which the property was defined. Examples: “REPID”, “OMNIITOX”, “RAVEL”.
	Entity	Physical or other type of entity that the property represents. Examples: “Weight”, “Length”, “Price”, “Concentration of hydrogen ions”. In OMNIITOX this is the property names as defined in the “minimum requirement list” and “extended requirement list”.
	PropertyName	Name of a property. Examples: “Measured weight”, “Target weight”, “Declared weight”, "Boiling point", “Acidic dissociation constant”. In OMNIITOX the name is created by concatenating the entity name and the observation method(s) specified
	Category	Name of the property category to facilitate efficient information management. Examples: “Classification”, “Economical”, "Ecotoxicological”, “Physical”.
	Description	Explanatory description of the property type
	Applicability	Description of the domain of application for the property.
	PropertySpecificationDate	Date when the substance property type was specified.
	Notes	Any additional relevant information.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.
	Category, Entity	Referencing the table SubstanceEntity.
	Class	Referencing the table SubstancePropertyClass.

5.2.4 SPTObservationMethod

This table stores data on which methods are allowed to obtain a value for the property. It is a bridging table that connects SubstancePropertyType to ObservationMethod, see ObservationMethod chapter 5.17.22.

5.2.5 SubstancePropertyParameterType

A substance property may be specified by more than one parameter, e.g. to specify a boiling point it may be required to also specify the value the atmospheric pressure at the measurement.

Fields:	**Nomenclature**	Name of nomenclature in which the property was defined. Examples: “REPID”, “OMNIITOX”, “RAVEL”.
	Entity	Physical or other type of entity that the property represents. Examples: “Weight”,
	PropertyName	Name of a property. Examples: “Measured weight”, “Target weight”, “Declared weight”, "Boiling point", “Acidic dissociation constant”.
	PropertyParameter	Name of a property parameter. The ‘main’ parameter is given the same name as the property type. Condition paramters must be named differently. Examples: “Acidic dissociation constant”, “Temperature”, “pH”, “Atmospheric pressure”.
	ValueType	Defining the unit and valid range of values for the substance property parameter.
	UnitSpecification	Explanation to the resulting unit. For example how the unit is derived.
Foreign keys:	Nomenclature, Entity, Name	Referencing the table SubstancePropertyType.
	ValueType	Referencing the table ValueType.

5.2.6 SubstanceProperty

The SubstanceProperty table stores information of substance characteristics. A substance property connects a substance with a substance property type. Any number of substance properties for a specific substance may be stored and values for each type of property may be obtained through different measurements. If the value of the property is numeric it is stored in the Value field of the SPAmountParamter table. See also Amount and Parameter chapters 5.13.1 and 5.13.2 respectively. The documentation of the meta data about the substance property is stored in the QMetaData table linked via the SPDocumentation table, see QMetaData chapter 5.15.

Fields:	**SubstanceId**	Unique identifier of the substance**.**
	PropertyNomenclature	Name of nomenclature in which the property was defined.
	PropertyEntity	Physical or other type of entity that the property represents.
	PropertyName	Name of a property.
	PropertyParameter	Name of a property parameter.
	MeasurementNumber	Number of the measurement. To allow the storage of several independent measurements. The table IdSeq can be used to easily retrieve unique numbers for each component. (This should not be confused with samples in a measurement series. In this case it is the same measurement and the distinction is made clear by the amount parameter.)
	TextValue	Holds the data for the specific parameter if the amount structure is not used for this parameter, (see description above). Must be NULL or empty string if value type is “Numeric”.
	Unit	The unit of the parameter. Only necessary if different from the default unit as specified by the value type of the substance property parameter type.
	Notes	Any additional relevant information.

Foreign keys:	SubstanceId	Referencing the table Substance.
	PropertyNomenclature, PropertyEntity, PropertyName	Referencing the table SubstancePropertyType.
	Unit	Referencing the table Unit.

5.3 Component structure and material content of products

This chapter refers to the physical component structure and material content of products. The data format for management of the procedural information on creating and evaluating designs of products and components is described in Design chapter 5.4.1.

It is necessary to know what part of the product is causing the impact to be able to know what environmental effects a design change will have. The internal structure of the product is therefore broken down in a component structure. A component is a physical part of the structure. The composition concept represents what a component consists of in terms of sub-components. It may further be relevant to know how the subcomponents are connected to each other and what role they have in the product structure. This is represented by the conjunction concept. Materials are at the lowest level in the structure, i.e. materials are not broken down further into sub-components, see figure 6 and 7.

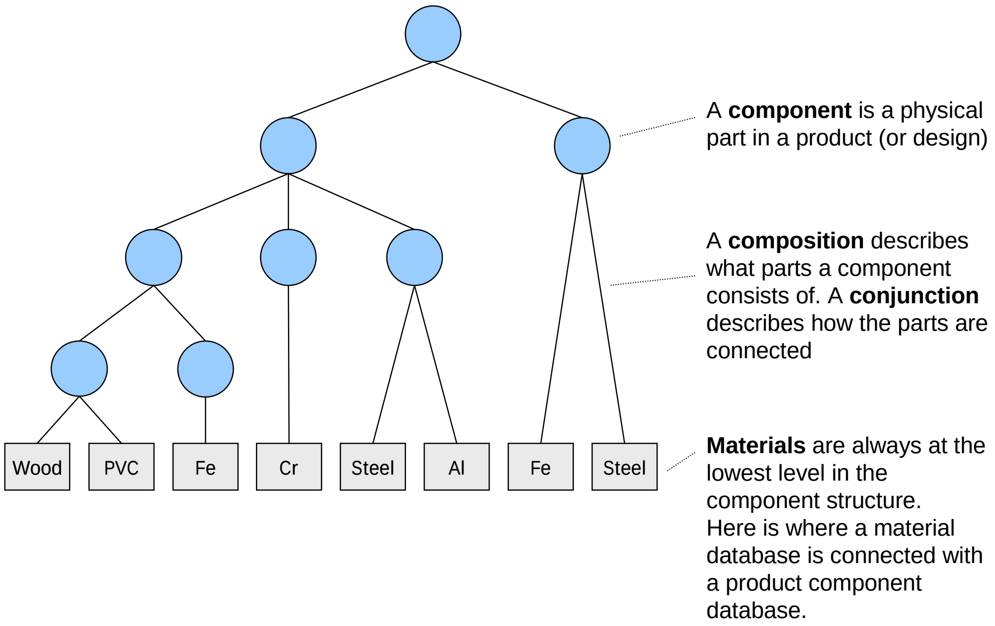

Figure 6 Conceptual model showing the main concepts related to the component structure of a product or design.

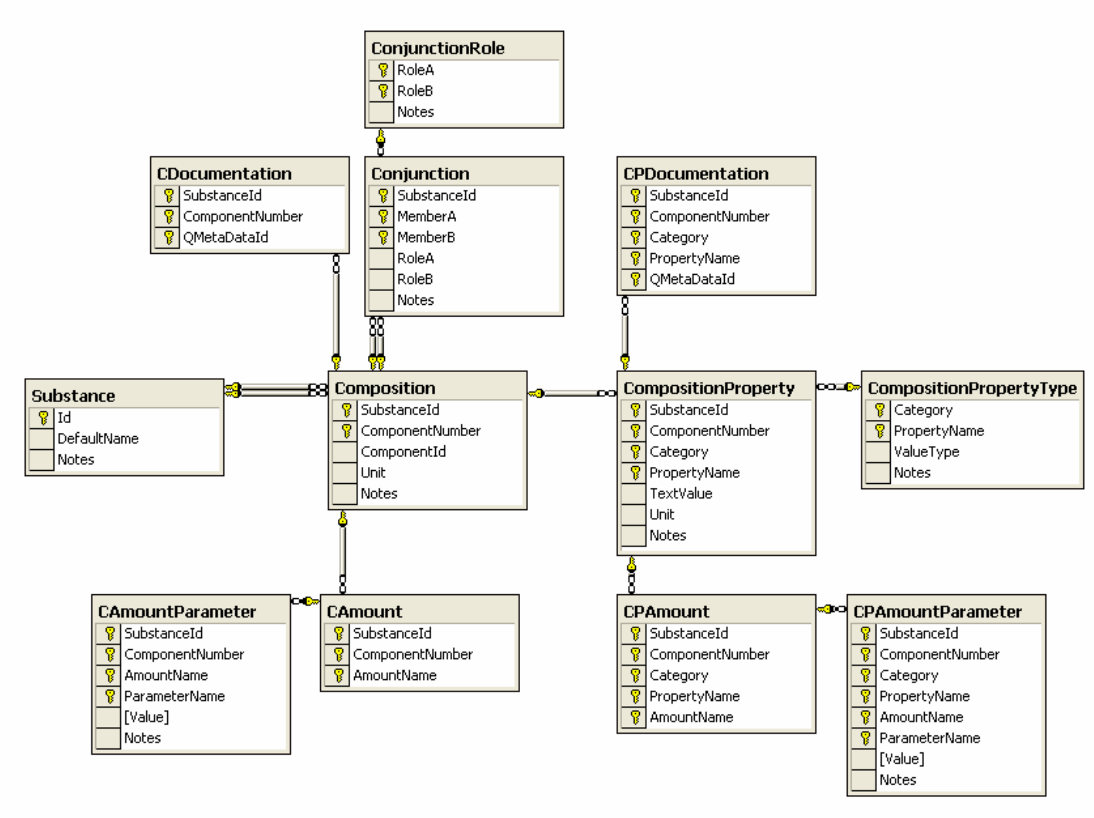

Figure 7 The main tables of the IMPRESS data format representing component structure and material content of a product or design. Note that both components and materials are stored as substances.

5.3.1 Composition

Component structures are built by relating substances two by two. This table describes how the different components are put together from (sub-)components and/or materials. Components, subcomponents, and materials are all different types of substances, see Substance chapter 5.2.1. If the parts in a composition are subcomponents, i.e. components themselves, the number of pieces of a subcomponent is stored in the field Value in the table CAmountParameters. See also Amount and Parameter chapters 5.13.1 and 5.13.2 respectively. If the part is a material, the weight of the material in the component is stored in CAmountParameters.Value instead. If the same subcomponent or material has different properties in a component, for example depending on different physical location, several compositions may have to be created. For that reason there is a component number in the composition table where each parent component gives their ‘children’ a sequential number. Metadata about the Composition is stored in QMetaData linked via the table CDocumentation, see QMetaData chapter 5.15.

Fields:	**SubstanceId**	A reference to the substance being the parent in the binary relation.
	ComponentNumber	Each parent is giving their children sequence numbers. The table IdSeq can be used to easily retrieve unique numbers for each component.
	ComponentId	A reference to the material or subcomponent being the child in the binary relation.
	Unit	A unit of the quantity supplied. Examples: “Kilogram”, “Pieces”
	Notes	Additional relevant information about the composition.
Foreign keys:	SubstanceId	Referencing the table Substance.
	ComponentId	Referencing the table Substance.
	Unit	Referencing the table Unit.

5.3.2 CompositionProperty

Some emergent properties at component level may not be calculated from the sum of the included materials and subcomponents. An example is 'forbidden material', a rule that may apply for a toxic substance e.g. cadmium, except when used in a battery. Another property that varies greatly depending on where a component or material is used is the length of life. A screw that keeps two railway wagons together is naturally fatigued earlier than the same screw holding a painting. A numeric value of a property is stored in the Value field of the CPAmountParameter table. See also Amount and Parameter chapters 5.13.1 and 5.13.2. Metadata about the composition property is stored in the QMetaData table linked via the CPDocumentation table, see QMetaData chapter 5.15.

Fields:	**SubstanceId**	A reference to the substance being the parent in the binary relation.
	ComponentNumber	Each parent is giving their children sequence numbers.
	Category	A category to which the property relates, e.g. 'Overrule material restriction'.
	PropertyName	Name of the property that further specifies the category. Example: 'Do not include with summation of toxic summation'.
	TextValue	If the value type (see ValueType) for the property is either ‘String’ or ‘StringNomenclature’ the value is given in this field. If the value type is ‘Numeric’ the value can be found in the table CPAmountParameters.
	Unit	The unit for the property.
	Notes	Additional relevant information about the composition property.
Foreign keys:	SubstanceId, ComponentNumber	Referencing the table Composition.
	Category, PropertyName	Referencing the table CompositionPropertyType.
	Unit	Referencing the table Unit.

5.3.3 CompositionPropertyType

Composition properties should be selected within a predefined range of types, e.g 'Overrule material restrictions', etc.

Fields:	**Category**	A category to which the property relates, e.g. 'Overrule material restriction'.
	PropertyName	Name of the property that further specifies the category. Examples 'Do not include with summation of toxic summation'.
	ValueType	Defines the valid range of values for the composition

		property. See table ValueType.
	Notes	Additional relevant information about the substance.
Foreign keys:	ValueType	Referencing the table ValueType.

5.3.4 Conjunction

To aid in the assessment of e.g. end of life phase for assembly products, the model allows for storage of information on how composites (Composition) relates. This is useful for identification of whether two components in a component are e.g. glued, welded, snapped or merged together. It is also useful when a material in a component has the role of being a surface treatment on another material in the component.

Fields:	**ComponentId**	A reference to the component in which a conjunction between two of its subcomponents or materials will be specified.
	MemberA	Reference to one of the subcomponents or materials within the component.
	MemberB	Reference to the other of the two subcomponents or materials within the component.
	RoleA	This is the role that member A has in the conjunction. Examples: “Base material”, “Glue”, “Coating”, “Surface treatment”, “Nut”.
	RoleB	This is the role that member B has in the conjunction.
	Notes	Additional relevant information about the conjunction.
Foreign keys:	ComponentId, MemberA	Referencing the table Composition.
	ComponentId, MemberB	Referencing the table Composition.
	RoleA, RoleB	Referencing the table ConjunctionRole.

5.3.5 ConjunctionRole

This table specifies the roles of the two members in a conjunction.

Fields:	**RoleA**	Possible roles for member A in conjunction. Examples: “Base material”, “Glue”, “Coating”, “Surface treatment”, “Nut”.
	RoleB	Possible roles for member B in conjunction.
	Notes	Additional relevant information about the roles-pair of a conjunction.
Foreign keys:	N/A	

5.4 Design

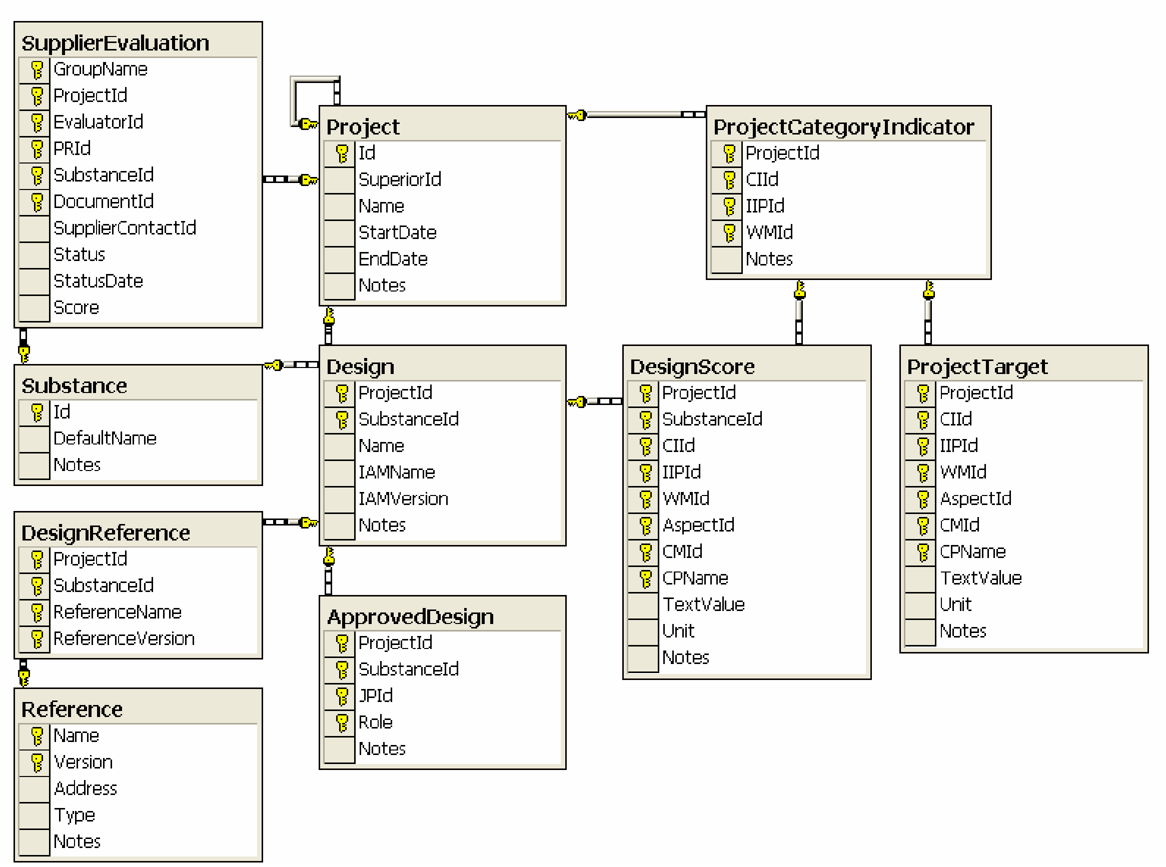

Figure 8 The tables in the IMPRESS data format representing product design. In this perspective Substance is representing the physical product or component.

To improve the environmental performance of a product is a matter of product development and design – hence design for environment. The design concept represents the modelling of physical product and component structures. The modelling work is done within a project context where the project structure describes how the task of building a product is organizationally divided into pieces, See Project chapter 5.1.1. Design alternatives are held together in a group by relating it with the same project. The component structure describes how the product is physically divided into parts, see Component structure and material content of products chapter 5.3. The task of a project can be fulfilled by many different choices of components. Each of these components is

connected to the project via the table Design. In this design perspective the table Substance functions as data format for the physical product or component, see Substance chapter 5.2.1. Normally only one design is chosen for each project but if many alternate components have been tested, the table ApprovedDesign points out which component shall be used in the project. The design data is often generated and stored elsewhere in a company's existing PDM system. Therefore, a design generally has associated information about the same design outside of the information platform, in e.g. a CAD or PDM file. It is possible to reference such outside occurrences of the design through DesignReference and Reference.
When assessing a product or component structure it is done in relation to an indicator choosen for the specific project. A target value may be set for the indicator and the resulting value of the design may be stored as a design score DesignScore. This is to allow different score measurement methods to be used in different projects that use the same (physical) design. The choice of method is project specific, not design specific. Figure 8 show the tables related to product design.

5.4.1 Design

This table specifies relations between specific physical designs of products or components to projects. A project may be associated with many different designs and design variants.

Fields:	**ProjectId**	References the project with which the design is related.
	SubstanceId	References the Substance, i.e. the physical product or component that is being designed.
	Name	A name of the design.
	Notes	Additional relevant information.
Foreign keys:	ProjectId	Referencing the table Project.
	SubstanceId	Referencing the table Substance.

5.4.2 DesignScore

Table for storing the environmental performance of designs in terms of impact on indicators.

Fields:	**ProjectId**	References the project with which the design is related.
	SubstanceId	References the physical product or component.

	CIName	Reference to the indicator that defines the type of impact caused by the product or component that is under consideration.
	IIPName	The impact indication principle (IIP), i.e. the principle applied to choose the indicator.
	IIPVersion	The version of the IIP.
	WMName	Reference to the weighting method (WM) applied when comparing impact on several indicators.
	WMVersion	The version of the WM.
	AspectId	Reference to the Aspect, i.e. the definition of what in the product or component that is causing the impact on the indicator under consideration.
	CMName	Reference to the characterisation method (CM) applied that is a model of the quantitative relation between the Aspect and the Indicator under consideration.
	CMVersion	The version of the CM
	CPName	Reference to the characterisation parameter (CP) used, e.g. a characterisation factor.
	ValueType	Defines the different possible values the score value may take (see ValueTypeDefinition). Examples: “Boolean”, “Numeric”, “Percent”.
	TextValue	Holds the data for the specific parameter if the amount structure is not used for this parameter, (see description above). Examples: ”Blue", "Yes", "Allowed”
	Unit	The unit of the design score.
	Notes	Any additional relevant information.
Foreign keys:	ProjectId, SubstanceId	Referencing the table Design.
	ProjectId, CIName, IIPName, IIPVersion, WMName,	Referencing the table ProjectCategoryIndicator.

	WMVersion	
	ValueType	Referencing the table ValueType
	Unit	Referencing the table Unit.

5.4.3 DesignReference

The description of a design within the information system will often have an associated description of the same design in another information system, e.g. a CAD- or PDM-file system. This bridging table connects a Design to one or several References.

Fields:	**ProjectId**	References the project with which the design is related.
	SubstanceId	References the Substance, i.e. the physical product or component.
	ReferenceName	References the name of the reference.
	ReferenceVersion	References the version of the reference.
Foreign keys:	ProjectId, SubstanceId	Referencing the table Design.
	ReferenceName, ReferenceVersion	Referencing the table Reference.

5.4.4 Reference

A description of the external reference to a design. The actual, physical design will be documented somewhere else and need not be repeated within the database.

Fields:	**Name**	A name of the reference.
	Version	A version of the reference.
	Address	An address to the reference. The address may be the location of a file within a computer or within a network, or it may be a reference to a physical file in e.g. a document shelf.
	Type	This field describes the type of reference described in Address, e.g. whether the address refers to a file, etc.
	Notes	Additional relevant information.
Foreign	N/A	

keys:		

5.4.5 ApprovedDesign

If many alternative designs are available for a project, the selection of the version that has passed a approval process is stored in this table.

Fields:	**ProjectId**	References the project with which the design is related.
	SubstanceId	References the Substance, i.e. the physical design.
	JPId	Reference to the (juridical) person approving the design.
	Role	The role that the approving person has in the project.
	Notes	Additional relevant information.
Foreign keys:	ProjectId, SubstanceId	Referencing the table Design.
	PersonId	Referencing the table JuridicalPerson.

