

Catalogue OntoRelCat

Documentation du schéma relationnel

Table des matières

1	Introduction	3
2	Organisation du catalogue	3
3	Tables centrales	4
3.1	OntoRel_entity	4
3.2	OntoRel	5
3.3	OntoRel_schema	5
4	Tables des classes ontologiques	5
4.1	Onto_class	5
4.2	Onto_class_inheritance	6
4.3	Onto_class_axiom	6
5	Tables des propriétés d'objet	7
5.1	Onto_object_property	7
5.2	Onto_object_property_domain	7
5.3	Onto_object_property_range	7
5.4	Onto_object_property_inheritance	7
6	Tables des propriétés de données	8
6.1	Onto_data_property	8
6.2	Onto_data_property_domain	8
6.3	Onto_data_property_range	8
6.4	Onto_data_axiom	9
7	Tables des types de données	9
7.1	Onto_data_type	9
7.2	Onto_data_type_sql	9
8	Tables linguistiques et métadonnées	10
8.1	Languages_codes	10
8.2	Label	10
8.3	Definition	10

9	Tables de gestion des ontologies	11
9.1	Ontology	11
9.2	Ontology_label	11
9.3	OntoRel_schema_definition	12
10	Configuration système	12
10.1	Ontorel_db_config	12
11	Procédures stockées	12
11.1	Procédures principales	13
11.2	Fonctions utilitaires	13
12	Vues du catalogue	13
13	Conclusion	13

1 Introduction

Ce rapport documente le schéma relationnel du catalogue OntoRelCat, référentiel central du système OntoRelQuery. Le catalogue stocke les métadonnées ontologiques permettant la transformation des requêtes [OntoRelQuery Language \(ORQL\)](#) en code [Langage de requêtes structurées \(SQL\)](#) exécutable.

OntoRelCat implémente un mapping entre le modèle ontologique et le modèle relationnel, assurant la correspondance entre entités ontologiques ([Identificateur de ressource internationalisé \(IRI\)](#), classes, propriétés) et structures relationnelles (relations, colonnes, clés étrangères).

2 Organisation du catalogue

Le catalogue OntoRelCat est organisé autour de plusieurs groupes interconnectés. Le tableau 1 présente une synthèse de ces groupes avec leurs tables respectives et leur rôle dans le système.

TABLE 1 – Groupes du catalogue		
Groupe fonctionnel	Tables principales	Rôle
Tables centrales	OntoRel_entity, OntoRel, OntoRel_schema	Gestion des identités et versions
Classes ontologiques	Onto_class, Onto_class_inheritance, Onto_class_axiom	Structure taxonomique
Propriétés d'objet	Onto_object_property, tables de domaine/portée	Relations entre classes
Propriétés de données	Onto_data_property, tables de domaine/portée	Attributs des instances
Types de données	Onto_data_type, Onto_data_type_sql	Mapping ontologie-SQL
Métadonnées linguistiques	Languages_codes, Label, Definition	Support multilingue
Gestion ontologies	Ontology, Ontology_label	Ontologie
Configuration système	Ontorel_db_config	Paramètres de génération OntoRel

La figure 1 présente le schéma relationnel complet du catalogue avec toutes les tables et leurs relations.

— `ontorel_uuid` (FK) : Référence vers la version `OntoRel` associée

Rôle. Cette table sert de référence centrale pour toutes les entités ontologiques du catalogue. Chaque entité (classe, propriété d’objet, propriété de données, type de données) doit posséder une entrée dans cette table, garantissant l’unicité des IRI à travers le système. La table implémente une relation de généralisation/spécialisation (héritage) avec les tables `Onto_class`, `Onto_object_property`, `Onto_data_property` et `Onto_data_type`.

3.2 `OntoRel`

Contient les métadonnées versionnelles du système `OntoRel`, gérant l’évolution du modèle et la traçabilité des versions.

Structure.

- `ontorel_uuid` (PK) : Identifiant unique de la version `OntoRel`
- `version` : Numéro de version du modèle
- `ontorel_generation_date` : Date et heure de génération avec fuseau horaire

Rôle. Table centrale qui présente une instance complète du catalogue `OntoRelCat`. Elle établit les liens vers l’ontologie source, les configurations de génération et les paramètres de types de données. Chaque version `OntoRel` correspond à une génération spécifique du modèle relationnel à partir d’une ou plusieurs ontologies sources.

3.3 `OntoRel_schema`

Définit les schémas organisationnels au sein d’`OntoRel`, permettant le regroupement logique des tables relationnelles.