5.4.6 SupplierEvaluation

The RAVEL workbench supports the DFE process all through the design process, such as the purchaser's evaluation of potential suppliers.

Fields:	**GroupName**	A name of a group of suppliers to evaluate.
	ProjectId	Reference to the project in which the evaluation occurs.
	EvaluatorId	Reference to the person responsible for performing the evaluation.
	PRId	Reference to information about a supplier.
	SubstanceId	Reference to the material or component potentially supplied by the supplier.
	DocumentId	Reference to the evaluation questionnaire document. When the questionnaire is sent, a reference to the document is stored in this position. When the questionnaire is received this field instead references the filled in reply.
	SupplierContactId	Reference to the contact person at the supplier.

	Status	This field contains 'Not submitted' before a document is submitted to the supplier, 'Submitted' when submitted and not received, 'Received' when received and not given a score, and 'Evaluated' when also given a score.
	StatusDate	Holds the date when the value in the field Status was last updated.
	Score	When the different potential suppliers are evaluated, they get a score.
	Notes	Additional relevant information.
Foreign keys:	ProjectId	Referencing the table Project.
	SubstanceId	Referencing the table Substance.
	PRId	Referencing the table PhysicalReference
	EvaluatorId	Referencing the table JuridicalPerson.
	DocumentId	Referencing the table Document
	SupplierContactId	Referencing the table JuridicalPerson.

5.5 Process Scenarios

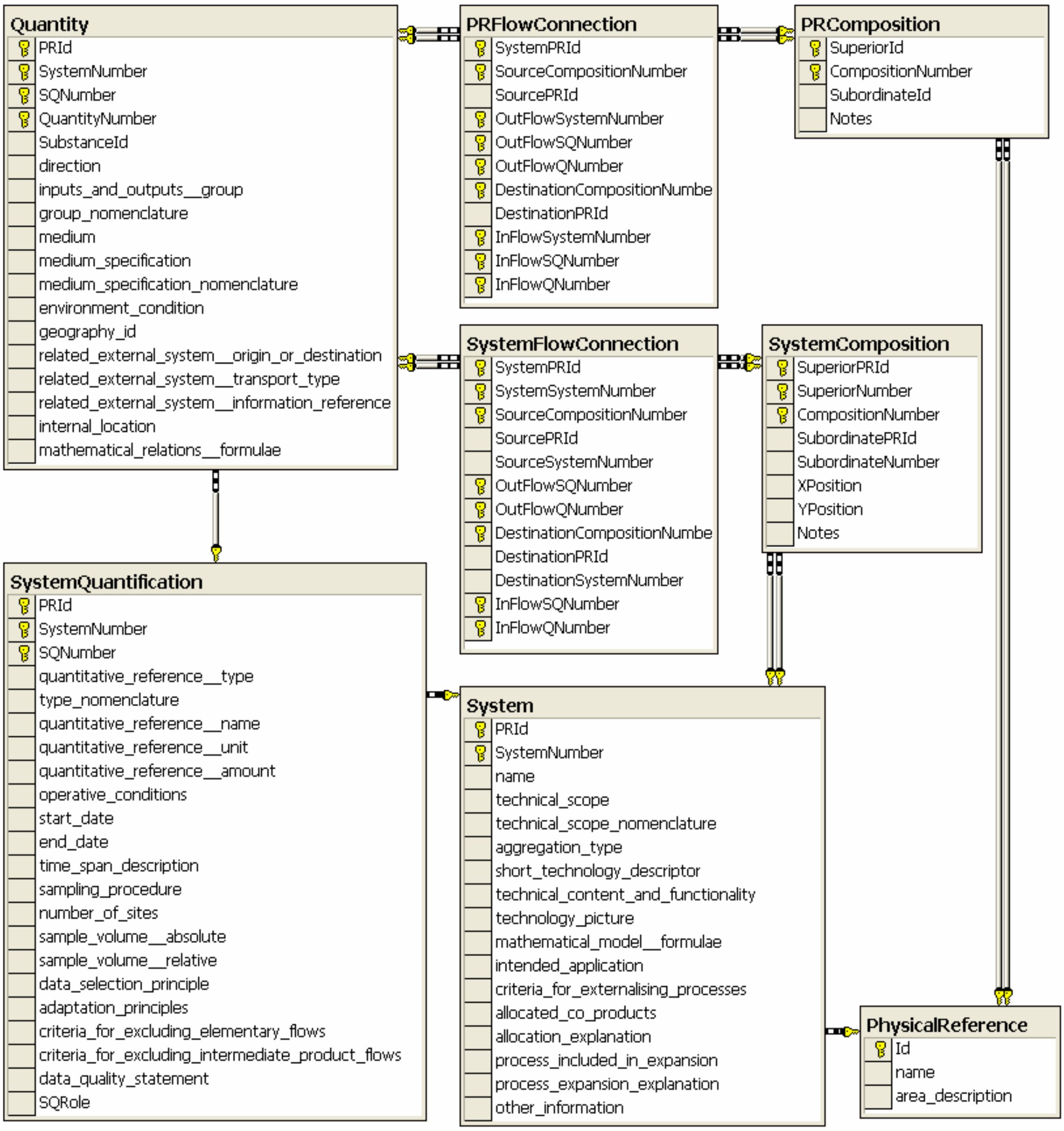

Figure 9 The main tables of the IMPRESS data format for related to process data.

The data format for process information can make full use of the capabilities of the LCA data documentation format of ISO/TS 14048 of handling environmental management and life cycle scenarios.

In this integrated data format there is a distinction between the reference to the real physical object (Physical reference) and the reference to the model of that object (System). The reference to the actual object is stored in the table PhysicalReference and the reference to the model of the real object is stored in the table System. When measurements are performed to retrieve values on the inputs and outputs of the object, information about the measurement situation is stored in SystemQuantification and the values from the measurements are stored in Quantity. The same physical object can be described by any number of models. The same model can be quantified many times, and any number of values can be registered each time a system is quantified.
This model is designed to support requirements on an Environmental Management System. Reports from EMS systems are often generated with a given time-interval. Much information in these report do not change often while other data changes frequently. The data on the physical reference will most likely be reused for many years and is likely to be referenced by different system models. Once a model is defined and documented in the table System will also most likely be reused many times. The details on what inputs and outputs are considered in the model is stored as a SystemQuantification template, templates are distinguished from actual measurements by stating the role of the system quantification in the field SQRole. All quantities that refer to a SystemQuantification denoted as a template is included in the system model. In this way a system model can be defined in detail. When documenting data actual measurements the model template can be copied to a system quantifications and all that needs to be added is the data for the specific measurement. For each report new measurements have to be performed that generates quantities on inputs and outputs from the system. By using this structure data format the environmental coordinators get a better control over which parts that he/she is reusing and which information that has to be collected.

Life cycle scenarios can be modelled through linking an output of a system model to the input of another system model. Systems can be aggregated into a composed system in the table SystemComposition and the specification of which flows are connected between systems are defined in the table SystemFlowConnection. Data on connected physical references can be documented in a corresponding way in the tables PRComposition and PRFlowConnection.

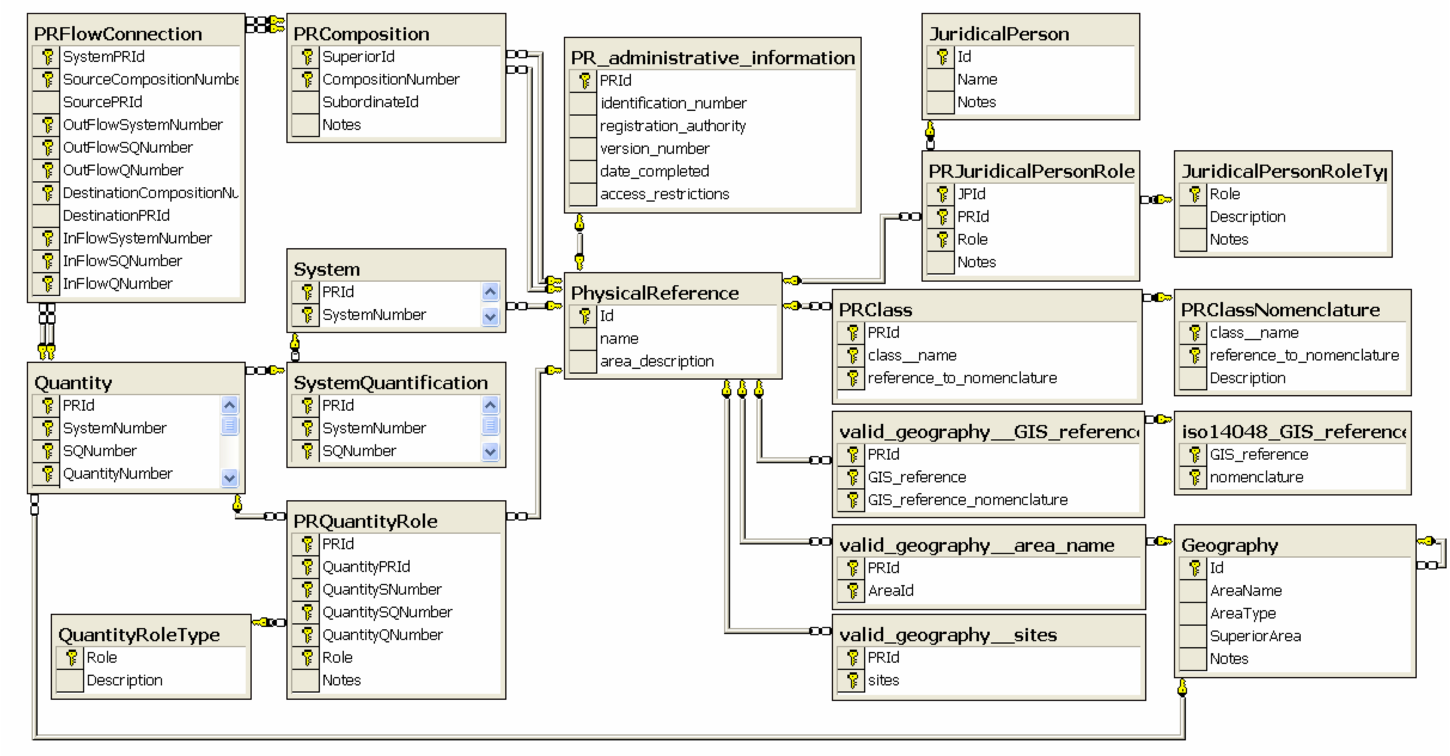

Figure 10. The tables in the IMPRESS data format related to the physical reference concept.

5.5.1 Physical Reference

The physical reference refers to an actual physical object in the real world. This should not be confused with system models which are documented in the System table.

Fields:	**Id**	A reference to the physical reference.
	Name	Name of the physical reference
	Area_description	General description of the actual geographical area considered.
Foreign keys:	N/A	

5.5.2 PR_administrative_information, System-_administrative_information, SQ_administrative_information, Quantity_administrative_information

To facilitate the administration of physical references, systems, system quantifications and quantities documented according to ISO/TS 14048, general administrative information is required since information will be exchanged between data generators and data users, it will be stored in databases and it will be administrated within different information systems.
Here the specification for the table PR_administrative_information follows. The tables System_administrative_information, SQ_administrative_information, and Quantity_administrative_information are defined in an analogous way. To follow the documentation field specification of LCA datasets according to ISO/TS 14048 the table PRJuridicalPersonRole shall be used to specify the roles data_commissioner, data_generator, data_commissioner, and copyright. This is done since many different persons can perform those tasks. The reference to publication as specified in ISO/TS 14048 is done by storing information in the Document table and creating references via the Context structure, see Document chapter 5.18.1 and Context chapter 5.3.15.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	Identification_number	A unique number, within the context of the registration authority, used to identify the process.
	Registration_authority	Identification of the registration authority for the identification number of the process.
	Version_number	Can be used to identify updates of data for a specified process.
	Date_completed	The date at which the data of the process were finally completed, edited or updated.
	Publication	Reference to a printed or otherwise stable and published literature source where the original copy of

		this document may be found.
	Access_restrictions	An unambiguous flag indicating how the information may be disclosed both outside and within the information system.
Foreign keys:	PRId	Referencing the table PhysicalReference

5.5.3 PRComposition

A physical reference can be broken down into smaller pieces, a system can be composed of many models, and a quantification of a system can be compiled of a set of quantifications. From a bottom up perspective physical references can also be considered as parts of a larger physical reference, and similarly for system models. A physical reference may only occur once. This differs from the possibility to create an aggregated system model where a subordinate model may occur several times.

Fields:	**SuperiorId**	A reference to the physical reference being the parent in the binary relation.
	SubordinateId	A reference to the physical reference being the child in the binary relation.
	Notes	Any additional relevant information.
Foreign keys:	SuperiorId	Referencing the table PhysicalReference.
	SubordinateId	Referencing the table PhysicalReference.

5.5.4 PRFlowConnection

Physical references can be connected to each other by their flows. An output of PR is connected to an input of another. In this way we can create references to a network of PRs with products, materials, energy, etc, flowing between them. The systems participating in the network will together constitute a system as described above in PRComposition chapter 5.5.3. All non-connected flows are considered to pass the boundaries of the superior PR. Any PR can be broken down into smaller PRs or itself be a part of a larger PR. This means that the documentation of physical references can have hierarchical structures of arbitrary depths.

Fields:	**SystemPRId**	Identification number to uniquely identify the physical reference of which the superior system is a model.
	SourcePRId	Identification number to uniquely that identifies the child PR that is the source of the flow
	SourceSystemNumber	Sequential number that identifies the child system that is the source of the flow.
	OutFlowSQNumber	Identification number to uniquely identify the quantification of the system that identified the output of current interest.
	OutFlowQNumber	Identification number to uniquely identify an output quantity of the source system.
	DestinationPRId	Identification number to uniquely that identifies the child PR that is the destination of the flow
	DestinationSystemNumber	Sequential number that identifies the child system that is the destination of the flow.
	InFlowSQNumber	Identification number to uniquely identify the quantification of the system that identified the input of current interest.
	InFlowQNumber	Identification number to uniquely identify an input quantity of the source system.
Foreign keys:	SystemPRId, SourcePRId	Referencing the table PRComposition.
	SystemPRId, DestinationPRId	Referencing the table PRComposition.
	SourcePRId, SourceSystemNumber, OutFlowSQNumber, OutFlowQNumber	Referencing the table Quantity.

	DestinationPRId, DestinationSystemNumber, OutFlowSQNumber, OutFlowQNumber	Referencing the table Quantity.

5.5.5 PRClass, SystemClass, SQClass

Physical references, systems and system quantifications can be classified. A class makes it easy to search and identify the data. Compared to a name, a class gives an unambiguous structure, allowing easy access for users to all data within an area of interest. For any given process, several classes can be used, but within each class the process can only belong to one name in the class.

Here the specification for the table PRClass follows. SystemClass and SQClass are defined in the same way.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	Class_name	Specification of the class name to which the physical reference belongs.
	Reference_to_ nomenclature	Specification of the nomenclature from which the name is chosen.
Foreign keys:	PRId	Referencing the table PhysicalReference.
	Reference_to_ nomenclature	Referencing the table PRClassNomenclature.

5.5.6 PRQuantityRole, SystemQuantityRole, SQQuantityRole

Each quantity has a specific role in relation to the Physical Reference, the System, and the System Quantification.

The role of a **quantity** in relation to a **physical reference** is stored in **PRQuantityRole**.

The role of a **quantity** in relation to a **system** is stored in **SystemQuantityRole**.

The role of a **quantity** in relation to a **system quantification** is stored in **SQQuantityRole**.

Here the specification for the table PRQuantityRole follows. SystemQuantityRole and SQQuantityRole are defined in the same way.

Specifically a LCA a reference flow (functional unit) is defined in the table SQQuantityRole where the Role is set to “Reference flow”.

Fields:	**PRId**	Reference to the physical reference.
	QuantityPRId	Reference to the physical reference of the quantity.
	QuantitySNumber	Reference to the system of the quantity.
	QuantitySQNumber	Reference to the system quantification of the quantity.
	QuantityQNumber	Reference to the quantity.
	Role	Role of the quantity in relation to the physical reference. Example “Base line product”, “Auxiliary energy supply”, “Restricted emission”.
	Notes	Any additional relevant information.
Foreign keys:	PRId	Referencing the table PhysicalReference.
	QuantityPRId, QuantitySNumber, QuantitySQNumber, QuantityQNumber	Referencing the table Quantity.
	Role	Referencing the table QuantityRoleType

5.5.7 Valid_geography__area_name

This table specifies the geographical area that the physical reference is located in.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	AreaId	Unique identifier of geographical area in the Geography nomenclature table
Foreign keys:	PRId	Referencing the table PhysicalReference.
	Area_name	Referencing the table Geography.

5.5.8 Valid_geography__GIS_reference

This table specifies the position of the physical reference using a Geographical Information System (GIS) reference.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	GIS_reference	Name of the Geographical Information System (GIS) reference. Example 57°41’21N 11°58’44E
	GIS_reference_nomenclature	Name of the GIS nomenclature used. Example ISO 6709
Foreign keys:	PRId	Referencing the table PhysicalReference.
	GIS_reference, GIS_reference_nomenclature	Referencing the table iso14048_GIS_reference.

5.5.9 Valid_geography__sites

This table specifies one or more addresses to included sites.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	Sites	Address to included site.
Foreign keys:	PRId	Referencing the table PhysicalReference.

5.5.10 System

System is the central concept for system models of processes in the technosphere.

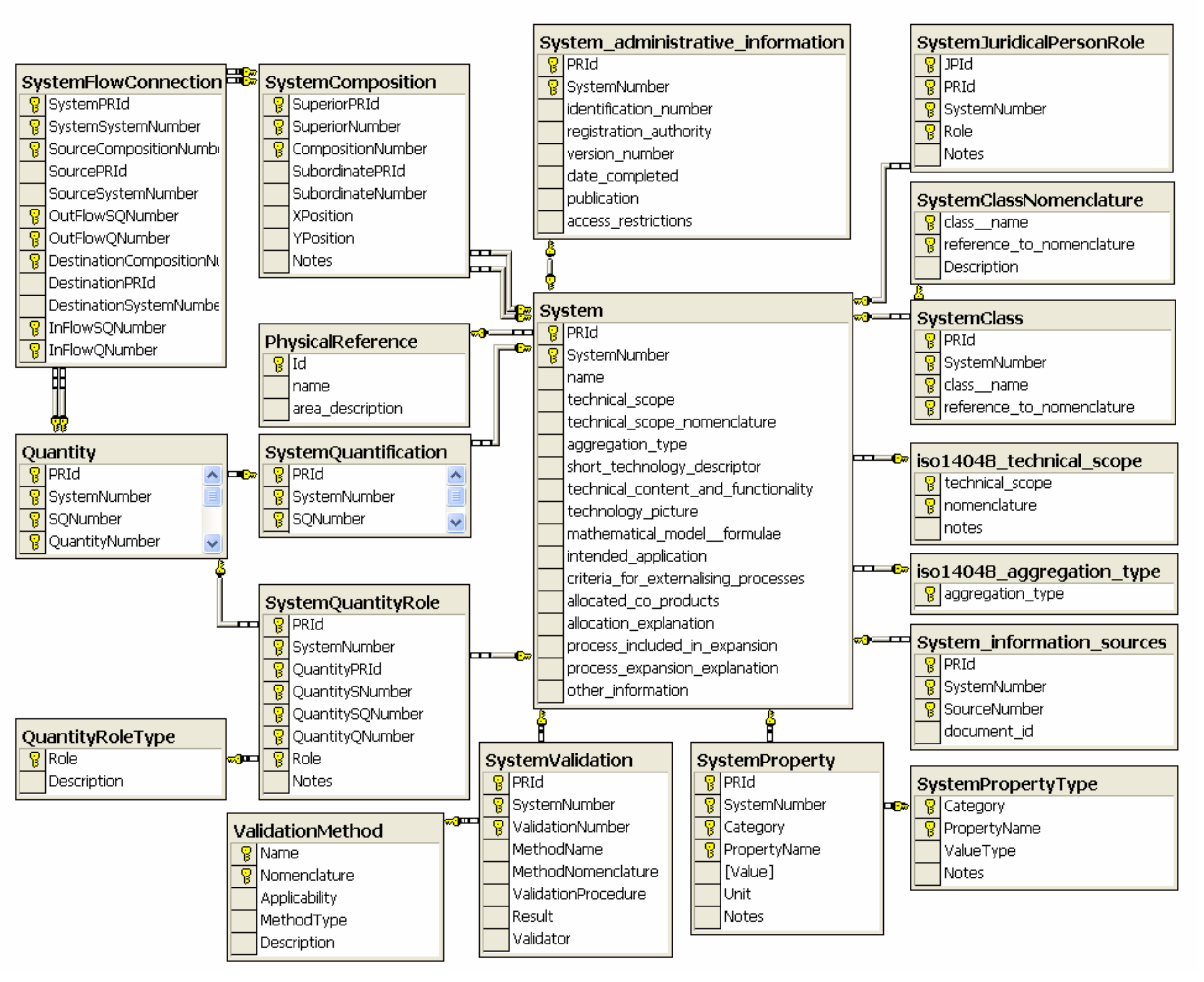

Figure 11The tables in the IMPRESS data format related to the system concept.

In this table references to different models of physical objects are stored.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	SystemNumber	Artificial identification number to uniquely identify the system.
	Name	Descriptive name of the model of the technical system. Examples: “Combined heat and power plant with support system” and “Long distance transportation by heavy truck”.
	Technical_scope	A short general description of the technical scope of the process in terms of the operation(s) included in the data, using a nomenclature. This may be one single operation or several operations covering the full lifecycle of a product, e.g. gate-to-gate or cradle-to-grave.
	Technical_scope_nomenclature	The nomenclature used for the technical scope
	Aggregation_type	This is used to indicate aggregated unit processes, i.e. representing averages of several processes providing the same function (horizontal) or the sum of several interconnected processes (vertical), represented by a nomenclature.
	Short_technology_descriptor	Short descriptor for the included technology.
	Technical_content_and_ functionality	Detailed description of the individual included operations and how they are technically and materially related. When data are aggregated, and the process within the aggregation is not presented, a description of the process within the aggregation should be given here.
	Technology_picture	Graphical representation of the technology, e.g. a graphical flowchart of the process.
	Mathematical_model__ formulae	Specification of the formulae in the mathematical model.
	Intended_application	Documentation of the intended application and a

		general description of the task. This may also include an explanatory documentation of the function of the system.
	Criteria_for_externalising_ processes	Description of the criteria or the principles that have been used for externalizing technical subsystems. This should include justifications and can include informative descriptions of the excluded systems. The description clarifies the technical boundaries of the process.
	Allocated_co_products	The co-products that have been allocated.
	Allocation_explanation	Description of the allocations that have been performed with regard to the choice of allocation method, allocation procedures, and information used in the allocation.
	Process_included_in_ expansion	Specification of the systems that have been included in the process expansion.
	Process_expansion_ explanation	Description of the process expansions that have been performed with regard to choices made, information used, etc.
	Other_information	In addition to the overall documentation of the system some other information may be supplied regarding for instance, advice on how to use the system, recommendations on the applicability of the system, known limitations, etc.
Foreign keys:	PRId	Referencing the table PhysicalReference.
	Technical_scope, Technical_scope_nomenclature	Referencing the table iso14048_technical_scope.
	Aggregation_type	Referencing the table iso14048_aggregation_type.

5.5.11 SystemValidation, SQValidation

Documentation of any validations that have been performed on the model respectively on the gathered data describing the process. The validation can be part of a critical review of an LCA study. Validation refers to checks made when entering the data, checks made by the data generator, and checks by a third party. Knowledge of each individual validation and its result is of great importance for a data user or reviewer when assessing the reliability of data.
Here the specification for the table SystemValidation follows. The table SQValidation referring to validations of system quantifications is defined in the same way.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	SystemNumber	Identification number to uniquely identify the system.
	ValidationNumber	Identification number to uniquely identify the validation.
	MethodName	Name of the validation method.
	MethodNomenclature	The nomenclature of the method name.
	ValidationProcedure	Description of the actual procedure applied in the validation.
	Result	Description of the result of the validation. Also, if errors or missing data were identified, but no corrections were made to the data, then the validation findings should be given here.
	Validator	A reference to the identifier of the person performing the validation.
Foreign keys:	PRId, SystemNumber	Referencing the table System.
	Method, MethodNomenclature	Referencing the table ValidationMethod.
	Validator	Referencing the table JuridicalPerson

5.5.12 SystemComposition, SQCompilation

A system model can be broken down into smaller pieces, a system can be composed of many models, and a quantification of a system can be compiled of a set of quantifications. From a bottom up perspective a system can be considered as a part of a larger model. To express and store these structures the tables SystemComposition, and SQCompilation are available. There is a sequential ordering number for the children in all these tables, since the same system, or system quantification can be used several times in a superior system, and system quantification respectively.

Here the specification for the table SystemComposition follows. SQCompilation is defined in the same way. Compare with PRComposition chapter 5.5.3 and also SystemFlowConnection chapter 5.5.13.

Fields:	**SuperiorPRId**	A reference to the physical reference being the parent in the binary relation.
	SuperiorNumber	A reference to the system model number of the parent.
	CompositionNumber	A sequential number to the system model number of the child. This allows having the same system model occur several times in the same aggregated system model.
	SubordinatePRId	A reference to the physical reference being the child in the binary relation.
	SubordinateNumber	A reference to the system model number of the child.
	Notes	Any additional relevant information.
Foreign keys:	SuperiorId	Referencing the table PhysicalReference.
	SubordinateId	Referencing the table PhysicalReference.

5.5.13 SystemFlowConnection

Physical references and systems can be connected to each other by their flows. An output of a system is connected to an input of another system. In this way we can model a network of systems with products, materials, energy, etc, flowing between them. The systems participating in the network will together constitute a system as described above in SystemCompilation chapter 5.5.12. All non-connected flows pass the system boundaries of the superior system. Any system can both have an inner structure and participate in an inner structure. This means that the documentation of physical references and system models can have hierarchical structures of arbitrary depths.

Fields:	**SystemPRId**	Identification number to uniquely identify the physical reference of which the superior system is a model.
	SystemSystemNumber	Identification number to uniquely identify the superior system that contains the systems that are to be connected.
	SourceComposition-Number	Sequential number that identifies the child system that is the source of the flow.
	SourcePRId	Identification number to uniquely identify the physical reference of which the source system is a model.
	SourceSystemNumber	Identification number to uniquely identify the system that is the source of the flow.
	OutFlowSQNumber	Identification number to uniquely identify the quantification of the system that identified the output of current interest.
	OutFlowQNumber	Identification number to uniquely identify an output quantity of the source system.
	DestinationComposition-Number	Sequential number that identifies the child system that is the destination of the flow.
	DestinationPRId	Identification number to uniquely identify the physical reference of which the destination system is a model.

	DestinationSystemNumber	Identification number to uniquely identify the system that is the destination of the flow.
	InFlowSQNumber	Identification number to uniquely identify the quantification of the system that identified the input of current interest.
	InFlowQNumber	Identification number to uniquely identify an input quantity of the destination system.
Foreign keys:	SystemPRId, SystemSystemNumber, SourceComposition-Number	Referencing the table SystemComposition.
	SystemPRId, SystemSystemNumber, DestinationComposition-Number	Referencing the table SystemComposition.
	SourcePRId, SourceSystemNumber, OutFlowSQNumber, OutFlowQNumber	Referencing the table Quantity.
	DestinationPRId, DestinationSystemNumber, OutFlowSQNumber, OutFlowQNumber	Referencing the table Quantity.

5.5.14 SystemQuantification

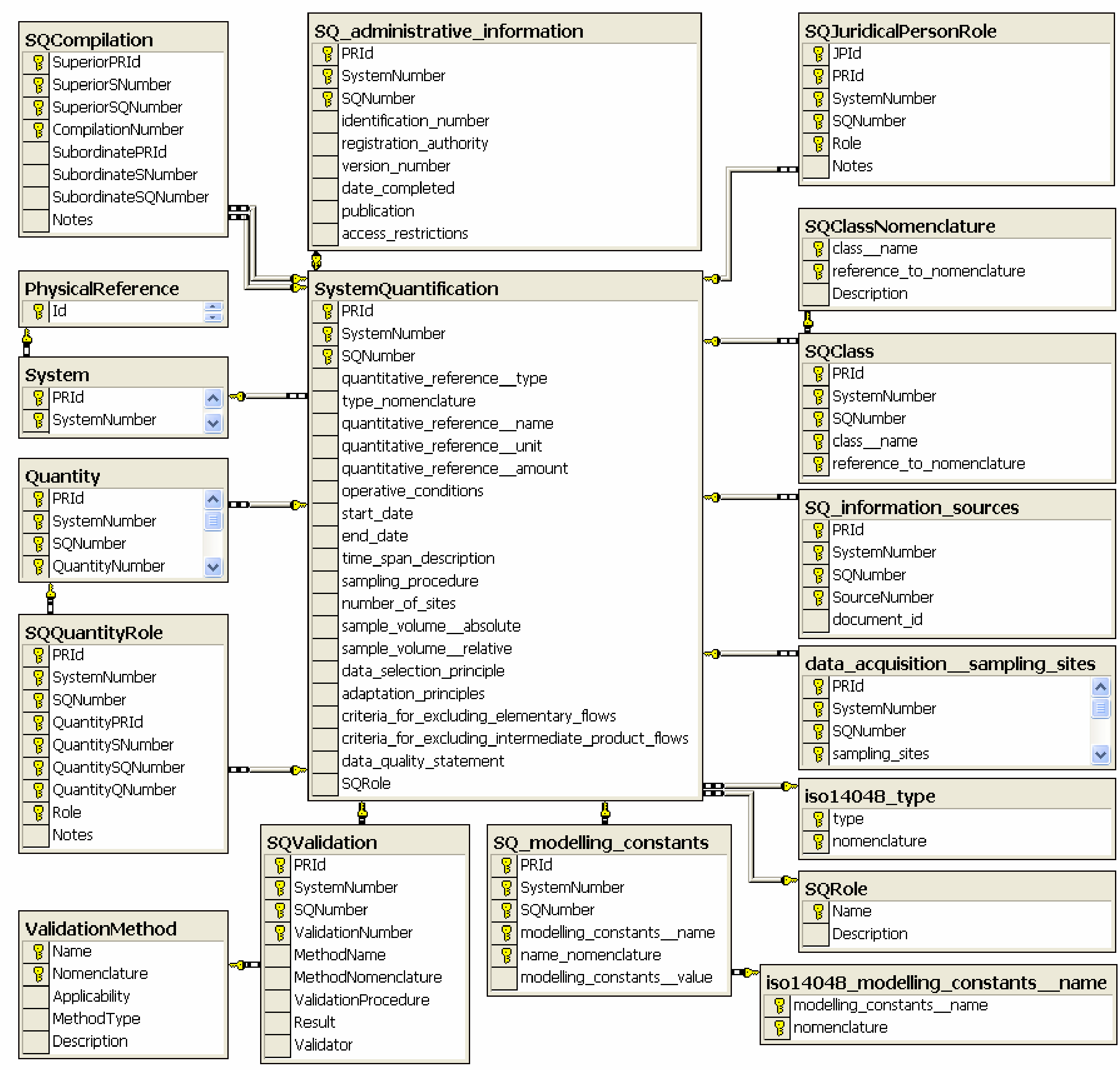

Figure 12. The tables in the IMPRESS data format related to the system quantification concept.

In this table information about measurements, for monitoring and controlling a system, is stored. The data can be used both as a link to the quantitative aspects considered in a system model or as a reference to actual measurements. For example a system model may describe a model for calculating CO2 emissions from a production site. The SystemQuantification stores data on the quantitative refererence which in this case could typically be a time-period, and further the procedural requirement on data collection. The specific quantitative data considered in the system models are defined as quantities in the table Quantity, see Quantity chaper 5.5.17. For CO2 emission this would be the CO2 emissions and the input of fuel and material needed to calculate CO2 emissions. The system model data is re-used when documenting the actual quantitative measured data for a given time-period. See also QAmountParameter chapter 5.13.4 for storage of quantitative process data.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	SystemNumber	Identification number to uniquely identify the system.
	SQNumber	Artificial identification number to uniquely identify the quantification of the system.
	Quantitative_reference__type	The type of quantitative reference. Examples: “Production period”, “Reference flow of process”, “Outgoing product flow”
	Type_nomenclature	The nomenclature of the quantitative reference.
	Quantitative_reference__name	The name of the quantitative reference.
	Quantitative_reference__unit	The unit of the quantitative reference.
	Quantitative_reference__amount	The amount of the quantitative reference.
	Operative_conditions	Explanation of the operating conditions for the process, i.e. actual (possibly non-linear) relations between inputs and outputs.

	Start_date	The start date of the valid time span. Unless projections or other forecasts have been applied, the valid time span is identical to the time of the data collection.
	End_date	The end date of the valid time span.
	Time_span_description	A free time-span description, i.e. as a description of the valid time span for the quantification of the system.
	Sampling_procedure	Description of the way the included processes have been selected from the population for which the data are valid, including notes of any bias in the procedure.
	Number_of_sites	The number of included sampled sites.
	Sample_volume__absolute	The total production volume of the sampled sites.
	Sample_volume__relative	The percentage of the total volume of the population for which the data is valid.
	Data_selection_principle	Description of the principle by which sites have been included in the average should be documented.
	Adaptation_principles	Description of the extrapolations and adjustments that may have been applied to remodel the acquired data into a unit process suited for LCI.
	Criteria_for_excluding_ elementary_flows	Description of the criteria used for selecting which elementary flows to include and, if deliberate and conscious, which to exclude. Generally not all elementary flows of an actual technical system are included when modelling

		it as a process. The criteria used for selecting which elementary flows to include and which to exclude are important information in order for a data user to assess the quality and the relevance of the process for a specific study.
	Criteria_for_excluding_ intermediate_product_flows	Description of the criteria used for exclusion of intermediate product flows, i.e. inputs and outputs that are not elementary flows. Such information is useful, for example, when assessing data gaps in the process. For example, some minor inputs of raw materials for the process may have been neglected in the data acquisition, due to lack of raw data (compare with the data field Criteria for excluding elementary flows).
	Data_quality_statement	Description of known general and specific quality strengths and weaknesses in the process.
	SQRole	The role of the system quantification data.
Foreign keys:	PRId, SystemNumber	Referencing the table System.
	Quantitative_reference__ type, Type_nomenclature	Referencing the table iso14048_type.
	SQRole	Referencing the table SQRole

5.5.15 Data_acquisition__sampling_sites

This table contains addresses to sites that is sampled in the data acquisition procedure.

Fields:	**PRId**	Identification number to uniquely identify the physical reference of which the system is a model.
	SystemNumber	Identification number to uniquely identify the system.
	SQNumber	Identification number to uniquely identify the

		quantification of the system.
	Sampling_sites	Address to site that has been sampled
Foreign keys:	PRId, SystemNumber, SQNumber	Referencing the table SystemQuantification.

5.5.16 SQ_modelling_constants

This table contains assumptions that have been held constant throughout the modelling of the process.

Fields:	**PRId**	Identification number to uniquely identify the physical reference of which the system is a model.
	SystemNumber	Identification number to uniquely identify the system.
	SQNumber	Identification number to uniquely identify the quantification of the system.
	Modelling_constants__name	The name of the modelling constant.
	Name_nomenclature	The nomenclature of the modelling constant
	Modelling_constants__value	The value for the constant that has been used in the modelling.
Foreign keys:	PRId, SystemNumber, SQNumber	Referencing the table SystemQuantification.
	Modelling_constants__name, Name_nomenclature	Referencing the table iso14048_modelling_constants__name.

5.5.17 Quantity

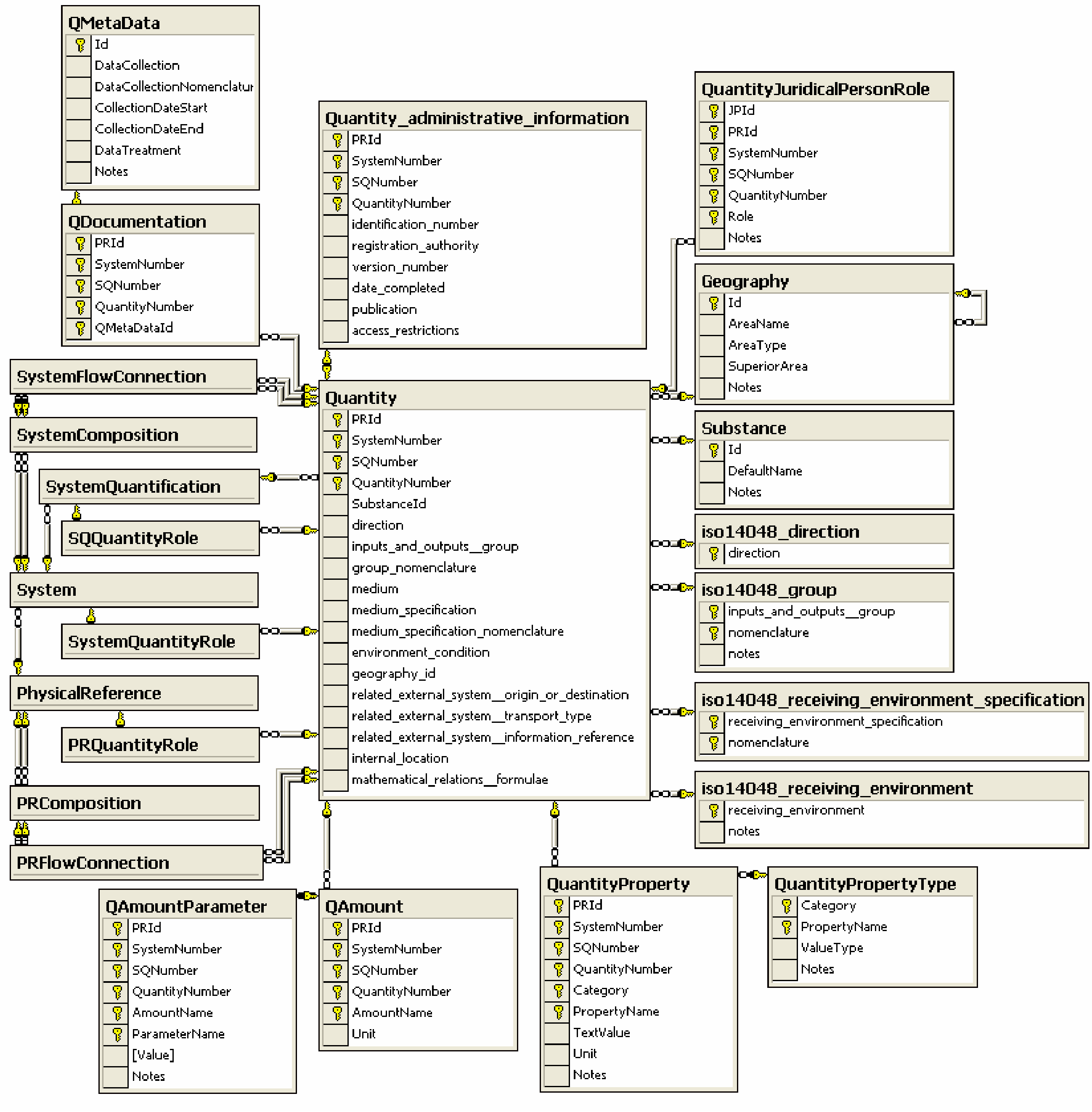

Figure 13. The tables in the IMPRESS data format related to the quantity concept.

In the Quantity table information is stored about inputs flowing to, and outputs flowing from a system. The quantitative value of the quantity is stored in QAmountParameter according to the general amount data structure, see Amount chapter 5.13.1. Documentation (meta data) about the quantity data is stored using the general QMetaData structure, see QMetaData chapter 5.15.

Fields:	**PRId**	Identification number to uniquely identify the physical reference.
	SystemNumber	Identification number to uniquely identify the system.
	SQNumber	Identification number to uniquely identify the quantification of the system.
	QuantityNumber	Identification number to uniquely identify an input or output from the system.
	SubstanceId	Identifier of the substance of the quantity. E.g. to the Id of CO2 if the quantity is an emission flow of CO2.
	Direction	The direction of the input or output, i.e. input to or output from a process.
	Inputs_and_outputs__group	The group to which the input or output belongs. The specification of group facilitates identification of the role of different inputs and outputs in the process.
	Group_nomenclature	The nomenclature of the group.
	Medium	Indication of how outputs and inputs are delivered from or to a process. Describes the type of environment that a resource is extracted from, or an emission is let out through; e.g. air, water, ground.
	Medium_specification	Indication of the type of environment that an input or output impacts. In LCIA data in this field should be matched with data on the cause-effect chain start of

		characterisation models.
	Medium_specification_nomenclature	The nomenclature of the medium.
	Environment_condition	Description of the environmental conditions indicated in medium and medium_specification.
	Geography_id	Identifier of the geographical location where flow (input or output) occur.
	Related_external_system__origin_or_destination	Geographical identification of delivering or receiving processes (upstream or downstream processes) for intermediate product flows.
	Related_external_system__transport_type	The name of the transport supplier or the type of transport.
	Related_external_system__information_reference	References to contact persons or other documents where information on the described related external systems may be found.
	Internal_location	Information about the use of an input or output within a process, e.g. the use of steam for a specific application within the process.
	Mathematical_relations__formulae	Specification of formula that describes the quantitative relation between an input and an output. A protocol for documenting formulas with references between inputs and outputs unambiguously and interpretable by a computer may be defined in relation to this field.
Foreign keys:	PRId, SystemNumber, SQNumber	Referencing the table SystemQuantification.

	SubstanceId	Referencing the table Substance.
	direction	Referencing the table iso4048_direction.
	inputs_and_outputs__group, group_nomenclature	Referencing the table iso4048_group
	Medium	Referencing the table iso4048_receiving_environment
	Medium_specification, Medium_specification_nomenclature	Referencing the table iso4048_receiving_environment_specification.
	Geography_id	Referencing the table Geography.

5.5.18 QuantityProperty

This table is used to store additional properties of the quantity.

Fields:	**PRId**	Identification number to uniquely identify the physical reference of which the system is a model.
	SystemNumber	Identification number to uniquely identify the system.
	SQNumber	Identification number to uniquely identify the quantification of the system.
	QuantityNumber	The name of the modelling constant.
	Category	The category of the property.
	PropertyName	The name of the property. Example: “Yearly production”.
	TextValue	The value for the property if value type is not “Numeric”
	Unit	The unit for the property.
	Notes	Additional relevant information
Foreign keys:	PRId, SystemNumber, SQNumber, QuantityNumber	Referencing the table Quantity.

	Category, PropertyName	Referencing the table QuantityPropertyType.

5.5.19 QuantityPropertyType

Table to store definitions of properties of quantities.

Fields:	**Category**	The category of the property.
	PropertyName	The name of the property. Example: “Yearly production”.
	ValueType	The value type for the property.
	Notes	Additional relevant information.
Foreign keys:	ValueType	Referencing the table ValueType.