Structure.

- `ontorel_uuid` (FK) : Référence vers la version `OntoRel`
- `schema_name` (PK) : Nom unique du schéma

Rôle. Permet de structurer logiquement les tables générées en regroupant les entités ontologiques connexes. Les schémas facilitent l’organisation du modèle relationnel résultant et son déploiement dans des espaces de noms distincts au sein du [Système de gestion de bases de données \(SGBD\)](#).

4 Tables des classes ontologiques

Les tables de classes capturent la taxonomie ontologique et ses contraintes sémantiques, formant la structure hiérarchique du modèle.

4.1 `Onto_class`

Représente les classes ontologiques avec leur correspondance relationnelle directe.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `OntoRel_entity`) : IRI de la classe héritant de l'entité
- `table_id` : Identifiant de la table relationnelle correspondante
- `origin` : Source de la définition de la classe (ontologie d'origine)

Rôle. Table centrale pour la résolution des classes ontologiques lors de l'analyse des requêtes ORQL. Le champ `table_id` établit le lien direct vers la table relationnelle générée dans OntoRel, permettant la traduction des références ontologiques en requêtes SQL.

4.2 Onto_class_inheritance

Capture les relations de subsomption (héritage) entre classes, formant la hiérarchie taxonomique.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `superclass_iri` (PK, FK vers `Onto_class`) : IRI de la classe parente
- `subclass_iri` (PK, FK vers `Onto_class`) : IRI de la classe enfant

Rôle. Représente la relation *is-a* entre classes ontologiques. Cette structure est essentielle pour la navigation hiérarchique lors de l'exploration topologique avec le type de navigation ISA.

4.3 Onto_class_axiom

Stocke les axiomes de classes, définissant les contraintes sémantiques liées aux propriétés d'objet.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `domain_iri` (PK, FK vers `Onto_class`) : Classe de domaine
- `range_iri` (PK, FK vers `Onto_class`) : Classe de portée
- `property_iri` (PK, FK vers `Onto_object_property`) : Propriété d'objet concernée
- `table_id` (PK) : Identifiant de table pour les axiomes complexes
- `domain_card` : Cardinalité du domaine
- `range_card` : Cardinalité de la portée
- `origin` : Source de l'axiome

Rôle. Définit les règles d'utilisation des propriétés d'objet avec les classes, incluant les contraintes de cardinalité. Ces axiomes sont cruciaux pour la validation sémantique et la construction des jointures SQL lors de l'évaluation relationnelle.

5 Tables des propriétés d'objet

Les tables de propriétés d'objet cataloguent les relations sémantiques entre instances de classes.

5.1 `Onto_object_property`

Catalogue les propriétés d'objet du modèle avec leur correspondance relationnelle.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version `OntoRel`
- `iri` (PK, FK vers `OntoRel_entity`) : IRI de la propriété
- `table_id` : Identifiant de la table relationnelle ou colonne de clé étrangère

Rôle. Table centrale pour la résolution des propriétés d'objet. Le champ `table_id` peut référencer soit une table d'association (pour les relations many-to-many), soit une colonne de clé étrangère dans une table de classe.

5.2 `Onto_object_property_domain`

Associe chaque propriété d'objet à ses classes de domaine possibles.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version `OntoRel`
- `class_iri` (PK, FK vers `OntoRel_entity`) : IRI de l'entité de domaine
- `property_iri` (PK, FK vers `Onto_object_property`) : IRI de la propriété

Rôle. Définit quelles classes peuvent être sources d'une relation donnée. Cette information est utilisée pour valider la cohérence des chemins de navigation.

5.3 `Onto_object_property_range`

Spécifie les classes de portée pour chaque propriété d'objet.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version `OntoRel`
- `class_iri` (PK, FK vers `OntoRel_entity`) : IRI de l'entité de portée
- `property_iri` (PK, FK vers `Onto_object_property`) : IRI de la propriété

Rôle. Détermine les classes cibles possibles pour une relation. Conjointement avec le domaine, cette table permet de valider les chemins de navigation et d'optimiser les requêtes de parcours de graphe.

5.4 `Onto_object_property_inheritance`

Maintient la hiérarchie des propriétés d'objet via les relations de spécialisation.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `superproperty_iri` (PK, FK vers `Onto_object_property`) : Propriété parente
- `subproperty_iri` (PK, FK vers `Onto_object_property`) : Propriété enfant

Rôle. Capture les relations de subsomption entre propriétés d'objet, permettant l'inférence de relations implicites lors de l'exploration topologique.