5.6 Relating design and process data

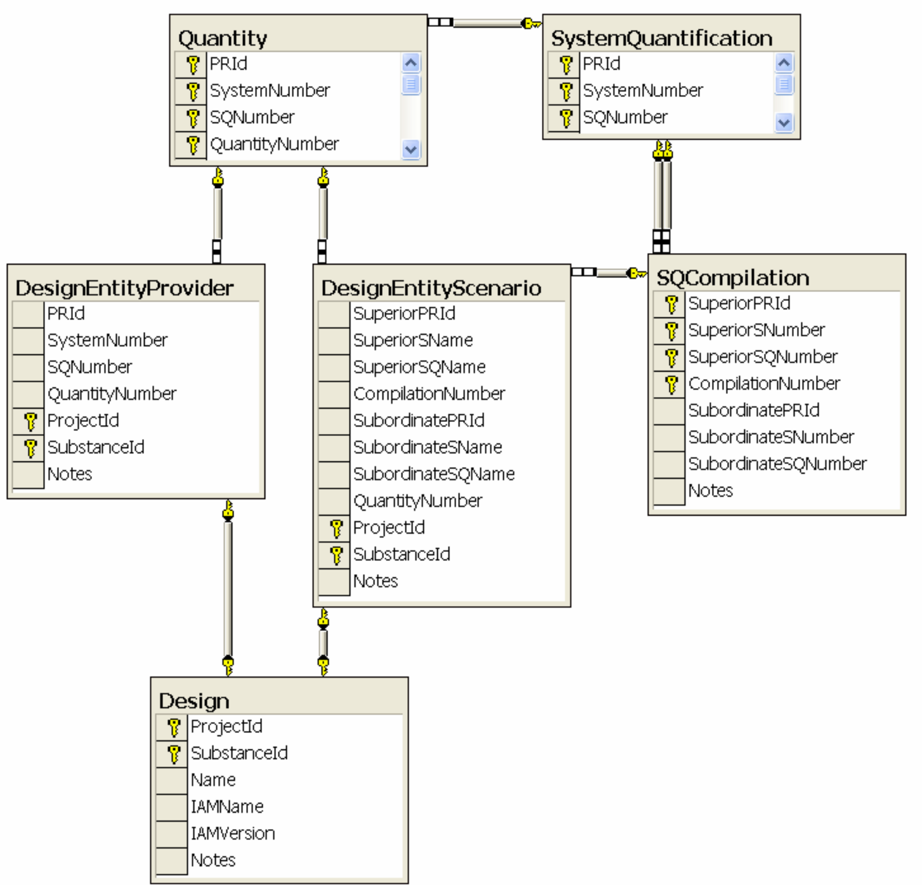

Figure 14 The tables in the IMPRESS data format for connecting design data with process data.

5.6.1 DesignEntityProvider

This table is used to relate a product that is being designed with the process that is producing the product. The product is related to a quantity output from a process. (DesignEntityProvider assumes the design to be delivered from the process).

Fields:	PRId	Identification number to uniquely identify the physical reference of which the system is a model.
	SystemNumber	Identification number to uniquely identify the system.

	SQNumber	Identification number to uniquely identify the quantification of the system.
	QuantityNumber	Identification number to uniquely identify the quantity.
	ProjectId	Identifier of the project of the design.
	SubstanceId	Identifier of the physical product or component.
	Notes	Additional relevant information.
Foreign keys:	PRId, SystemNumber, SQNumber, QuantityNumber	Referencing the table Quantity.
	ProjectId, SubstanceId	Referencing the table Design.

5.6.2 DesignEntityScenario

This table is used to relate a product or component that is being designed to the production of the same product or component, i.e. as a quantity output from a process.
To enable life cycle assessment of design, a design may be associated with a life cycle process/activity scenario. (DesignEntityScenario assumes the design to be exchanged between activities within the scenario.)

Fields:	**SuperiorPRId**	Identification number to uniquely identify the physical reference of which the system is a model.
	SuperiorSystemNumber	Identification number to uniquely identify the system.
	SuperiorSQNumber	Identification number to uniquely identify the quantification of the system.
	CompilationNumber	The name of the modelling constant.
	SubordinatePRId	Identification number to uniquely identify the physical reference of which the system is a model.
	SubordinateSystemNumber	Identification number to uniquely identify the system.
	SubordinateSQNumber	Identification number to uniquely identify the quantification of the system.

	QuantityNumber	Identification number to uniquely identify the quantity.
	ProjectId	Identifier of the project of the design.
	SubstanceId	Identifier of the physical product or component.
	Notes	Additional relevant information.
Foreign keys:	SuperiorPRId, SuperiorSNumber, SuperiorSQNumber, CompilationNumber	Referencing the table SQComplation.
	SubordinatePRId, SubordinateSystemNumber, SubordinateSQNumber, QuantityNumber	Referencing the table Quantity.
	ProjectId, SubstanceId	Referencing the table Design.

5.7 Aspect - interface between technical and nature system

An aspect is a definition of what it is in the technical system under consideration that causes an impact in the nature system. This can be compared with the term ‘environmental aspect’ as defined in ISO 14040:

Environmental aspect - Element of an organization’s activities, products or services that can interact with the environment.

According to the definition above, any organisations activities product or services is potentially an environmental aspect, as they all can interact with the environment directly or indirectly. However, only those elements that are explicitly defined as aspects can be considered in an assessment of the environmental impact of the system under consideration. In this way, the selection of aspect scopes the technical system quantity data needed to assess the environmental impact of the system. The principle and perspective applied when selecting the aspects – the aspect selection principle - therefore becomes important in order to specify meaningful aspects. Some examples of aspects considered in different applications are listed here:

LCA

Emissions to water.

Emissions of CO2 to air.

Resource extractions of iron ore from ground.

Emissions of dioxin in waste water from Gothenburg municipality to River Göta.

DfE

The amount of PVC in a component.

The amount of recycled material in a component.

The ratio of known materials in a component

The area exposed to water of coating paint

The energy efficiency of a component

EMS
Environmentally educated personnel
Emissions of CO_2 to air.

From the environmental impact modelling perspective an aspect specifies what it is in technical systems that triggers environmental mechanisms leading to environmental impacts. Many methods for calculating environmental impact are not developed for a specific technical system but intended to be generally applicable to many technical systems. For example, when performing classification according to ISO 14040 LCA methodology, aspects are used to match inputs and outputs in process inventory data with characterisation factors in impact assessment methods. If not handled transparently and rigorously there is a risk that mismatches occur, e.g. an elementary flow is not linked to a matching characterization factor or a flow is linked to a non-matching characterization factor. Aspects are necessary to consistently define interfaces between data from impact methods to data from technical systems.

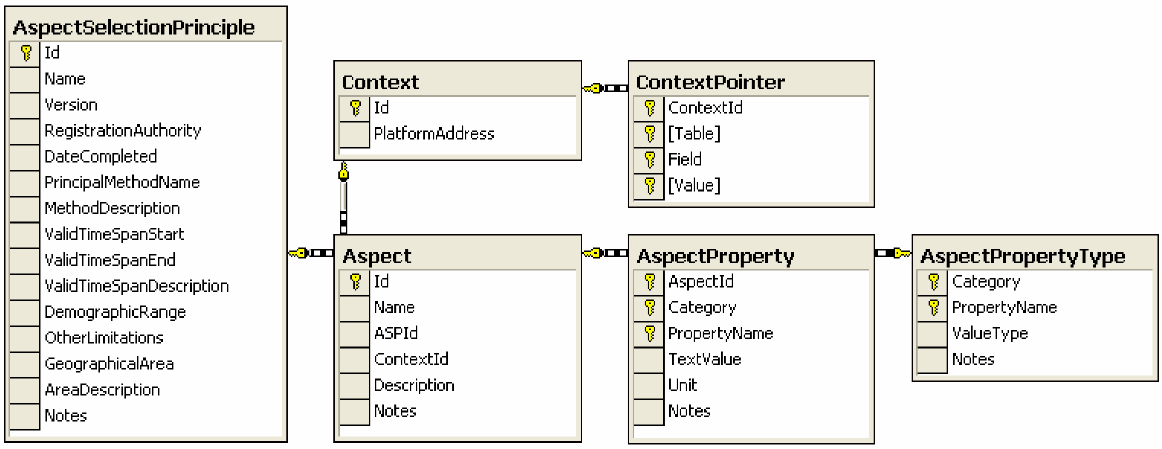

Figure 15 The tables in the IMPRESS data format related to the aspect concept

5.7.1 Aspect

The data format representing the aspect concept has a flexibility matching the variability of possible perspectives that can be applied to assess environmental impact of technical systems.

Each aspect has a unique identification number; a name may also be given each aspect for more understandable identification. To provide flexibility in the data format the context data structure is used by the aspect, which enables relations to any set of tables, columns, or records that represent the contextual framework in which the aspect resides, see Context chapter 5.13.5. For example, elementary flows in LCA are typical aspects in LCA studies. The context in this case refers to the tables representing inputs and outputs from processes in the technical system namely Quantity, QAmount, and QAmountParameter. Another example are the aspects based on the internal composition of a product in design for environment applications. In this case the contexts refer to the Composition and Conjunction tables representing a component structure.

The context of the aspect places the aspect in a contextual framework. The precise specification of an aspect within that context is represented by aspect properties.
The aspect property type contains the namespace in terms of property names for different aspect properties. The category concept is used categorise the different types of aspect properties, e.g. according to ISO/TS 14048 LCA data documentation format where an LCA flow is defined by a substance (or energy), direction, receiving environment, receiving environment specification, geographical location, and group (emission, natural resource, etc.)

Fields:	**Id**	The identifier of the aspect.
	Name	A name can be given to the aspect for understandable identification, e.g. “Emission of CO2 to air in Sweden”
	ASPName	Name of the selection principle that has been used when selecting the aspect.
	ASPVersion	Version of the selection principle that has been used when selecting the aspect.
	ContextId	Reference to a context that points out table area, e.g. a specific substance
	Notes	Any additional relevant information.
Foreign keys:	ASPName, ASPVersion	Referencing the table AspectSelectionPrinciple.
	ContextId	Referencing the table Context.

5.7.2 AspectSelectionPrinciple

The principle applied when selecting aspect is documented in this table.

Fields:	**Name**	Name of the principle that has been used for selection of the aspect.
	Version	Version of the principle.
	RegistrationAuthority	The name of the organisation responsible for maintaining the data and specifically the naming and versioning of the characterisation method
	DateCompleted	The date when the principle was finalised or published.
	PrincipalMethodName	Aspect selection principles often are variants of principal methods. For clarification the principal method can be named.
	MethodDescription	A description of the method that has been used when selecting the aspect.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the aspect selection principle may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Start of the time span for which the aspect selection principle is valid
	ValidTimeSpanEnd	End of the time span for which the aspect selection principle is valid
	ValidTimeSpanDescription	Clarifying description of the valid time span
	DemographicRange	Description of the demographic range of people that has been involved in the selection of the aspects
	GeographicalArea	The geographical area that the method is applicable for
	AreaDescription	Clarifying description of the geographical area
	OtherLimitations	Description of other boundaries of the aspect

		selection principle
	Notes	Any additional relevant information.
Foreign keys:	GeographicalArea	Referencing the table Geography.

5.7.3 AspectPropertyType

Each aspect can be described with an arbitrary number of properties. The defining of a namespace for aspect properties is done in close relation to how aspects are applied within a specific method and tool.

Fields:	**Category**	A category to which the property relates.
	PropertyName	Name of a property.
	ValueType	Defines the set of different possible values a property may take (see ValueTypeDefinition).
	Notes	Any additional relevant information.
Foreign keys:	N/A	

5.7.4 AspectProperty

In this table each aspect can be described with an arbitrary number of properties.

Fields:	**AspectId**	Reference to the aspect for which the property is valid.
	Category	A category to which the property relates.
	PropertyName	Name of a property.
	TextValue	Holds the data for the aspect property if it is not a numeric the amount structure is not used for this property, (see description above).
	Unit	The unit for the property if applicable
	Notes	Any additional relevant information.
Foreign keys:	AspectId	Referencing the table Aspect.
	Category,	Referencing the table AspectPropertyType.

	PropertyName	

Below is an example of aspect properties with values populated based on the ISO/TS 14048 FlowSpecification category.

PropertyName	**TextValue**
Unit	kg
Substance	CO2
Direction	Output
GeographicalLocation	Europe
Group	Emission
ReceivingEnvironment	Air
ReceivingEnvironmentSpecification	Air

Another example based on IMI component structure specification applicable to EPI definitions in DfE methodology is given in the below table.

PropertyName	**TextValue**
Unit	kg
Substance	PVC
RoleCategory	Container
ExposureArea	m3
ExposureMedium	Water
GeographicalLocation	Europe

5.8 Environmental indicators

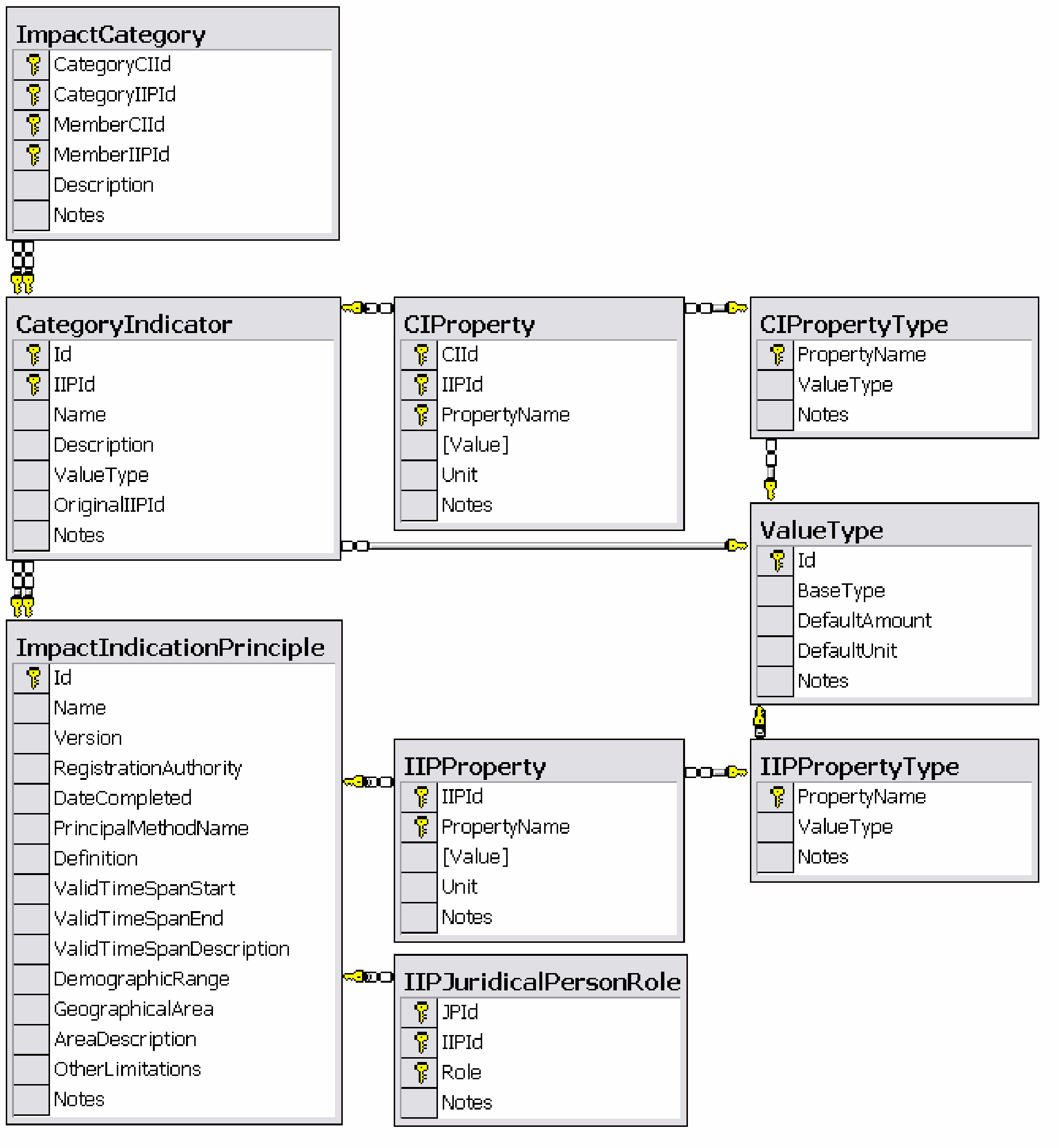

Figure 16 The tables in the IMPRESS data format related to category indicators.

5.8.1 CategoryIndicator

A category indicator is a defined quantity in the nature system that represents an environmental effect of concern.

Fields:	**Id**	Unique identifier of the category indicator.
	IIPId	Reference to the Impact Indication Principle applied when defining the category indicator.
	Name	Name of the category indicator.
	Description	A description of the environmental category indicator as well as a motivation of why the indicator is selected.
	ValueType	Defines the set of different possible values a property may take.
	OriginalIIPId	
	Notes	Additional relevant information.
Foreign keys:	IIPId	Referencing the table ImpactIndicationPrinciple.
	ValueType	Referencing the table ValueType.
	OriginalIIPId	Referencing the table ImpactIndicationPrinciple. A reference to the Impact Indication Principle that originally was the starting point when creating this indicator. If any changes are made to an Impact Indication Principle a new version of the principle is created. The name might also be replaced.

5.8.2 CIPropertyType

It may be required to define properties of indicators in order to distinguish between them in a structured way. For example two indicators concerning acidification potential may refer to different geographical areas.

Fields:	**PropertyName**	Name of a category indicator property. Examples: "Geographical area", “Minimum body weight”, “Age kohort”
	ValueType	Reference to the value type of the property.
	Notes	Any additional relevant information.
Foreign	ValueType	Referencing the table ValueType.

keys:		

5.8.3 CIProperty

This table stores values of property for a specific category indicator.

Fields:	**CIId**	
	IIPId	
	PropertyName	Name of a property.
	Value	The value of the property.
	Unit	The unit for the property.
	Notes	Any additional relevant information.
Foreign keys:	CIId, IIPId	Referencing the table CategoryIndicator
	PropertyName	Referencing the table CIPropertyType.
	Unit	Referencing the table Unit.

5.8.4 ImpactIndicationPrinciple

An impact indication principle (IIP) describes the reason why a set of indicators are chosen.

Fields:	**Id**	Unique identifier of the IIP
	Name	A name of the overall principle or policy applied when selecting a set of category indicators, e.g. the environmental policy of a company
	Version	If the principle or policy is updated the updated versions are given successive version numbers.
	RegistrationAuthority	Name of organisation endorsing the documentation of the IIP.
	DateCompleted	Date when the principle or policy was formulated.
	PrincipalMethodName	Impact indication principles often are variants of principal methods, e.g. willingness to pay, distance to target, etc. For clarification the principal method

		can be named.
	Definition	A definition or statement of the principle or policy. A motivation of why the chosen indicators have been selected shall be included.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the IIP may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Date of the start valid time span of the IIP
	ValidTimeSpanEnd	Date of the end valid time span of the IIP
	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the IIP. (E.g. only adults are considered)
	GeographicalArea	A name of the geographical area considered. Reference to geography nomenclature.
	AreaDescription	Description of the valid geography of the IIP, e.g. only European conditions considered.
	OtherLimitations	Description of other limitations of the IIP
	Notes	Additional relevant information.
Foreign keys:	GeographicalArea	Referencing the table Geography.

5.8.5 IIPPropertyType

Additional properties of IIPs can be defined in this table.

Fields:	**PropertyName**	Name of a category indicator property. Examples: "Geographical area”, “Minimum body weight”, “Age kohort”
	ValueType	Reference to the value type of the property.
	Notes	Any additional relevant information.

Foreign keys:	ValueType	Referencing the table ValueType.

5.8.6 IIPProperty

This table is used to store additional properties for IIPs.

Fields:	**IIPId**	Refence to the IIP.
	PropertyName	The name of the property.
	Value	The value of the property.
	Unit	The unit for the property.
	Notes	Any additional relevant information.
Foreign keys:	IIPId	Referencing the table ImpactIndicationPrinciple.
	PropertyName	Referencing the table IIPPropertyType.
	Unit	Referencing the table Unit.

5.8.7 ImpactCategory

This table can be used to group indicators in categories. One way of using this table is to move the indicators closer to the environment, i.e. the category “Global warming” can be used to group all the indicators that affect global warming.

Fields:	**CategoryCIId**	The name of the environmental indicator category that shall be used to group indicators, i.e. the category “Global warming” can be used to group all the indicators that influence the global warming.
	CategoryIIPId	The Impact Indication Principle that has been a basis when creating the impact category.
	MemberCIId	The name of an environmental indicator that is a member of the impact category
	MemberIIPId	The Impact Indication Principle that has been a basis when creating the member indicator.

	Description	A motivation of why the indicator member is included in the indicator category.
	Notes	Any additional relevant information.
Foreign keys:	CategoryCIId, CategoryIIPId	Referencing the table CategoryIndicator.
	MemberCIId, MemberIIPId	Referencing the table CategoryIndicator.

5.9 Characterisation - quantifying environmental effects

The data format for storing models for quantifying effects on the environment due to activities in the technical system is based on the concepts regarding characterisation in LCA methodology. Essentially the structure supports the storage of quantitative relations between aspect and category indicators in terms of characterisation parameters.

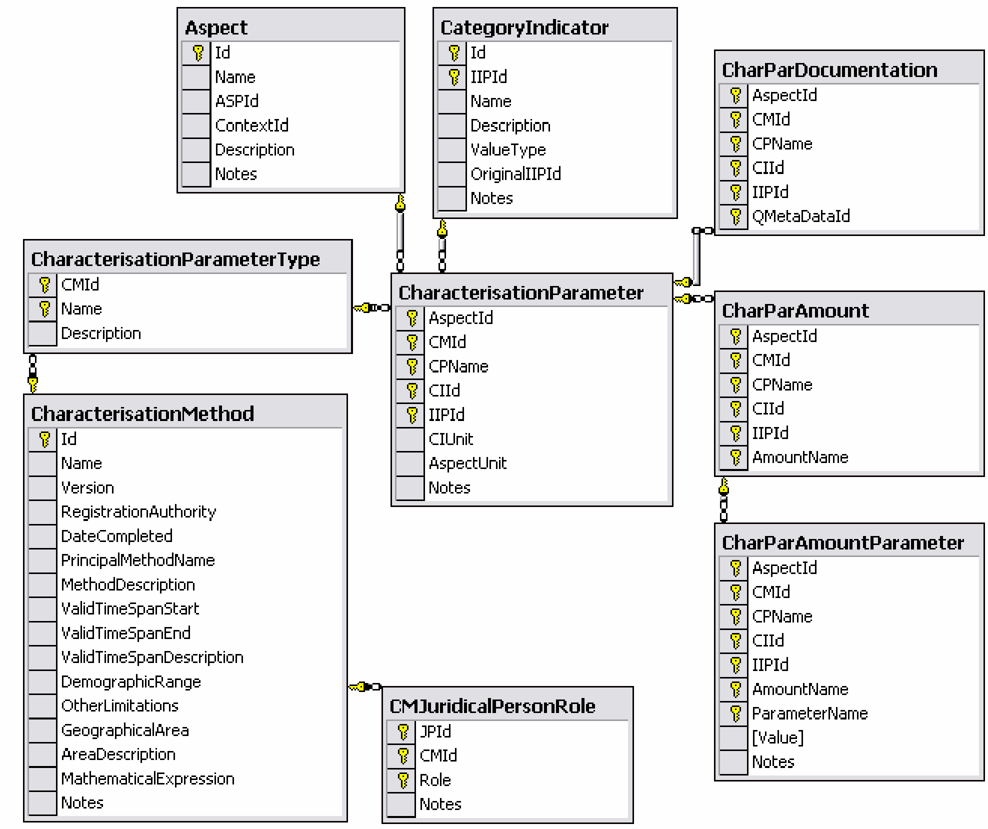

Figure 17 The tables in the IMPRESS data format for quantification of environmental effects

The category indicator is a definition of the effect of concern, e.g. years of lost lives (YOLL), see Environmental Indicators chapter 5.8. The aspect points out what is considered to be the cause of the environmental effect, e.g. emissions of cadmium into water in Europe. The aspect serves as an interface to match the characterisation data with data about the technical system, see also Aspect chapter 5.7.

5.9.1 CharacterisationMethod

The method to acquire the characterisation parameters are referred to as a characterisation method.

Fields:	**Id**	Unique identifier of the characterisation method
	Name	Name of the characterisation method.
	Version	If the characterisation method is updated the updated versions are given successive version numbers.
	RegistrationAuthority	Name of organisation endorsing the documentation of the characterisation method.
	DateCompleted	The date when the method was finalised or published.
	PrincipalMethodName	Characterisation methods often are variants of principal methods. For clarification the principal method can be named.
	MethodDescription	A description of the method that has been used when calculation the indicator.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Date of the start valid time span of the method
	ValidTimeSpanEnd	Date of the end valid time span of the method
	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the method.
	GeographicalArea	A reference to a geographical nomenclature, addressing the geographical area for the range of the method described.
	AreaDescription	A description of the applicable geographical area.

	OtherLimitations	Any other limits in terms of system boundaries or applicability of the method.
	MathematicalExpression	Syntactic description of the mathematical rule to apply to the characterisation parameter types and parameters, if more than one parameter type is defined.
	Notes	Any additional relevant information.
Foreign keys:	GeographicalArea	Referencing the table Geography.

5.9.2 CharacterisationParameterType

The characterisation method includes the mathematical expression that represents the relation between aspect and category indicator. This table define the parameters in the expression required. In a linear model the expression only requires one parameter in terms of proportionality constants. This type of model with single characterisation factors is most commonly used within LCIA practice. The expression is documented in the table CharacterisationMethod in the field MathematicalExpression.

Fields:	**CMId**	Reference to the characterisation method.
	Name	The name of the characterisation parameter type.
	Description	Description of how the parameter should be interpreted and used.
Foreign keys:	CMId	Referencing the table CharacterisationMethod.

5.9.3 CharacterisationParameter

This table holds a characterisation parameter in the relation of a specific aspect and environmental category indicator pair in a characterisation method. In case of a linear model in LCA, this is equivalent to characterisation factors. The numerical value of the characterisation parameter is stored using the table CPAmountParameters, see Amount chapter 5.13.1.

Fields:	**AspectId**	Reference to the aspect.
	CMId	Reference to the characterisation method.
	CPName	Reference to the characterisation parameter type.
	CIId	Reference to the category indicator.
	IIPId	Reference to the impact indication principle.
	Notes	Any additional relevant information.
Foreign keys:	AspectId	Referencing the table Aspect.
	CMName, CPName	Referencing the table CharacterisationParameterType.
	CIId, IIPId	Referencing the table CategoryIndicator

5.10 Environmental impact modelling

This section concerns the underlying models that generate the quantitative parameters between aspect and indicator, e.g. a characterisation parameter. The model is based on results from the OMNIITOX project[12]. The concept model resembles the cause effect model for characterisation for quantifying environmental effects on technical systems. However, the focus in this data structure for environmental impact models lies on the environmental mechanism. This model is intended for storing data of any environmental cause effect models. The models may be used to calculate e.g. generic characterisation parameters, site specific environmental impacts, environmental risk assessment, etc.

The core model is presented in figure 18. Here, the load represents the cause that triggers a mechanism that results in an impact on an indicator. The mechanism contains both a quantitative as well as a qualitative description of the relation between load and indicator. The quantitative relation is based on analytical data in terms of substance and nature properties. These properties are defined as data input required to execute the calculations in the model. Each mathematical operation on the input data is stored in a formalised syntax in order to enable it to be interpreted into operations by a computer.

[12] OMNIITOX CM and data format

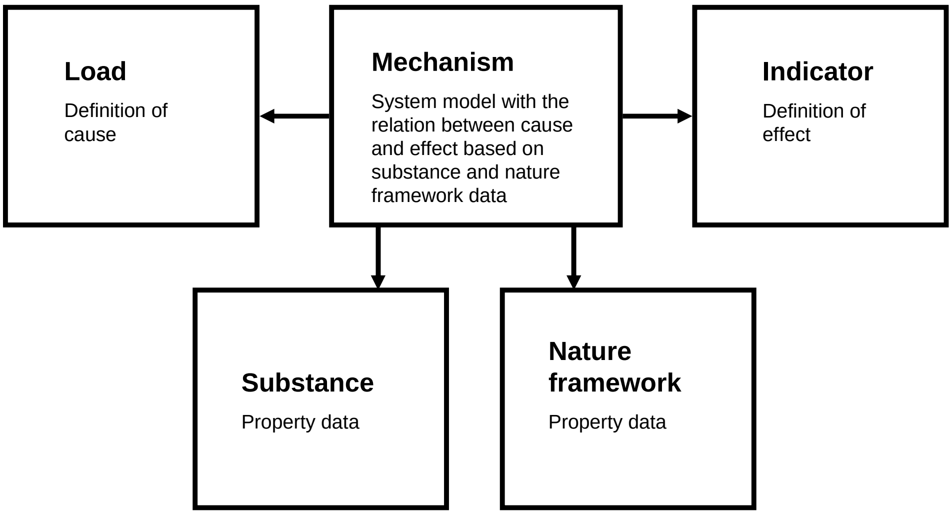

Figure 18 The core concepts of the concept model for environmental impact modelling.

A higher resolution of the OMNIITOX concept model is presented in figure 19. By considering an indicator of a mechanism as being the load to another (indicator load connection), mechanisms can be connected in cause effect chains. This connection of mechanism implies an expansion of the system boundaries into a larger mechanism. Further, the same mechanism may concern multiple loads and multiple indicators. The cause effect connection defines which loads affect which indicators. Finally the model structures a mechanism parameter algorithm, i.e. a definition of required input data in terms of substance and nature properties how these data are aggregated into mechanism parameters.

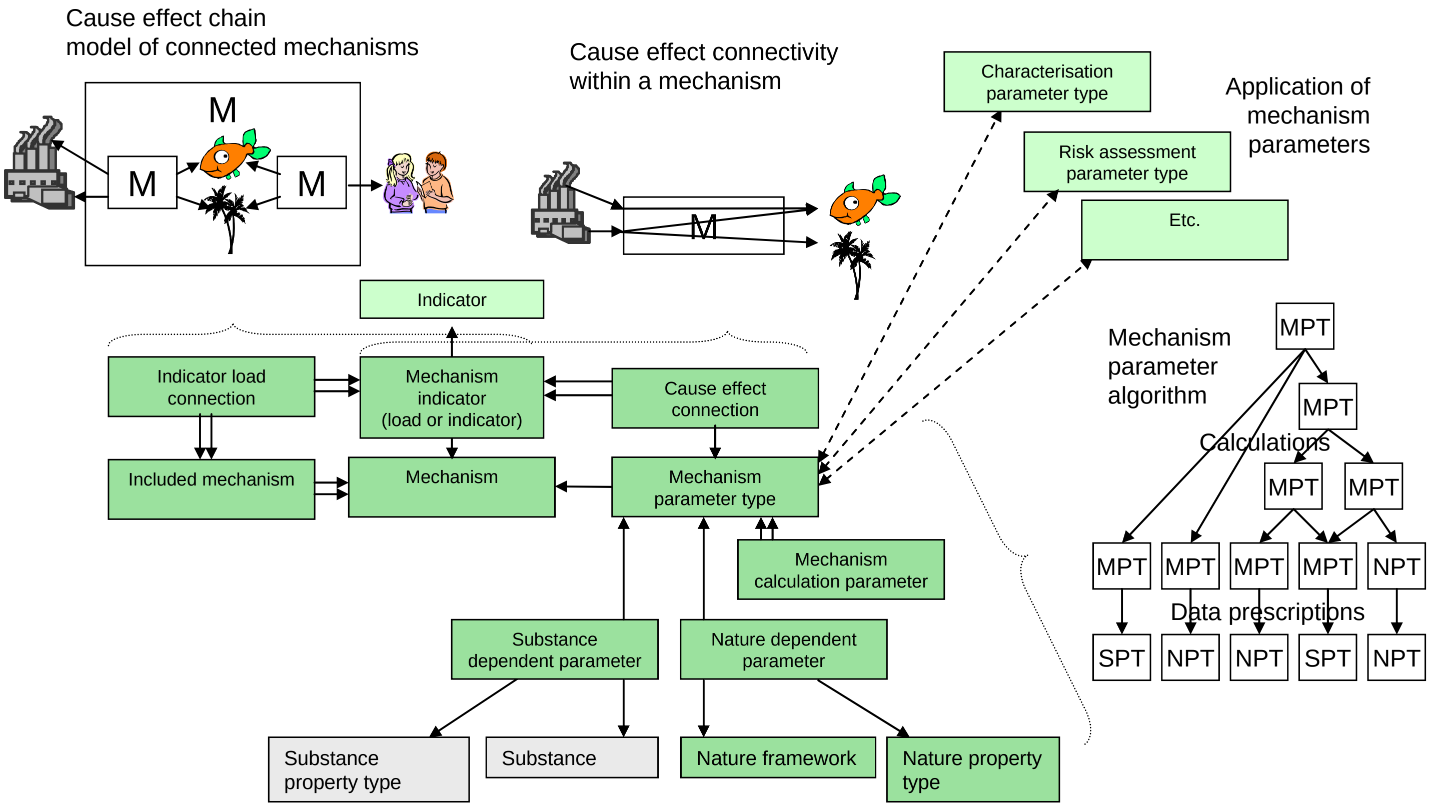

Figure 19 Environmental impact modeling concepts based on the OMNIITOX concept model.

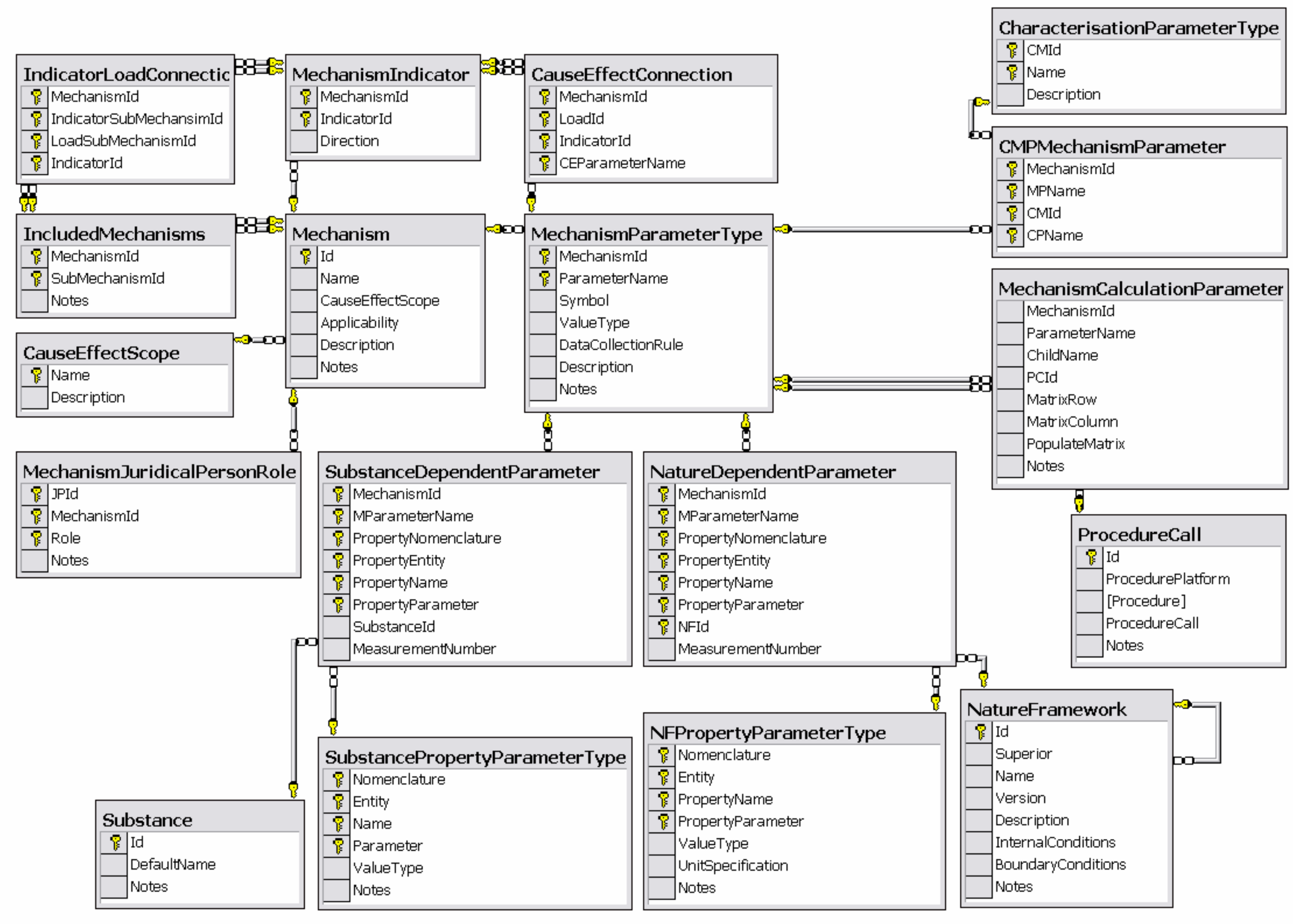

Figure 20 The tables in the IMPRESS data format related to environmental impact modeling. Compare with figure 19 above.

5.10.1 Mechanism

A mechanism describes the connection between indicators in a cause effect chain. An indicator triggering the mechanism is called a load. In general this can be formalised as $I=f(\lambda)$ where I is the indicator value and λ is the load value. The function f is equivalent to the quantitative representation of the mechanism concept.

Fields:	**Id**	Unique identifier of the mechanism.
	Name	Name of the mechanism.
	CauseEffectScope	A brief description of the modelled cause effect chain scope of the mechanism, examples: “Fate”, “Exposure”, “Fate and exposure”
	Applicability	A description of the applicability of the model. I.e. what substances are applicable to the model; how the Mechanism model shall and shall not be used, etc.
	Description	A complete description of the mechanism model explaining how substance and nature property data is aggregated into Cause effect Parameters. As this can be a very extensive description depending on the complexity of the mechanism model this can be a reference to a document such as a MS Word document that is available to the information system so that it can be presented to a user and edited by a data documenter.
	Notes	Any additional relevant information.
Foreign keys:	CauseEffectScope	Referencing the table CauseEffectScope.

5.10.2 MechanismIndicator

The MechanismIndicator attaches indicators and loads to a mechanism.

Fields:	**MechanismId**	The mechanism
	IndicatorId	The affected indicator.
	Direction	Specification of the direction of the indicator in the cause effect chain. Either “Load” or “Indicator”.
Foreign keys:	MechanismId	Referecing the table Mechanism
	IndicatorId	Referencing the table Indicator

5.10.3 IncludedMechanisms

Mechanisms can be grouped into hierarchical levels. This means that a mechanism can be performed in several smaller steps which all can be reused in another mechanism.

Fields:	**MechanismId**	Refering to the superior mechanism
	SubMechanismId	Refering to the subordinate mechanism
	Notes	Any additional relevant information.
Foreign keys:	SystemId	Referencing the table Mechanism.
	SubsystemId	Referencing the table Mechanism.

5.10.4 IndicatorLoadConnection

Mechanisms included as sub-mechanisms in a superior system can be “linked” by connecting an indicator of one sub-mechanism to a matching load of another sub-mechanism.

Fields:	**MechanismId**	Referring to the superior mechanism
	IndicatorSubMechanismId	Referring to the subordinate mechanism connected to the indicator
	LoadSubMechanismId	Referring to the subordinate mechanism connected to the load
	IndicatorId	Referring to the indicator being connected between the sub-mechanisms
	Notes	Any additional relevant information.
Foreign keys:	MechanismId, IndicatorSubMechanismId	Referencing the table IncludedMechanisms.
	IndicatorSubMechanismId, IndicatorId	Referencing the table MechanismIndicator.
	MechanismId, LoadSubMechanismId	Referencing the table IncludedMechanisms.
	LoadSubMechanismId, IndicatorId	Referencing the table MechanismIndicator.

5.10.5 CauseEffectConnection

The cause effect connection specifies explicit relations between a load and an indicator “inside” a mechanism. This allows the reuse of the same mechanism information for more than one cause effect connection. Also it enables a structure for multivariate mechanism models where several loads can affect an indicator, e.g. synergism, antagonism.

Fields:	**MechanismId**	The mechanism
	LoadId	The causing load.
	IndicatorId	The affected indicator.
	CEParameterName	Cause-effect parameter; the most aggregated mechanism parameter of a mechanism applicable to the cause effect connection.
Foreign keys:	MechanismId	Referencing the table Mechanism
	LoadId	Referencing the table Indicator
	IndicatorId	Referencing the table Indicator
	MechanismId, CEParameterName	Referencing the table MechanismParameter.

5.10.6 MechanismParameterType

This table defines all parameters included in the equations in the mechanism model. This includes the substance and nature dependent properties, intermediate aggregates and cause-effect parameters.

Mechanism parameters can either be collected from the database, or it can be aggregated through calculation based on a set of other mechanism parameters.

Fields:	**MechanismId**	The mechanism.
	ParameterName	The name of the parameter, if the parameter is an substance property or a nature property already defined the name the original name should be used; examples: “Characterisation factor”, “No effect load threshold”, “Boiling point”, “Acidity dissociation constant”.

	Symbol	The symbol that represents the parameter in the mechanism model equations; subscript indexing are represented by the character “@”. All symbols must be unique within a mechanism. Examples: “DALY”, “a”, “X@1”, “C@water”.
	ValueType	Defines the valid value type of the parameter.
	DataCollectionRule	Defines what method is applied if the parameter should be collected from the database. Examples: “aritmetric mean”, “geometric mean”.
	Description	A description of what the parameter represents.
	Notes	Additional relevant information.
Unique constraint:	MechanismId, Symbol	A symbol used may only represent one parameter type per mechanism.
Foreign keys:	MechanismId	Referencing the table Mechanism.
	ValueType	Referencing the table ValueType.
	DataCollectionRule	Referencing the table DataCollectionRule.

5.10.7 SubstanceDependentParameter

This table prescribes the required input data in terms of substance properties and relates the data to mechanism parameter types.

Fields:	**MechanismId**	Referencing the the mechanism.
	MParameterName	Referencing the name of the substance dependent mechanism parameter.
	PropertyNomenclature	Referencing the nomenclature of the substance property.
	PropertyEntity	Referencing the entity of the substance property.
	PropertyName	Referencing the name of the substance property.
	PropertyParameter	Referencing the parameter of the substance property.
	SubstanceId	Referencing a specific substance. This is not necessary if all substance dependent parameters of a

		mechanism are referring to the same substance. If this is the case the substance referred to in the load is regarded, e.g. if the load is specified as an input of carbon dioxide into the mechanism system model.
	MeasurementNumber	Option to point out a specific measurement.
Foreign keys:	MechanismId, MParameterName	Referencing the table MechanismParameterType.
	PropertyNomenclature, PropertyEntity, PropertyName, PropertyParameter	Referencing the table SubstancePropertyParameterType.
	SubstanceId	Referencing the table Substance.

5.10.8 NatureDependentParameter

This table prescribes the required input data in terms of nature framework properties and relates the data to mechanism parameter types.

Fields:	**MechanismId**	The mechanism.
	MParameterName	Referencing the name of the nature dependent mechanism parameter.
	PropertyNomenclature	Referencing the nomenclature of the substance property.
	PropertyEntity	Referencing the entity of the substance property.
	PropertyName	Referencing the name of the substance property.
	PropertyParameter	Referencing the parameter of the substance property.
	NFId	Unique identifier of the nature framework.
	MeasurementNumber	Option to point out a specific measurement.
Foreign keys:	MechanismId, MParameterName	Referencing the table MechanismParameterType.
	PropertyNomenclature, PropertyEntity, NDParameterName,	Referencing the table NFPropertyType.