6 Tables des propriétés de données

Les tables de propriétés de données gèrent les attributs littéraux des instances de classes.

6.1 `Onto_data_property`

Répertorie les propriétés de données utilisées pour associer des valeurs littérales aux instances.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `OntoRel_entity`) : IRI de la propriété de données

Rôle. Table de base pour toutes les propriétés de données du modèle. Contrairement aux propriétés d'objet, les propriétés de données ne possèdent pas de `table_id` car elles sont généralement représentées comme colonnes dans les tables de classes.

6.2 `Onto_data_property_domain`

Définit les classes autorisées comme domaine pour chaque propriété de données.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `class_iri` (PK, FK vers `OntoRel_entity`) : IRI de l'entité de domaine
- `property_iri` (PK, FK vers `Onto_data_property`) : IRI de la propriété

Rôle. Spécifie quelles classes peuvent porter une propriété de données donnée, assurant la cohérence sémantique des attributs.

6.3 `Onto_data_property_range`

Spécifie les types de données permis comme portée pour chaque propriété de données.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `datatype_iri` (PK, FK vers `Onto_data_type`) : IRI du type de données
- `property_iri` (PK, FK vers `Onto_data_property`) : IRI de la propriété

Rôle. Définit les types de valeurs littérales acceptables pour chaque propriété de données, garantissant la cohérence des types lors de la génération SQL.

6.4 Onto_data_axiom

Stocke les axiomes spécifiques aux propriétés de données, incluant les contraintes.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `domain_iri` (PK, FK vers `Onto_class`) : Classe de domaine
- `range_iri` (PK, FK vers `Onto_data_type`) : Type de données de portée
- `property_iri` (PK, FK vers `Onto_data_property`) : Propriété concernée
- `table_id` (PK) : Identifiant de table associé
- `domain_card` : Cardinalité du domaine
- `origin` : Source de l'axiome

Rôle. Définit les contraintes sémantiques sur les propriétés de données, particulièrement les cardinalités qui influencent la génération des contraintes SQL.

7 Tables des types de données

Les tables de types de données assurent le mapping entre types ontologiques abstraits et types SQL concrets.

7.1 Onto_data_type

Catalogue les types de données ontologiques avec leur représentation SQL.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `OntoRel_entity`) : IRI du type de données
- `table_id` : Identifiant de table (pour types complexes)
- `owlsql_type` : Type SQL correspondant de haut niveau

Rôle. Établit la correspondance entre types de données abstraits (`xsd:string`, `xsd:int`, etc.) et leurs représentations SQL génériques. Le champ `owlsql_type` est utilisé lors de la génération de schéma SQL pour déterminer les types de colonnes appropriés.

7.2 Onto_data_type_sql

Détaille les mappings de types vers différents systèmes de bases de données.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `owlsql_type` (PK) : Type générique OWL-SQL
- `postgresql_type` (PK) : Type PostgreSQL spécifique
- `iri` : IRI du type de données

Rôle. Permet l'adaptation du système à différents SGBD en maintenant des mappings spécifiques. Actuellement centré sur PostgreSQL, cette table peut être étendue pour supporter d'autres systèmes (MySQL, Oracle, etc.).

8 Tables linguistiques et métadonnées

Les tables linguistiques supportent le multilinguisme du catalogue, permettant des interfaces utilisateur adaptées au contexte linguistique.

8.1 Languages_codes

Maintient les codes ISO 639-1 des langues supportées par le système.

Structure.

- `code` (PK) : Code ISO 639-1 à deux lettres (ex : `en`, `fr`)
- `language` : Nom complet de la langue

Rôle. Sert de référence pour toutes les métadonnées multilingues du catalogue. Le système supporte actuellement plus de 180 langues selon la norme ISO 639-1.

8.2 Label

Stocke les libellés multilingues pour toutes les entités OntoRel.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `OntoRel_entity`) : IRI de l'entité
- `code` (PK, FK vers `Languages_codes`) : Code de langue
- `value` : Libellé dans la langue spécifiée

Rôle. Fournit des libellés lisibles par l'humain pour toutes les entités ontologiques, facilitant l'interface utilisateur et la documentation. Ces libellés sont récupérés lors de l'annotation sémantique.

8.3 Definition

Contient les définitions textuelles des entités dans différentes langues.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `OntoRel_entity`) : IRI de l'entité
- `code` (PK, FK vers `Languages_codes`) : Code de langue
- `value` : Définition dans la langue spécifiée

Rôle. Enrichit la documentation automatique du système en fournissant des descriptions détaillées des entités ontologiques. Ces définitions sont également utilisées pour améliorer l'expérience utilisateur lors de la construction de requêtes ORQL.