	PropertyParameter	
	NFId	Referencing the table NatureFramework.

5.10.9 MechanismCalculationParameter

The table MechanismCalculationParameter specifies what procedure call (e.g. mathematical formula) is used to calculate a mechanism parameter type and what mechanism parameter type(s) are used as input data to the procedure call. In addition, for matrix operations it specifies what parameters are elements in a matrix, or the location of the element in a matrix if the parameter being calculated is an element within that matrix. The input parameter can be seen as branches (children) to the parameter being calculated and the complete collection of calculations in a mechanism can be regarded as a “calculation tree”.

Fields:	MechanismId	Referring to the mechanism the procedure call is used within.
	ParameterName	The name of the resulting mechanism parameter.
	ChildName	The name of a mechanism parameter needed for the calculation, or being an element within a matrix.
	PCId	Referring to the procedure call, e.g. a formula in a specified syntax that can be interpreted by a software application.
	MatrixRow	The row position in the matrix (or vector) if the calculated mechanism parameter is an element in a matrix parameter type. If the calculated mechanism parameter is a scalar: NULL
	MatrixColumn	The column position in the matrix (or vector) if the calculated mechanism parameter is an element in a matrix parameter type. If the calculated mechanism parameter is a scalar: NULL
	PopulateMatrix	“YES” if the parameter is a matrix populated by its children, NULL if the parameter is a result of a calculation (procedure call).
Foreign	MechanismId,	Referencing the table MechanismParameterType.

keys:	ParameterName	
	MechanismId, ChildName	Referencing the table MechanismParameterType.
	PCId	Referencing the table ProcedureCall.

5.10.10 ProcedureCall

The procedures to transform input parameters into output parameters are defined in this table. The syntax may specific to the application.

Fields:	**Id**	Unique identifier of the procedure call.
	ProcedurePlatform	Name of platform where the procedure will be executed. Examples: “Matlab”, “OMNIITOX IS”
	Procedure	The procedure expression. Syntax is platform specific. Examples valid for the OMNIITOX IS: “A*B”, “c + d*sin(e)“.
	ProcedureCall	Option to explicitly store the functional procedure call as a string. Syntax is platform specific. Example: “CFcalcproc(A,B,C)”
	Notes	Any additional relevant information.
Foreign keys:	N/A	

5.10.11 CMPMechansismParameter

A mechanism parameter type can be used in various applications. In the case of LCA the CMPMechanismParameter table is used to connect mechanism parameter types to characterisation parameter types.

Fields:	**MechanismId**	Reference to the mechanism
	MPName	Reference to the mechanism parameter type
	CMId	Reference to the characterisation method
	CPName	Reference to the characterisation parameter type
Foreign keys:	MechanismId, MPName	Referencing the table MechanismParameterType

	CMId, CPName	Referencing the table CharacterisationParameterType

5.10.12 NatureFramework

The nature framework concept represents a system model of the nature. System boundaries and internal distribution of compartment media, geography, time frame, etc. can be specified. A hierarchy of nature frameworks can also be defined. Nature properties and property types such as precipitation, wind speed etc. are defined similarly as in the substance concept. The boundaries defined for nature frameworks are in general specific for the environmental impact model. However the nature framework is considered a stand alone concept to which enables the use of nature framework data in different impact models.

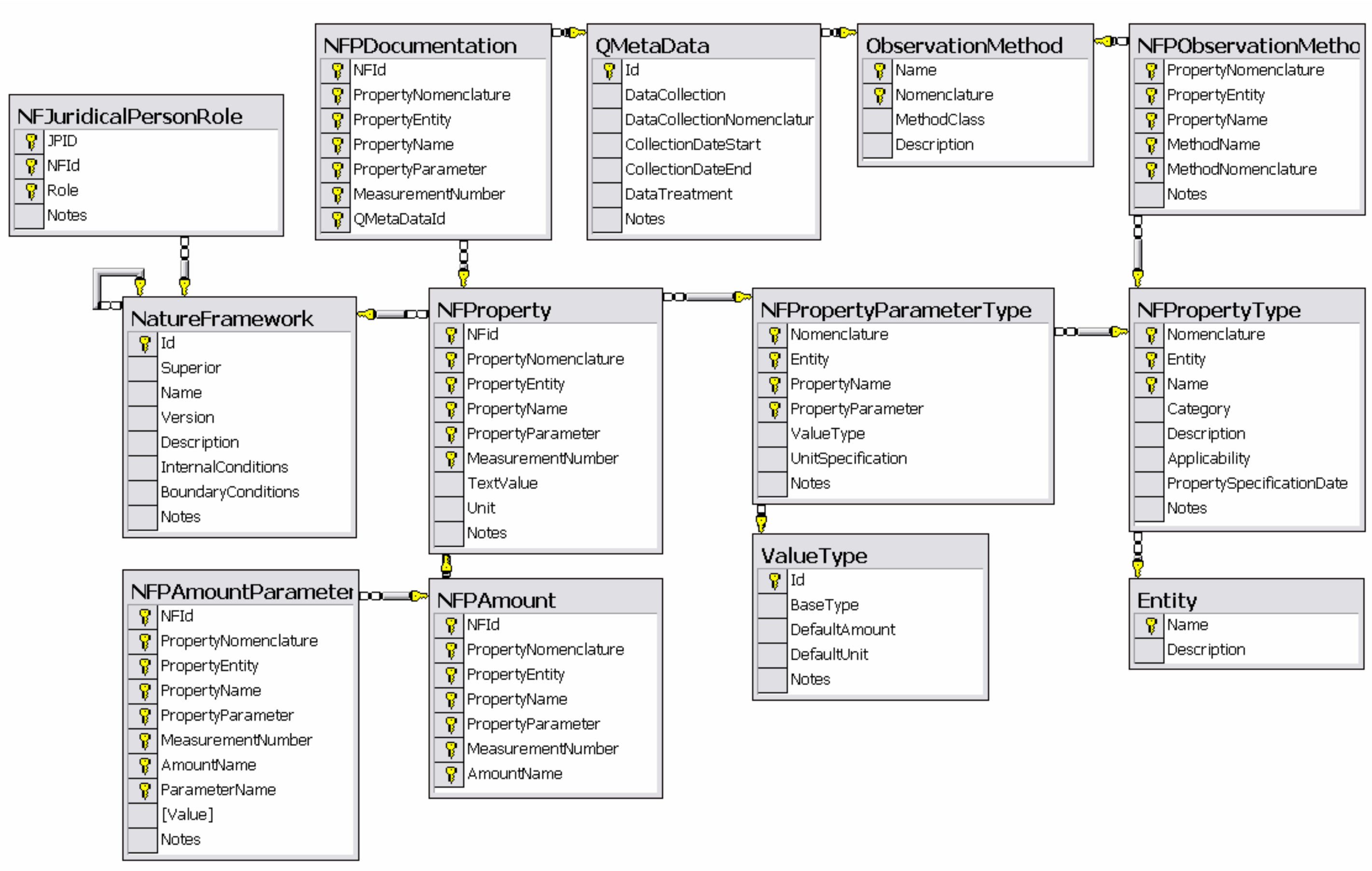

Figure 21 The tables in the IMPRESS data format related to the nature framework concept

The nature framework is in general a specification of chosen system boundaries and system conditions for a model of (a part of) the nature system.

Fields:	**Id**	Unique Id of the nature framework
	Superior	Identifier of a superior nature framework, if the nature framework is included in a larger framework
	Name	The name of the nature framework model. Examples: “Global average biosphere”, “Europe fresh waters”, “Chronic toxicity laboratory test water tank”
	Version	The version of the nature framework model.
	Description	A qualitative general description of the nature framework model
	InternalConditions	Description of conditions inside the system boundary, e.g. internal distribution of environmental compartments, internal geographical boundaries, etc.
	BoundaryConditions	Description of conditions at the system boundary, e.g. time boundary, compartment boundary
	Notes	Additional notes for the description of the nature framework.
Foreign keys:	Superior	Referencing the table NatureFramework

5.10.13 NFPropertyType

The NFPropertyType is short for “NatureFrameworkPropertyType”. It is used to define properties in the nature system (analogous to SubstancePropertyType for substances). In OMNIITOX this is called Nature Property Specification.

Fields:	**Nomenclature**	Name of nomenclature in which the property was defined. Examples: “REPID”, “OMNIITOX”, “RAVEL”.
	Entity	Physical or other type of entity that the property represents. Examples: “Wind speed”, “Precipitation of iron”, “Concentration of hydrogen ions”.

	Name	Name of a property. Examples: "Average wind speed", "Annual precipitation". In OMNIITOX the name is created by concatenating the entity name and the observation method(s) specified.
	Category	Name of the property category. Examples: "Geophysical", "meteorological"
	Description	A description of the nature framework property type.
	Applicability	Domain of application of the property type.
	PropertySpecificationDate	The last date the nature framework property type was changed**.**
	Notes	Additional relevant information.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.
	Entity	Referencing the table Entity.

5.10.14 NFPObservationMethod

This bridging table connects a nature framework property type to one of several observation methods. In this the allowed methods to quantify the property are specified, see ObservationMethod chapter 5.17.22.

5.10.15 NFPropertyParameterType

Short for "NatureFrameworkPropertyParameterType". Analogous to SubstancePropertyParameterType the NFPropertyParameterType is needed as a nature framework property may require more than one parameter.

Fields:	**Nomenclature**	Name of nomenclature in which the property was defined. Examples: "REPID", "OMNIITOX", "RAVEL".
	Entity	Physical or other type of entity that the property represents. Examples: "Wind speed", "Precipitation of iron", "Concentration of hydrogen ions".

	PropertyName	Name of a property. Examples: "Average wind speed", "Annual precipitation". In OMNIITOX the name is created by concatenating the entity name and the observation method(s) specified.
	PropertyParameter	Name of the parameter. Examples: "Wind direction", "Species"
	ValueType	Defining the valid range of values for the substance property.
	UnitSpecification	Explanation of the unit if its not obvious.
Foreign keys:	Nomenclature, Entity, PropertyName	Referencing the table NFPropertyType.
	ValueType	Referencing the table ValueType

5.10.16 NFProperty

Short for "NatureFrameworkProperty". Stores values of the properties considered within the nature framework. The nature framework properties and parameters for the properties are defined and typed in the table NFPropertyType and NFPropertyParameterType. If the value of the property is numeric it is stored in the Value field of the NFPAmountParamter table. See also Amount and Parameter chapters 5.13.1 and 5.13.2 respectively. The documentation of the meta data about the nature property is stored in the QMetaData table linked via the NFPDocumentation table, see QMetaData chapter 5.15.

Fields:	**NFId**	Reference to the nature framework model.
	PropertyNomenclature	Name of nomenclature in which the property was defined.
	PropertyEntity	Physical or other type of entity that the property represents.
	PropertyName	Name of a property.
	PropertyParameter	Name of a property parameter.
	MeasurementNumber	Number of the measurement. To allow the storage of several independent measurements. (This should not

		be confused with samples in a measurement series. In this case it is the same measurement and the distinction is made clear by the amount parameter.)
	TextValue	Holds the data for the specific parameter if the amount structure is not used for this parameter (analogous to SubstanceProperty).
	Unit	The unit of the parameter.
	Notes	Any additional relevant information.
Foreign keys:	NFId	Referencing the table NatureFramework.
	PropertyNomenclature, PropertyEntity, PropertyName, PropertyParameter	Referencing the table NFPropertyParameterType.
	Unit	Referencing the table Unit.

5.11 Data structure for priorities and values

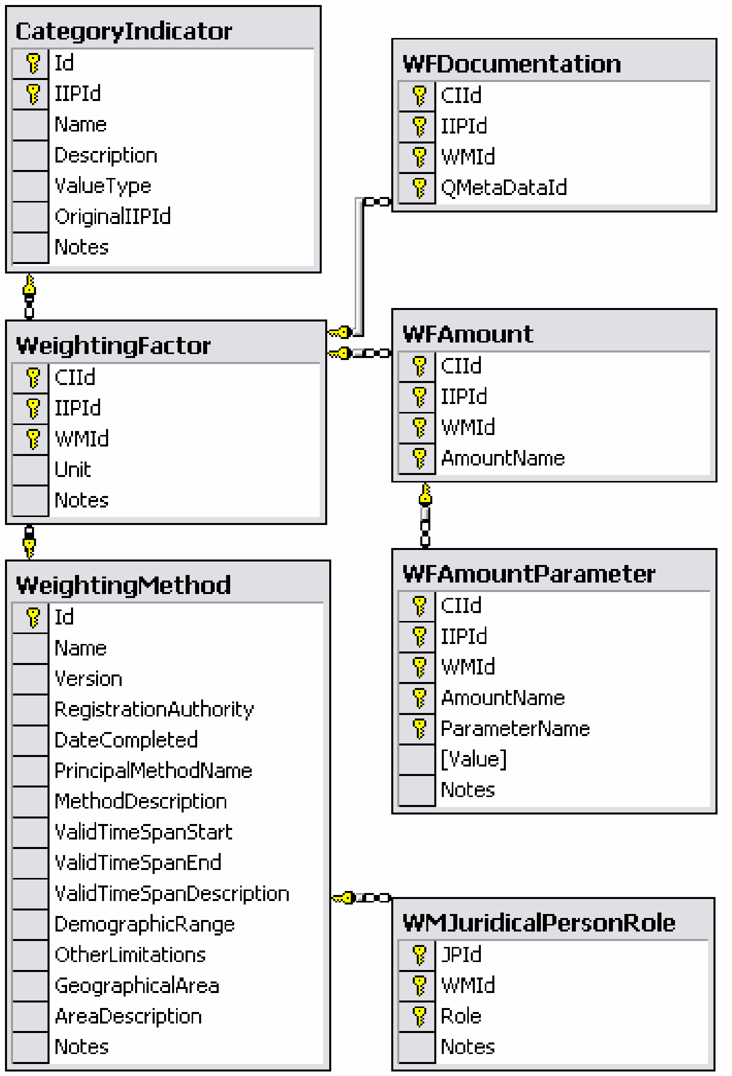

Figure 22 The tables in the IMPRESS data format related to priorities and values

5.11.1 WeightingMethod

Environmental indicators are all important, but in order to prioritise the work within a project different indicators need to be associated with different weights. There may be many different ways to make this weighting. This table holds information of such weighting methods.

Fields:	**Id**	Unique identifier of the weighting method.
	Name	Name of the weighting method.
	Version	If the weighting method is updated the updated versions are given successive version numbers.
	RegistrationAuthority	Name of organisation endorsing the documentation of the weighting method.
	DateCompleted	The date when the method was finalised or published.
	PrincipalMethodName	Weighting methods often are variants of principal methods. For clarification the principal method can be named.
	MethodDescription	A description of the method that has been used when calculation the indicator.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Date of the start valid time span of the method
	ValidTimeSpanEnd	Date of the end valid time span of the method
	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the method.
	GeographicalArea	A reference to a geographical nomenclature, addressing the geographical area for the range of the method described.
	AreaDescription	A description of the applicable geographical area.
	OtherLimitations	Any other limits in terms of system boundaries or applicability of the method.
	Notes	Any additional relevant information.
Foreign	GeographicalArea	Referencing the table Geography.

keys:		

5.11.2 WeightingFactor

A weighting method is associated with a set of category indicators, and each indicator is associated with a weighting factor, expressing this indicators relative weight to the other indicators in that set.

Fields:	CIId	Referece to the category indicator.
	IIPId	Referece to the IIP.
	WMId	Referece to the weighting method.
	Unit	The unit for the weighting factor.
	Notes	Any additional relevant information.
Foreign keys:	CIName, IIPName, IIPVersion	Referencing the table CategoryIndicator
	WMName, WMVersion	Referencing the table WeightingMethod.
	Unit	Referencing the table Unit.

5.12 Environmental Impact Assessment

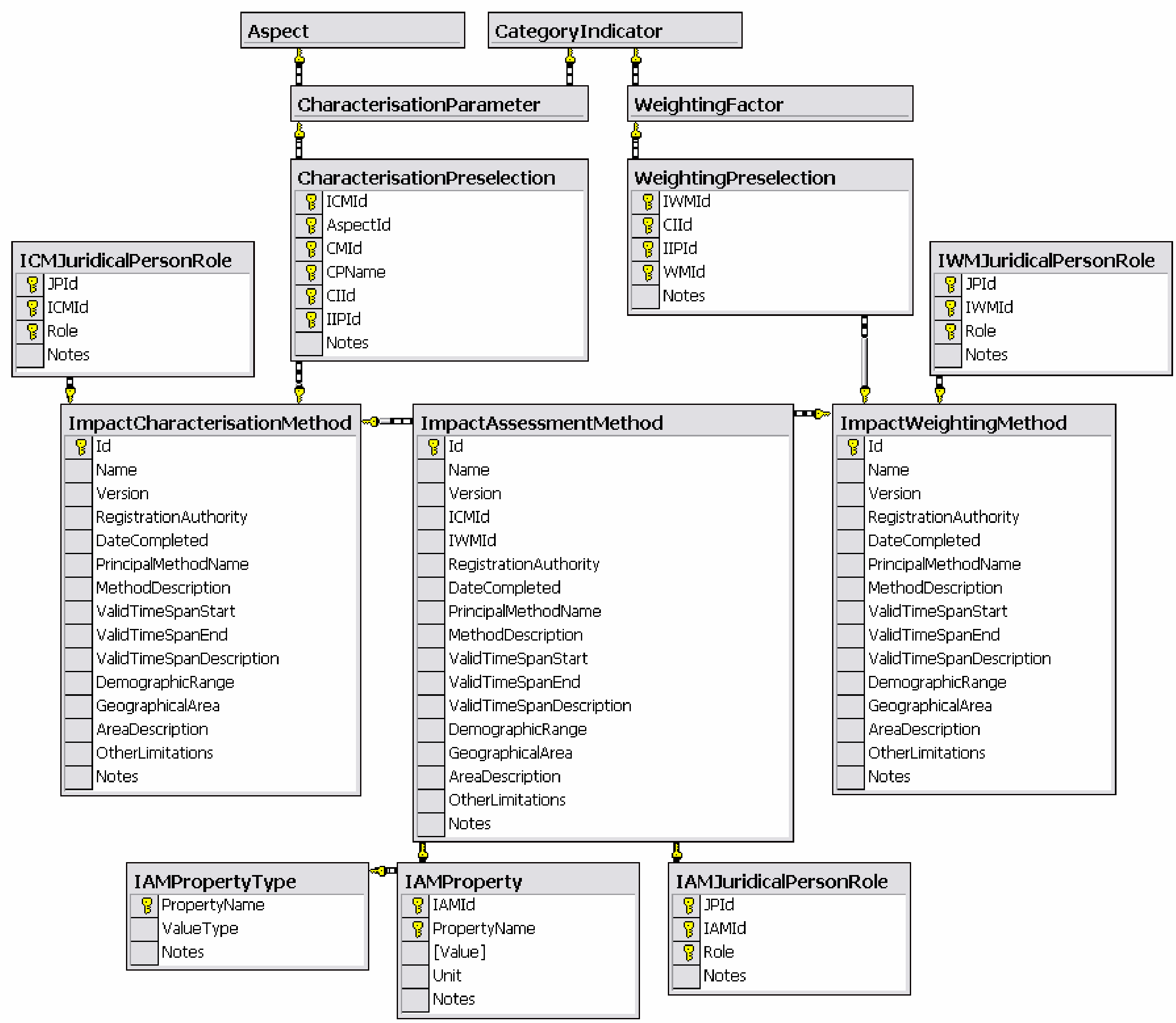

Figure 23 Main tables in the IMPRESS data format representing environmental impact assessment

The data format representing environmental impact assessment is based on LCA modelling concepts according to the ISO 14040 standard series. The values are transparently documented in terms of the models and principles applied to acquire the quantitative values. This includes the principle for selecting aspects and indicators, the characterisation and weighting methods. A set of characterisation parameters and weighting factors can be selected and connected in an impact assessment method that can be applied in a e.g. an LCA study or in evaluating the environmental performance of a design.

It is in general regarded as a complex and difficult task to select appropriate elements for an impact assessment. This includes selecting relevant indicators, applicable characterisation parameters and weighting factors that match the goal and scope of a study. One strongly contributing reason for this is that this relation is connecting the technical system to the environmental system; two different worlds are hence being joined. To make the model more user-friendly the possibility to add ready-made connections between flows and characterisation models has been included (CharacterisationPreselection). Such ready-made connections are among others delivered from the EPS-model. The model does also support ready-made weighting of indicators (WeightingPreselection). Normally both connection of aspects with characterisation methods and weighting is performed in an Impact Assessment Method. These two concepts are separated in the tables ImpactCharacterisationMethod and ImpactWeightingMethod.

5.12.1 CharacterisationPreselection

Table with ready-made connections of flows and characterisation methods.

Fields:	**ICMId**	Reference to the impact characterisation method.
	AspectId	Reference to the aspect.
	CMId	Reference to the characterisation method.
	CPName	Reference to the characterisation parameter.
	CIId	Reference to the category indicator.
	IIPId	Reference to the impact indication principle.
	Notes	Any additional relevant information.
Foreign keys:	ICMId	Referencing the table ImpactCharacterisationMethod.

	Aspectid, CMId, CPName, CIId, IIPId	Referencing the table CharacterisationParameter.

5.12.2 ImpactCharacterisationMethod

Fields:	**Id**	Unique identifier of the impact characterisation method.
	Name	Name of the method that has been used for the development of a set of ready-made weightings of category indicators.
	Version	If the weighting method is updated the updated versions are given successive version numbers.
	RegistrationAuthority	Name of organisation endorsing the documentation of the weighting method.
	DateCompleted	The date when the method was finalised or published.
	PrincipalMethodName	Impact characterisation methods often are variants of principal methods. For clarification the principal method can be named.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	MethodDescription	A description of the method that has been used when calculation the indicator.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Date of the start valid time span of the method
	ValidTimeSpanEnd	Date of the end valid time span of the method

	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the method.
	GeographicalArea	A reference to a geographical nomenclature, addressing the geographical area for the range of the method described.
	AreaDescription	A description of the applicable geographical area.
	OtherLimitations	Any other limits in terms of system boundaries or applicability of the method.
	Notes	Any additional relevant information.
Foreign keys:	GeographicalArea	Referencing the table Geography.

5.12.3 WeightingPreselection

Table with selections of weighting of indicators. To make the weightings relevant pre-selections from different weighting methods should not be mixed.

Fields:	**IWMId**	Reference to the category impact weighting method.
	CIId	Reference to the category indicator.
	IIPId	Reference to the category impact indication principle.
	WMId	Reference to the category weighting method.
	Notes	Any additional relevant information.
Foreign keys:	IWMId	Referencing the table ImpactWeightingMethod.
	CIId, IIPId, WMId	Referencing the table WeightingFactor.

5.12.4 ImpactWeightingMethod

Fields:	**Id**	Unique identifier of the impact weighting method.
	Name	Name of the method that has been used for the development of a set of ready-made weightings of

		category indicators.
	Version	If the weighting method is updated the updated versions are given successive version numbers.
	RegistrationAuthority	Name of organisation endorsing the documentation of the weighting method.
	DateCompleted	The date when the method was finalised or published.
	PrincipalMethodName	Impact weighting methods often are variants of principal methods. For clarification the principal method can be named.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	MethodDescription	A description of the method that has been used when calculation the indicator.
	ValidTimeSpanStart	Date of the start valid time span of the method
	ValidTimeSpanEnd	Date of the end valid time span of the method
	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the method.
	GeographicalArea	A reference to a geographical nomenclature, addressing the geographical area for the range of the method described.
	AreaDescription	A description of the applicable geographical area.
	OtherLimitations	Any other limits in terms of system boundaries or applicability of the method.
	Notes	Any additional relevant information.
Foreign keys:	N/A	

5.12.5 ImpactAssessmentMethod

Fields:	**Id**	Unique identifier of the impact characterisation method.
	Name	Name of the method that has been used for the development of a set of ready-made weightings of category indicators.
	Version	If the weighting method is updated the updated versions are given successive version numbers.
	ICMId	Reference to the impact characterisation method.
	IWMId	Reference to the impact weighting method.
	RegistrationAuthority	Name of organisation endorsing the documentation of the weighting method.
	DateCompleted	The date when the method was finalised or published.
	PrincipalMethodName	Impact assessment methods often are variants of principal methods. For clarification the principal method can be named.
	MethodDescription	A description of the method that has been used when calculation the indicator.
	MethodologicalRange	The geographical, technical, environmental etc. conditions under which the method may be applicable. It is recommended to use the below more highly resolved fields.
	ValidTimeSpanStart	Date of the start valid time span of the method
	ValidTimeSpanEnd	Date of the end valid time span of the method
	ValidTimeSpanDescription	Description of the valid time span
	DemographicRange	Description of valid demographical limitations of the method.
	GeographicalArea	A reference to a geographical nomenclature, addressing the geographical area for the range of

		the method described.
	AreaDescription	A description of the applicable geographical area.
	OtherLimitations	Any other limits in terms of system boundaries or applicability of the method.
	Notes	Any additional relevant information.
Foreign keys:	ICMId	Referencing the table ImpactCharacterisationMethod.
	IWMId	Referencing the table ImpactWeightingMethod.
	GeographicalArea	Referencing the table Geography.

5.12.6 IAMPropertyType

Additional properties of the impact assessment method may be defined. This table store the definitions of applicable properties.

Fields:	**PropertyName**	The name of the property.
	ValueType	The value type of the property.
	Notes	Additional relevant information.
Foreign keys:	ValueType	Referencing the table ValueType

5.12.7 IAMProperty

This table store the values of properties for specific impact assessment methods.

Fields:	**IAMId**	Reference to the impact assessment method.
	PropertyName	The name of the property.
	Value	The value of the property.
	Unit	The unit of the property.
	Notes	Additional relevant information.
Foreign keys:	ValueType	Referencing the table ValueType

5.13 Amount structure

5.13.1 Amount

The amount structure enables each numerical value in the database to be associated with any number of parameters and a distribution function. This makes it possible not only to store a single value, but also to say something about the uncertainty for the value by for example storing min and max or all the sampling points. The amount should be documented in terms of statistical properties, i.e. the name of a distribution function, names of parameters of the distribution function and quantitative values on each parameter. String values are stored in the corresponding 'TextValue'-field of properties, etc. The 'TextValue'-field may also be useful when exporting data to another data model.

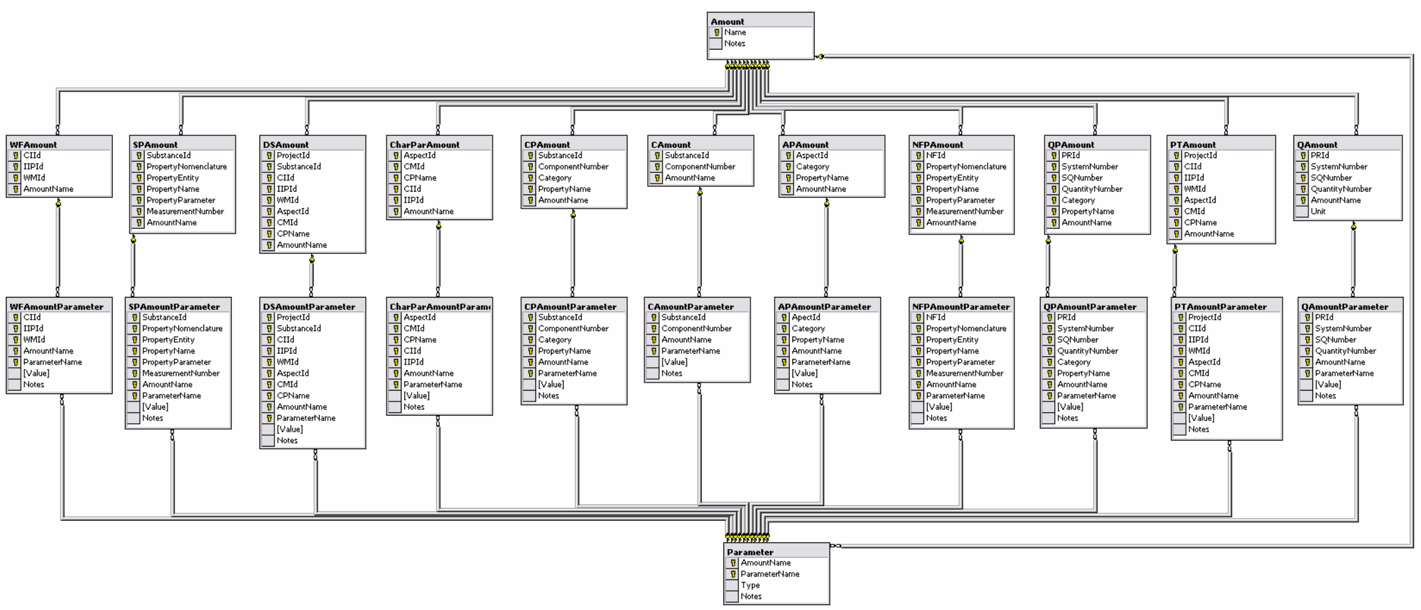

Figure 24 The tables in the IMPRESS data format related to the amount structure for storing quantitative values.

The amount table contains a nomenclature with all the available distribution functions in the information system. It is important that the distribution function used to describe an amount can be identified by a commonly understood name. The available frequency functions and their related parameters shall be documented in system documentation, user manuals and in the software code.

Fields:	**Name**	Name of a frequency function. Examples: "Single point", "Range", “Mean”, “Mode”, "N-distribution".
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.13.2 Parameter

Each amount distribution has a set of parameters. For example, in practice, data are often available in the form of ranges expressed by the parameter's minimum value and maximum value. If, in addition, the sample size and the mode (the most likely value) are known, the coefficient of variance can easily be calculated.

Fields:	**AmountName**	Name of a frequency function. Examples: "Single point", "Range", “Mean”, “Mode”, "N-distribution". "N-distribution"
	ParameterName	Name of an amount parameter for the given amount name. Examples: "Single point", "Minimum value", "Mean", "Standard deviation". "Single point", "Minimum value", "Standard deviation", "Mean"
	Type	Specification of data type of the parameter. Examples: “Integer”, “Real”.
	Notes	Additional relevant information.
Foreign keys:	AmountName	Referencing the table AmountName.

5.13.3 APAmount, CAmount, CharParAmount, CPAmount, DSAmount, NFPAmount, PTAmount, QAmount, QPAmount, SPAmount, WFAmount,

Amounts are connected to a property through bridging tables.

APAmount connects an **AspectProperty** to one or several amounts.

CAmount connects a **Composition** to one or several amounts.

CharParAmount connects a **CharacterisationParameter** to one or several amounts.

CPAmount connects a **CompositionProperty** to one or several amounts.

DSAmount connects a **DesignScore** to one or several amounts.

NFPAmount connects a **NatureFrameworkProperty** to one or several amounts.

PTAmount connects a **ProjectTarget** to one or several amounts.

QAmount connects a **Quantity** (technical system flows) to one or several amounts.

QPAmount connects a **QuantityProperty** to one or several amounts.

SPAmount connects a **SubstanceProperty** to one or several amounts.

WFAmount connects a **WeightingFactor** to one or several amounts.

5.13.4 APAmountParameter, CAmountParameter, CharParAmountParameter, CPAmountParameter, DSAmountParameter, NFPAmountParameter, PTAmountParameter, QAmountParameter, QPAmountParameter, SPAmountParameter, WFAmountParameter

Tables to store the values of the parameters within the amounts connected to a property.

5.13.5 Context

The Context table stores (implicit) references to any other table, column, or record in a table in a database.

Fields:	**Id**	Unique identification of the Context.
	PlatformAddress	Used when the Context refers to an external data source.
Foreign keys:	N/A	

5.13.6 ContextPointer

The ContextPointer defines the table, column and field(s) that the Context refers to. The reference should contain information equivalent to an SQL SELECT clause.

Fields:	**ContextId**	Unique identification of the context.
	Table	Name of the referred table.
	Field	Name of the referred column in the table.
	Value	The valid value of the field. All posts matching the Value in the specified Field and Table are referred. If any match is valid the asterisk "*" character is used.
Foreign keys:	ContextId	Referencing the table Context.

5.14 ValueType – definitions of valid values

5.14.1 ValueType

This table is used to define the valid values of properties etc.

Fields:	Id	Unique identifier of the value type.
	BaseType	The possible values for the base type are: ‘Quantitative’, ‘StringNomenclature’, or ‘String’. -‘Numeric’ shall be used for all numeric figures compatible with mathematical operators. It is only for this base type that the amount structure is being used. -‘StringNomenclature’ shall be used for strings related to a countable. An example of a StringNomenclature’ is Boolean, which has the range ‘False’ and ‘True’, or ToxClass, which has the range ‘EcoTox’, ‘HumanTox’ and NonTox’. -‘String’ shall be used for anything else.
	DefaultAmount	To enable easy creation of several properties with the same amount, it is also possible to specify a default amount.
	DefaultUnit	The default unit for the value type.
	Notes	Additional relevant information.
Foreign keys:	DefaultAmount	Referencing the table AmountName.
	DefaultUnit	Referencing the table Unit.

5.14.2 Range

Specifies the valid range for a “string nomenclature” value type. (Not used for numeric value type). This table can be used to specify allowed string values for a property.

Fields:	ValueType	Name of the value type.
	FieldValue	Examples: “Yellow”, “Yes”, “False”, “Hazardous”.
	Notes	Additional relevant information.
Foreign keys:	ValueType	Referencing the table ValueType.

5.15 QMetaData – documentation about data

5.15.1 QMetaData

The table QMetaData stores meta data for all types of quantitative data.

Fields:	**Id**	Unique identifier of the QMetaData.
	DataCollection	The method used to acquire the data. Same nomenclature as observation method.
	DataCollectionNomenclature	Nomenclature of the data collection method
	CollectionDateStart	The start of the time period during which the data was collected.
	CollectionDateEnd	The end of the time period during which the data was collected.
	DataTreatment	Description of the methods, sources and assumptions used to generate, recalculate, and reformat the amount.
	Notes	Additional relevant information.
Foreign keys:	DataCollection, DataCollectionNomenclature	Referencing the table ObservationMethod.

5.15.2 DataValidation

Documentation of any validations that have been performed on the entered data. A validation can be performed in a number of ways by many different persons. Validation refers to e.g. checks made when entering the data, checks made by the data generator, and third party peer reviews. Knowledge of each individual validation and its result is of importance for a data user or reviewer when assessing the reliability of data. Each validation is described using the terms in of a Method, Procedure, Result and a Validator.

Fields:	**QMetaDataId**	Unique identifier of the meta data.
	ValidationNumber	Ordering number for the validation.
	MethodName	The name of the validation method. Examples:

		“Onsite validation”, “Recalculation”, “Mass balance”, “Cross-check with other source”, “Proofreading of data entries”.
	MethodNomenclature	Nomenclature for the validation method
	ValidationProcedure	Description of the quality aspect that has been checked. Examples: “Mass balance of raw material and incoming package material checked to mass of outgoing waste and packaged products” or “Result benchmarked with expert having years of experience from similar measurements.”
	Result	Description of the result of the validation. Examples: “A deviation of 3% was found on the raw material versus product and waste. This may be acceptable.” or “The value for SO_2 seems a little high, but that may be due to the quality of oil used for the heating.” Also, if errors or missing data were identified, but no corrections were made to the data, then the validation findings should be given here.
	Validator	The identity, competence, name, organization and address of the person performing the validation.
Foreign keys:	QMetaDataId	Referencing the table QMetaData.
	MethodName, MethodNomenclature	Referencing the table ValidationMethod.
	Validator	Referencing the table JuridicalPerson.

5.15.3 DataAdministrativeInformation

A table to facilitate the administration of data.

Fields:	**QMetaDataId**	Identification number to uniquely identify the documented data.

	Date_completed	The date at which the data were finally completed, edited or updated.
	Access_restrictions	An unambiguous flag indicating how the document may be spread outside of the information system within which it is contained.
Foreign keys:	QMetaDataId	Referencing the table QMetaDataId.

5.15.4 CDocumentation, CharParDocumentation, CPDocumentation, DSDocumentation, NFPDocumentation, PTDocumentation, QDocumentation, QPDocumentation, SPDocumentation, WFDocumentation

This table links a specific data to the documentation stored in the QMetaData table, see QMetaData chapter 5.15.

CDocumentation links a composition to documentation in the QMetaData table.
CharParDocumentation links a characterisation parameter to documentation in the QMetaData table.
CPDocumentation links a composition property to documentation in the QMetaData table.
DSDocumentation links a design score to documentation in the QMetaData table.
NFPDocumentation links a nature framework property to documentation in the QMetaData table.
PTDocumentation links a project target to documentation in the QMetaData table.
QDocumentation links a quantity to documentation in the QMetaData table.
QPDocumentation links a quantity property to documentation in the QMetaData table.
SPDocumentation links a substance property to documentation in the QMetaData table.
WFDocumentation links a weighting factor to documentation in the QMetaData table.

The same principal structure is used for all these tables. The table QDocumentation for linking documentation about to a Quantity is shown below.

Fields:	**PRId**	Reference to the physical reference
	SystemNumber	Reference to the system model.
	SQNumber	Reference to the system quantification.
	QuantityNumber	Reference to the quantity.
	QMetaDataId	Reference to the meta data about the quantity.
Foreign keys:	PRId, SystemNumber, SQNumber QuantityNumber	Referencing the table Quantity.
	QMetaDataId	Referencing the table QMetaData.

5.16 Juridical Person

5.16.1 JuridicalPerson

This table store a list of juridical persons including individual persons or organisations. Contact information including address, telephone e-mail etc can be stored as JPProperties, or merged in the field MergedInformation if there is no need to have a resolve this information further.

Fields:	**Id**	Unique identifier of the juridical person
	Name	Name of the juridical person.
	MergedInformation	Information about the juridical person that is not distinguished as JPProperties. E.g. contact details, such as address, email, telephone etc.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.16.2 JPPropertyType

This table specifies types of JP properties. Depending on the application it may be relevant to distinguish e.g. e-mail, mail address, telephone, etc.

Fields:	**Category**	The category of the property
	PropertyName	The name of the juridical person property, e.g. Mail address, Telephone, etc.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.16.3 JPProperty

This table stores values of JP properties.

Fields:	**JPId**	Reference to a specific juridical person.
	Category	The category of the property.

	PropertyName	The name of the juridical person property.
	Value	The value of the property.
	Notes	Additional relevant information.
Foreign keys:	JPId	Referencing the table JuridicalPerson.
	Category, PropertyName	Referencing the table JPPropertyType.

5.16.4 JuridicalPersonRoleType

This table specifies types of roles that juridical persons can have in different contexts, e.g. in relation to a physical reference or a system model. Examples of possible roles are: Supplier (the wholesaler), Supplier contact, Producer, Producer contact, Production facility, Production facility contact, Modeller, Documenter, etc.

Fields:	**Role**	Name of the role of the juridical person.
	Description	Explanatory description of the role type.
Foreign keys:	N/A	

5.16.5 ASPJuridicalPersonRole, CMJuridicalPersonRole, DataJuridicalPersonRole, IAMJuridicalPersonRole, ICMJuridicalPersonRole, IIPJuridicalPersonRole, IWMJuridicalPersonRole, MechanismJuridicalPersonRole, NFJuridicalPersonRole, NFPTJuridicalPersonRole, PRJuridicalPersonRole, QuantityJuridicalPersonRole, SPTJuridicalPersonRole, SystemJuridicalPersonRole, SQJuridicalPersonRole, WMJuridicalPersonRole

The role of a juridical person can be specified in relation to a data collection rule, impact characterization method, impact indication principle, impact weighting method, load selection principle, mechanism, nature framework, physical reference, quantity, system, system quantification, and a weighting method.

The role of a juridical person in relation to an aspect selection principle is stored in ASPJuridicalPersonRole.
The role of a juridical person in relation to an impact characterization method is stored in CMJuridicalPersonRole.
The role of a juridical person in relation to QMetaData documentation is stored in DataJuridicalPersonRole.
The role of a juridical person in relation to an impact assessment method principle is stored in IAMJuridicalPersonRole.
The role of a juridical person in relation to an impact characterisation method principle is stored in ICMJuridicalPersonRole.
The role of a juridical person in relation to an impact indication principle is stored in IIPJuridicalPersonRole.
The role of a juridical person in relation to an impact weighting method is stored in IWMJuridicalPersonRole.
The role of a juridical person in relation to a mechanism model is stored in MechanismJuridicalPersonRole.
The role of a juridical person in relation to a nature framework is stored in NFJuridicalPersonRole.
The role of a juridical person in relation to a nature framework property type is stored in NFPTJuridicalPersonRole.
The role of a juridical person in relation to a physical reference is stored in PRJuridicalPersonRole.
The role of a juridical person in relation to a quantity is stored in QuantityJuridicalPersonRole.
The role of a juridical person in relation to a substance property type is stored in SPTJuridicalPersonRole.
The role of a juridical person in relation to a system quantification is stored in SQJuridicalPersonRole.
The role of a juridical person in relation to a system is stored in SystemJuridicalPersonRole.
The role of a juridical person in relation to a weighting method is stored in WMJuridicalPersonRole.

Here the specification for the table PRJuridicalPersonRole follows. The other tables are defined in the same way.

Fields:	**JPId**	Identification number to uniquely identify the juridical person.
	PRId	Identification number to uniquely identify the physical reference.
	Role	Role of the juridical person.
	Notes	Any additional relevant information.
Foreign keys:	JPId	Referencing the table JuridicalPerson.
	PRId	Referencing the table PhysicalReference.
	Role	Referencing the table JuridicalPersonRole.

5.16.6 JPContextType

Additional specification to facilitate correct interpretation of the data pointed out by the context.

Fields:	Name	Type of context. Examples: “Owner”, “IIPPointer”, “SystemPointer”
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.16.7 JPContext

A juridical person can be related to any set of data through this table.

Fields:	JPId	Identifier of the juridical person
	ContextId	Identifier of the context
	ContextType	The type of context
	Notes	Additional relevant information
Foreign keys:	JPId	Referencing the table JuridicalPerson
	ContextId	Referencing the table Context

	ContextType	Referencing the table JPContextType

5.17 Nomenclatures

A nomenclature is an agreed collection of names which is used to identify different entities. For example the CAS registry number nomenclature is a collection of names for identifying substances (the names in this case are in the form of numbers). Nomenclatures is useful as the users have agreed on what the objects are called things the amount of possibilities of names. Hence it helps the user to manage data and and also provides examples of what kind of data can is valid for a field which is tied to a nomenclature. All nomenclatures require administration to stay operational which involves a cost which means there is not practical to create .nomenclatures for any type of entity. A nomenclature should be based on physical relations in reality, to avoid the opposite where users are forced to adopt reality to an the nomenclature. A clear rule to add new names in the nomenclature should be defined. If this cannot be done or the rule is meaningless it is a signal that a nomenclature should not be used.

Some nomenclatures include a categorization of names for grouping the different names in the nomenclature. For example a material nomenclature may contain a three categories: wood-based plastics, and metal based materials. Categories are useful to make date search simpler. Rules on data processing can also be created based on the categories, e.g. all wood-based materials have are considered to be renewable. It is also possible to allow a name to belong to several categories. In such cases the categories are overlapping and care must be take when creating category based rules. Categories are abstractions and can be misinterpreted as physical entities. Hence it is important that the user understands the ontological meaning of the category. If the nomenclature only contains a few names it is unnecessary to sort them in categories.