9 Tables de gestion des ontologies

Les tables de gestion permettentnt l'intégration de multiples ontologies sources dans le catalogue.

9.1 Ontology

Stocke les métadonnées des ontologies intégrées dans OntoRelCat.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK) : IRI de l'ontologie
- `file_name` : Nom du fichier source
- `alias` : Alias court pour référence
- `version` : Numéro de version de l'ontologie
- `create_date` : Date de création avec fuseau horaire

Rôle. Table centrale pour la gestion des sources ontologiques. Permet la traçabilité des entités vers leur ontologie d'origine et supporte l'intégration de multiples ontologies au sein d'un même catalogue OntoRelCat.

9.2 Ontology_label

Associe des libellés multilingues aux ontologies.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `iri` (PK, FK vers `Ontology`) : IRI de l'ontologie
- `code` (PK, FK vers `Languages_codes`) : Code de langue
- `value` : Libellé de l'ontologie

Rôle. Facilite l'identification et la documentation des ontologies sources dans différentes langues.

9.3 OntoRel_schema_definition

Fournit des définitions textuelles multilingues pour les schémas.

Structure.

- `ontorel_uuid` (PK, FK) : Référence vers la version OntoRel
- `schema_name` (PK, FK vers `OntoRel_schema`) : Nom du schéma
- `code` (PK, FK vers `Languages_codes`) : Code de langue
- `value` : Définition du schéma

Rôle. Améliore la documentation du système en fournissant des descriptions détaillées des schémas organisationnels.

10 Configuration système

La table de configuration centralise les paramètres de génération du modèle relationnel.

10.1 Ontorel_db_config

Centralise les configurations de base de données pour les entités OntoRel.

Structure.

- `ontorel_uuid` (PK, FK) : Identifiant unique de configuration
- `key_name` : Nom des clés primaires générées
- `key_domain` : Domaine des clés primaires
- `key_sql_type` : Type SQL des clés primaires
- `attribute_name` : Préfixe pour noms d'attributs
- `attribute_domain` : Domaine des attributs
- `attribute_sql_type` : Type SQL par défaut des attributs
- `max_identifier_length` : Longueur maximale des identifiants SQL
- `use_iri_as_table_id` : Utiliser les IRI comme identifiants de table
- `normalize_datatype` : Normaliser les types de données
- `normalize_axiom` : Normaliser les axiomes
- `remove_thing_table` : Supprimer la table Thing générique
- `generate_optable` : Générer les tables de propriétés d'objet

Rôle. Contrôle tous les aspects de la génération du schéma relationnel OntoRel à partir des ontologies sources.

11 Procédures stockées

Le catalogue OntoRelCat fournit un ensemble de procédures stockées pour faciliter l'insertion et la manipulation des données. Ces procédures garantissent l'intégrité référentielle et simplifient l'interface d'accès au catalogue.

11.1 Procédures principales

- `ontorel_ins` : Insertion d'une nouvelle version OntoRel
- `onto_config_db_ins` : Configuration de base de données
- `onto_schema_ins` : Définition de schémas organisationnels
- `ontology_ins` : Enregistrement d'ontologies sources
- `onto_class_ins` : Insertion de classes ontologiques
- `onto_object_properties_ins` : Enregistrement de propriétés d'objet
- `onto_data_properties_ins` : Enregistrement de propriétés de données
- `onto_label_ins` : Ajout de libellés multilingues
- `onto_definition_ins` : Ajout de définitions

11.2 Fonctions utilitaires

- `getShortIRI` : Extraction du nom court d'un IRI complet
- `delete_ontorel` : Suppression complète d'une version OntoRel

12 Vues du catalogue

Le catalogue expose deux vues principales pour faciliter la consultation des définitions de tables :

- `Table_Definition_EN` : Définitions en anglais des classes, axiomes et types de données
- `Table_Definition_FR` : Définitions en français des classes, axiomes et types de données

Ces vues agrègent les informations des différentes tables pour fournir une vue consolidée et lisible de la structure du modèle OntoRel.

13 Conclusion

Le catalogue OntoRelCat constitue le cœur métier du système OntoRelQuery. Sa conception modulaire et extensible permet d'intégrer de multiples ontologies, de supporter le multilinguisme, et de s'adapter à différents systèmes de gestion de bases de données.