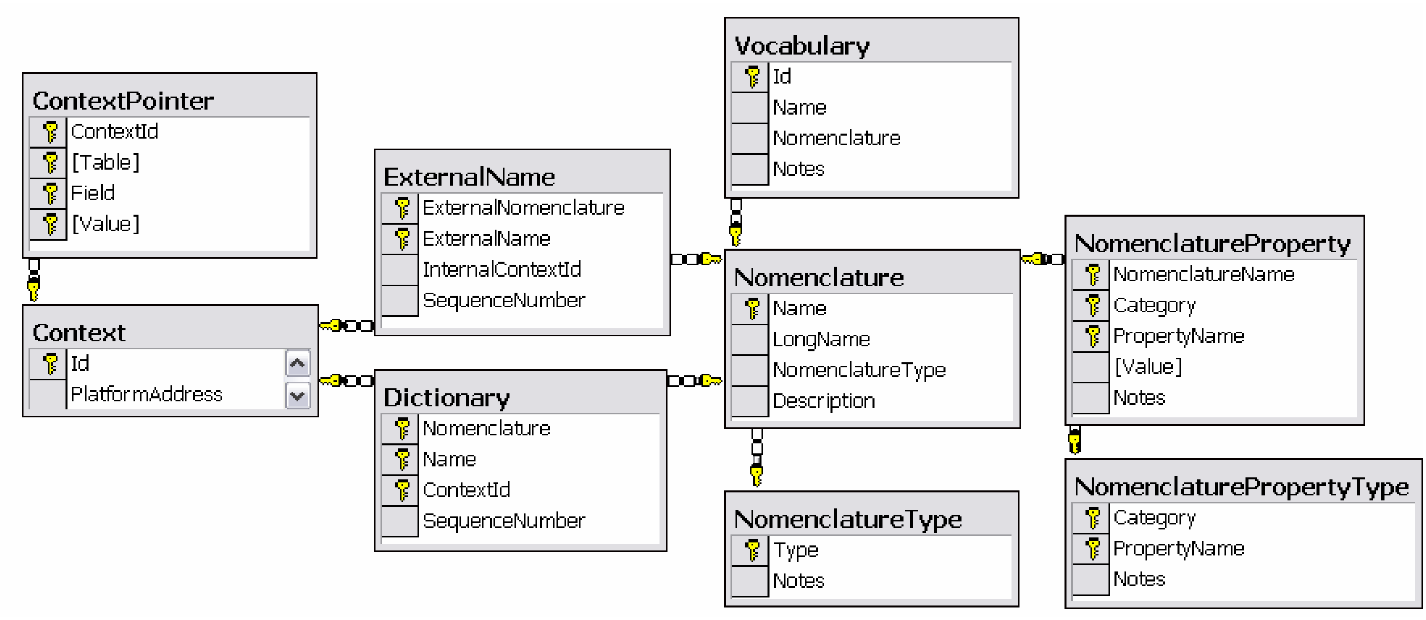

Figure 25 The tables in the IMPRESS data format related to nomenclature.

5.17.1 Nomenclature

This table stores the names of nomenclatures used in an information system. Note that some concepts that have a nomenclature in the IMPRESS data format do not refer to this table. This includes basic concepts such as Unit and Entity or tables that represent a specific nomenclature such as ISO/TS 14048 technical scope.

Fields:	Name	Name of the nomenclature.
	NomenclatureType	Specification of the type of nomenclature.
	Description	A description of the nomenclature.
Foreign keys:	NomenclatureType	Referencing the table NomenclatureType.

5.17.2 NomenclatureType

This table stores nomenclatures types. An ontology is a nomenclature where there is a one-to-one mapping between name and item (e.g. substance). In non-exclusive nomenclatures an item may have several names and/or a name may refer to several items. In categorised nomenclatures a name in the nomenclature may collectively represent many other names, e.g. in a substance nomenclature the name steel may represent many names for specific types of steel.

Fields:	Type	Specification of the nomenclature type. Examples: “Ontology”, “Ontology Categorized”, “Non-exclusive”, “Non-exclusive Categorized”.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.17.3 Vocabulary

Stores names of the vocabulary used within a specific context such as within a specific software or a project. The vocabulary refers to one or more nomenclatures used in the specific application.

Fields:	**Name**	The name of the vocabulary, examples: “OMNIITOX”, “REPID”, “WWLCAW”, “SPINE@CPM”.
	Notes	Additional relevant information**.**
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.4 VocabularyNomenclatures

Stores names of the vocabulary used within a specific context such as within a specific software or a project. The vocabulary refers to one or more nomenclatures used in the specific application.

Fields:	**Vocabulary**	Reference to the vocabulary.
	Nomenclature	Reference to a nomenclature used within the vocabulary.
	Notes	Additional relevant information**.**
Foreign keys:	Vocabulary	Referencing the table Vocabulary.
	Nomenclature	Referencing the table Nomenclature.

5.17.5 Dictionary

A dictionary is used to store alternative names. If for example a company buys screws named M10 and uses them in two different products, the same screw will in some most cases be assigned different article numbers based on where the screw is being used. The same thing will for that reason have two different names in the company's dictionary "Article number".

Fields:	**Nomenclature**	The name of the nomenclature in which the name of the thing is defined.
	Name	A name of a thing in the nomenclature in the dictionary.
	ContextId	The thing or context that the name is referring to.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.
	ContextId	Referencing the table Context.

5.17.6 ExternalNomenclature

When acquiring data from the outside an information platform, a look-up table for translation from external nomenclatures is needed.
In contrast to the dictionary a this table puts strict requirements that a name may not occur more than once in the same nomenclature. This type of nomenclature is called ontology, which implies that two things may not have the same name in one nomenclature. Since each thing only may be assigned a single name in an ontology, a restriction that the fields ExternalNomenclature and InternalContextId must be unique is added to the data model.

Fields:	**ExternalNomenclature**	The name of the external nomenclature in which the name of the thing is defined. Example: CAS
	ExternalName	A name of a thing or context in the external nomenclature.
	InternalContextId	The context that the name is referring to.
Unique constraint:	ExternalName, InternalContextId	Each thing may only be assigned a single name in an ontology.

Foreign keys:	ExternalNomenclature	Referencing the table Nomenclature.
	InternalContextId	Referencing the table Context.

5.17.7 Entity

An entity represents a basic physical phenomenon. A list of entities is useful when managing many types of properties in various contexts from different data sources. The list of entities should be managed internally by the data source owner.

Fields:	**Name**	The name of the entity. Examples: “Weight”, “Length”, “Price”, “Concentration of hydrogen ions”.
	Description	Explanatory description of the entity.
Foreign keys:	N/A	

5.17.8 Unit

Nomenclature of units.

Fields:	**Name**	Name of the unit. Examples: “meter”, “kilogram”, “Newton”, “Mega Watt”
	Symbol	The symbol used to specify the Unit. Examples: “kg”, “m”, “N”, “MW”. (Collation for this field is case sensitive in order to distinguish between e.g. H (henry) and h (hours))
	Entity	Specification of what entity the unit represents. Examples: “length”, “mass”, “force”, “energy”, “luminosity”.
	Notes	Additional relevant information.
Foreign keys:	Entity	Referencing the table Entity.

5.17.9 UnitConversion

This table stores information on how units can be converted, e.g. 1 millimetre equals 0.001 metres. This enables automatic unit conversion to enhance comparability between datasets.

Fields:	**UnitName**	Name of the unit, e.g. (1) “metre”.
	BaseUnitName	The name of the corresponding unit of the same quantity, e.g. “millimetre”.
	Factor	Specification of the factor between the units, e.g. “1000” (1 metre = 1000 millimetre)
	Offset	Offset to match origins between units, e.g. if unit name is “Kelvin” and base unit name is “Degree Celsius” the offset is “-273.15”
Foreign keys:	UnitName	Referencing the table Unit.
	BaseUnitName	Referencing the table Unit.

5.17.10 UnitCategory

This table stores information on how units can be converted, e.g. 1 millimetre equals 0.001 metres. This enables automatic unit conversion to enhance comparability between datasets.

Fields:	**Nomenclature**	Name of the unit, e.g. (1) “metre”.
	Superior	The name of the corresponding unit of the same quantity, e.g. “millimetre”.
	Subordinate	
	GroupNumber	
	Notes	Additional relevant information.
Foreign keys:	UnitName	Referencing the table Unit.
	BaseUnitName	Referencing the table Unit.

5.17.11 PRClassNomenclature, SystemClassNomenclature, SQClassNomenclature

Nomenclatures for physical references, system models, and system quantifications. The field names for all three tables are identical. Only the reference from class name differs.

Fields:	**Class_name**	Name of the class. Examples of PR class: “Pulp and Paper production facility” Examples of System class: “Static process model” Examples of System class: “CEMS for CO2 emission monitoring.”
	Nomenclature	The nomenclature of the class name.
	Description	Description of the class.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.12 Iso14048_aggregation_type

Exclusive nomenclature in ISO/TS 14048 that is used to indicate aggregated system models, i.e. representing averages of several processes providing the same function (horizontal) or the sum of several interconnected systems.

Fields:	**Aggregation_type**	Name of aggregation type. Examples: “Non-aggregated”, “Horizontally aggregated”, “Vertically aggregated”, “Both horizontally” and “Vertically aggregated”
Foreign keys:	N/A	

5.17.13 Iso14048_technical_scope

A nomenclature in ISO/TS 14048 with short general descriptions of the technical scope of the process in terms of the operation(s) included in the data, using a. This may be one single operation or several operations covering the full lifecycle of a product, e.g. gate-to-gate or cradle-to-grave. This inclusive nomenclature may be extended by the user.

Fields:	**Technical_scope**	Name of technical scope. Examples: “Cradle-to-gate”, Cradle-to-grave”, “Gate-to-gate”, “Gate-to-grave”
	Nomenclature	The nomenclature of the class name.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.14 Iso14048_direction

A nomenclature in ISO/TS 14048 to specify the direction of inputs and outputs, i.e. as input to or output from a process. Direction is an exclusive nomenclature.

Fields:	**Direction**	Name of direction. Examples: “Inputs”, “Outputs”, “Non-flow-related aspects”
Foreign keys:	N/A	

5.17.15 Iso14048_group

A nomenclature in ISO/TS 14048 to specify the group to which the input or output belongs, e.g. Resource, Raw material, Emission, Product. The specification of group facilitates identification of the role of different inputs and outputs in the process. Group is an inclusive nomenclature.

Fields:	**Inputs_and_outputs__group**	Name of group. For examples see ISO/TS 14048.
	Nomenclature	The nomenclature of the name.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.16 Iso14048_type

Inclusive nomenclature in ISO/TS 14048 that specifies the type of quantitative reference, i.e. Functional unit, Reference flow of process, or Other flow. The quantitative reference is the reference to which the size of the inputs and outputs in the process relate. This is for example the functional unit (e.g. 1 ton·km) or reference flow (e.g. 1 kW·h electricity), which can be the input or output of another process.

Fields:	**Type**	Type of quantitative reference. For examples see ISO/TS 14048.
	Nomenclature	The nomenclature of the type.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.17 Iso14048_GIS_reference

Inclusive nomenclature that specifies one or more Geographical Information System (GIS) references identifiable in a geographical information system. The GIS reference can refer to a geographically positioned zone such as a circle or rectangle or a point.

Fields:	GIS_reference	Name of GIS reference. For examples see ISO 6709.
Foreign keys:	N/A	

5.17.18 Iso14048_receiving_environment

Exclusive nomenclature in ISO/TS 14048 indicating the media in which outputs and inputs are delivered from or to a process. For non-elementary inputs and outputs, the receiving environment is “Technosphere”, indicating that the input or output connects with another process. For elementary inputs and outputs, a simple nomenclature describes the type of environment that a resource is extracted from, or an emission is let out through; e.g. air, water, ground.

Fields:	**Receiving_environment**	Name of receiving environment. For examples see ISO/TS 14048.
Foreign keys:	N/A	

5.17.19 Iso14048_receiving_environment_specification

Inclusive nomenclature indicating the type of environment that an input or output impacts. For non-elementary inputs and outputs, the Receiving environment specification is “Technosphere”, indicating that the input or output is not subject to impact assessment. For elementary inputs and outputs, the inclusive nomenclature distinguishes between environment conditions at the start of a characterization modelling. These values are applied when matching characterisation parameters via aspects, see Aspect chapter 5.7.

Fields:	**Receiving_environment_specification**	Specification of receiving environment. For examples see ISO/TS 14048.
	Nomenclature	The nomenclature of the environment name.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.20 Iso14048_modelling_constants_name

Assumptions that have been held constant throughout the modelling of the process.

Fields:	**Modelling_constants_name**	Name of modelling constant. For examples see ISO/TS 14048.
	Nomenclature	The nomenclature of the modelling constant name.
Foreign keys:	Nomenclature	Referencing the table Nomenclature.

5.17.21 Geography

All names of geographical locations are stored in this table.

Fields:	**Id**	Name of modelling constant. For examples see ISO/TS 14048.
	AreaName	The name of the geographical area
	AreaType	The type of area, e.g. “Country”, “Continent”,

		"Ocean"
	SuperiorArea	Reference to a superior area.
	Notes	Additional relevant information
Foreign keys:	SuperiorArea	Referencing the table Geography

5.17.22 ObservationMethod

Nomenclature of methods used for observation/measurment of data. In most cases these methods specify different procedures to acquire data.

Fields:	**Name**	Name of the method used for observation of the data. Can be the name of a recognised standard method. "Any" method means that it makes no difference which observation method is used.
	Nomenclature	The nomenclature of the observation method
	MethodType	Type of observation method. Examples: "direct measurement", "statistical derivation", "other"
	Description	Description of the method.
	Notes	Additional relevant information.
Foreign keys:	Nomenclature	Referencing the table nomenclature.

5.17.23 ValidationMethod

Nomenclature of methods used for validation of entered data, system models and quantification of models.

Fields:	**Name**	Name of the method used for validation of the data. Examples: “Onsite validation”, “Recalculation”, “Mass balance”, “Cross-check with other source”, “Proofreading of data entries”.
	Nomenclature	The nomenclature of the validation method
	MethodType	Type of validation method. Examples: “General”, “Data validation”, “System validation”, “System Quantification validation”
	Description	General description of the validation method.
	Notes	Additional relevant information.
Foreign keys:	Nomenclature	Referencing the table nomenclature.

5.18 Document

The integrated data model facilitates organisation of documents. Documents may either be stored in a field in a local database, at other locations within in a local computer, at a physically distant server in a network, or they may exist physically as books or paper files. A library consists of a hierarchical structure of documents. Any document may be present in many libraries, and can be contextually related to data, concepts, software or even user events.

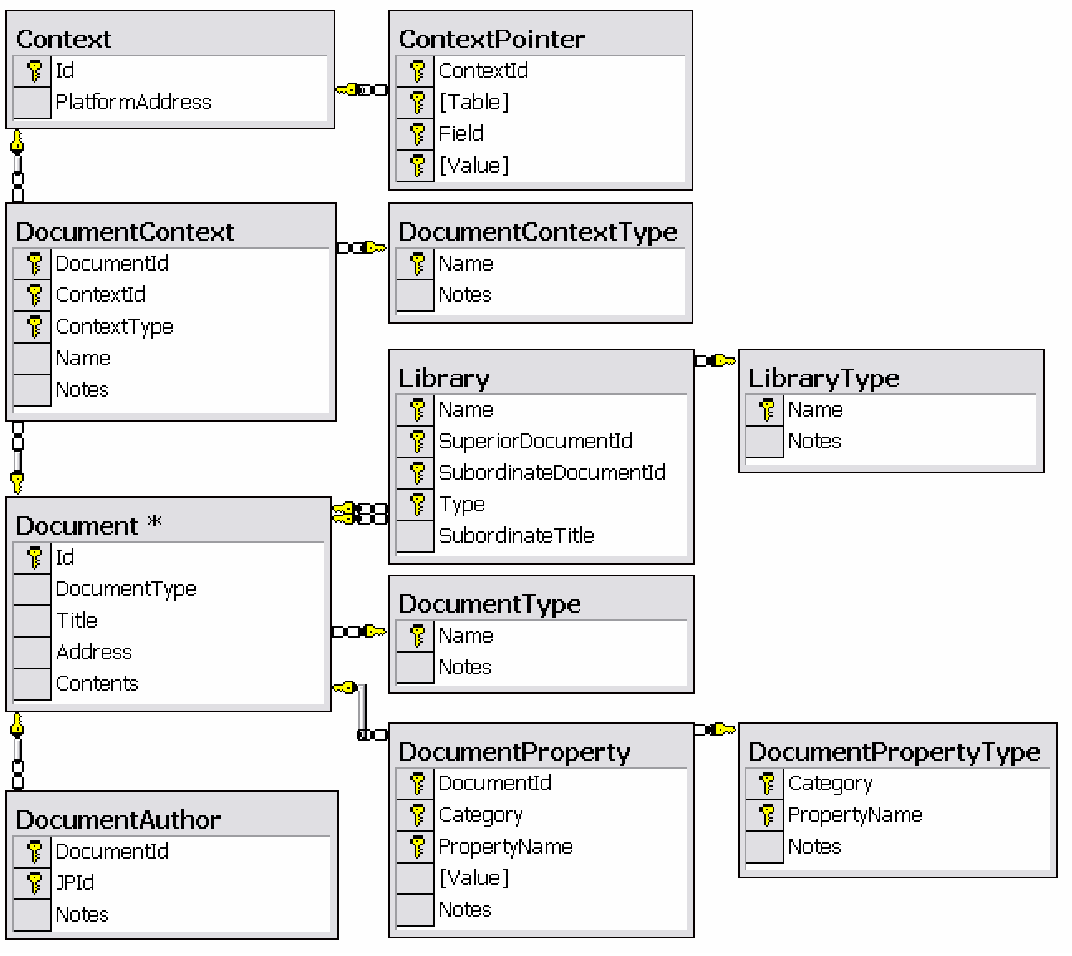

Figure 26 The tables in the IMPRESS data format related to document concept.

5.18.1 Document

Table for storage of documents or references (addresses) to documents.

Fields:	**Id**	Unique identifier of the document.
	DocumentType	A brief description of the type of document. Examples: “Article”, “Book”, “Contract”, “Glossary”.
	Title	Each document is given a title within the information system.
	Address	If the document is not inside the Contents-field, then this field supplies an address to the document.
	Contents	When stored in the database, a document is stored as a text string.
Foreign keys:	DocumentType	Referencing the table DocumentType.

5.18.2 DocumentType

Documents can be of any type of prescribed types.

Fields:	Name	A brief description of the type of Document. Examples: “Article”, “Book”, “Contract”, “Glossary”.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.18.3 DocumentPropertyType

The document properties are selected from a predefined set of allowed document property types. Compare with the document property types defined in i.e. Word to create a relevant set of document properties.

Fields:	Category	An overall category of different property types. Examples: “Keyword”, “Access restrictions”
	PropertyName	Name of document property, e.g. “Published year”
	Notes	Additional relevant information.
Foreign	N/A	

keys:		

5.18.4 DocumentProperty

When storing documents they may be stored with an arbitrary number of specifying properties, such as name and address of author, date of insertion etc.

Fields:	**DocumentId**	Reference to the document.
	Category	An overall category of different property types. Examples: “Access restrictions”
	PropertyName	Name of document property. Examples “Keyword”, “Published year”
	Value	Value of the property.
	Notes	Additional relevant information.
Foreign keys:	DocumentId	Referencing the table Document.
	Category, PropertyName	Referencing the table DocumentPropertyType.

5.18.5 DocuementContextType

Fields:	**Name**	Type of context. Examples: “Knowledge related”, “Software related”, “User related”, “Application related”.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.18.6 DocumentContext

Connect a document to a context by referencing the context table

Fields:	**DocumentId**	Reference to the document.
	ContextId	Reference to the context.

	ContextType	Type of context.
	Name	Name of document context. By using this field references in free text (i.e. [1]) can be handled, e.g. interactive links to documents in free texts can be created if the names follow a defined syntax that can be parsed by software applications.
	Notes	Additional relevant information.
Foreign keys:	DocumentId	Referencing the table Document.
	ContextId	Referencing the table Context.
	ContextType	Referencing the table DocumentContextType.

5.18.7 DocumentAuthor

Each document may be produced by many authors and an author may produce many documents. This table links documents with juridical persons.

Fields:	**DocumentId**	Reference to the document.
	JPId	Reference to the juridical person.
	Notes	Additional relevant information.
Foreign keys:	DocumentId	Referencing the table Document.
	JPId	Referencing the table JuridicalPerson.

5.18.8 Library

Each document may be produced by many authors and an author may produce many documents. This table links documents with juridical persons.

Fields:	**Name**	The name of the library.
	SuperiorDocumentId	Hierarchical structuring of documents.
	SubordinateDocumentId	Hierarchical structuring of documents.

	Type	There may be a need for e.g. a user to have libraries of different types.
	SubordinateTitle	It is possible to name the subordinate document in a way more appropriate for the user.
Foreign keys:	SuperiorDocumentId	Referencing the table Document.
	SubordinateDocumentId	Referencing the table Document.
	Type	Referencing the table LibraryType.

5.18.9 LibraryType

A defined set of library types.

Fields:	**Name**	The name of the library type.
	Notes	Additional relevant information.
Foreign keys:	N/A	

5.19 Additional support structures

5.19.1 IdSeq

Fields:	tableName	The table that the identifier sequence concerns.
	idValue	Each time a new record is inserted in the table referred by tableName the identifier shall use the integer value in the idValue field + 1. The idValue is then increased by 1.
	idString	Additional string to concatenate with the idValue, e.g. CPM[currentDate]
Foreign keys:	N/A	

5.19.2 SoftwareRegister

To aid the integration of new software tools each tool can be registered in the information platform.

Fields:	**SoftwareId**	An arbitrary identifier used enabling unique identification of the software.
	KeyName	Any identifier type within the software, such as a button, a text field, a user screen etc.
	KeyValue	A value which the e.g. button, text field user screen etc. should hold to trigger a defined action, such as retrieving a specific document from the knowledge system.
Foreign keys:	N/A	