**Университет ИТМО**

Кафедра ИПМ

Интеграция данных в информационных системах

**Лабораторная работа №1**

«Доступ к разнородным БД на основе онтологий»

Выполнили студенты группы P4217:

Пьянков Е.В.

Данилов М.В.

Преподаватель:Колчин М.А.

Санкт-Петербург

2018

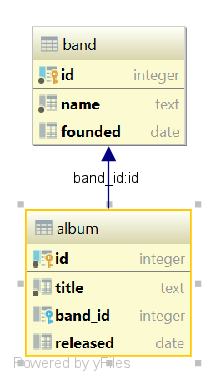
# **Задание**:

1. Выбрать предметную область. Примером предметной области могут быть базы данных 2-х библиотек с информацией о книгах и авторах, к которым нужно обеспечить доступ через единый интерфейс запросов.
2. Развернуть 2 реалиционные БД и наполнить их данными в соответствии с требованиями:
   1. Каждая из БД должна иметь отличную от другой схему данных, т.е. разные наименования таблиц и полей, а так же разная структура таблиц,
   2. Данные в обоих БД не должны полностью дублироваться.
3. Разработать онтологию для предметной области. При разработке онтологии необходимо определить компетентностные вопросы.
4. Развернуть Ontop и подключить его к БД.
5. Настроить маппинги онтологии на схемы данных БД с помощью соответствующего плагина к Protege.
6. Подговить SPARQL запросы к Ontop, в соответствии с определенными ранее компетентностными вопросами.

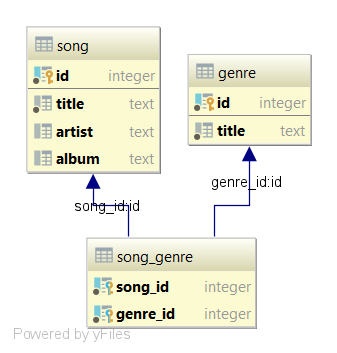
# **Ход работы:**

Предметная область представлена двумя базами данных, хранящими информацию о музыкальных группах, их альбомах и композициях.

БД 1:



БД 2:



С помощью protégé были разработаны онтологии и их маппинг на реляционную структуру с помощью ontop.

**Пример маппинга:**

|  |
| --- |
| [MappingDeclaration] @collection [[ |
|  | mappingId urn:MAPID-cd099d3289a24721a72f0804e5e92385 |
|  | target :band/{id} a :Band . |
|  | source select id from band; |
|  |  |
|  | mappingId urn:MAPID-0a1bb585989d484192e14b374e0e8316 |
|  | target :band/{id} :name {name} . |
|  | source select id, name from band |
|  |  |
|  | mappingId urn:MAPID-2137d9c17776476bb9683be6264ca018 |
|  | target :band/{id} :founded {founded} . |
|  | source select id, founded from band |
|  |  |
|  | mappingId urn:MAPID-d2377942c8b34825b1b2306360ac8373 |
|  | target :album/{id} a :Album . |
|  | source select id from album |

**Примеры запросов:**

* Найти все группы и их песни

PREFIX db1: <http://www.my.org/db1#>

PREFIX db2: <http://www.my.org/db2#>

SELECT ?band\_name ?song\_title

WHERE

{

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db1> {

?band a db1:Band; db1:name ?band\_name .

}

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db2> {

?song a db2:Song; db2:band ?band\_name; db2:title ?song\_title .

}

}

* Найти все песни, выпущенные после 2004

PREFIX db1: <http://www.my.org/db1#>

PREFIX db2: <http://www.my.org/db2#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?song\_title ?released

WHERE

{

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db1> {

?band a db1:Album;

db1:title ?album\_title;

db1:released ?released.

filter (?released > '2004-01-01T00:00:00.0'^^xsd:dateTime)

}

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db2> {

?song a db2:Song;

db2:album ?album\_title;

db2:title ?song\_title .

}

}

* Найти группы, исполняющие nu metal

PREFIX db1: <http://www.my.org/db1#>

PREFIX db2: <http://www.my.org/db2#>

SELECT distinct ?band ?band\_name

WHERE

{

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db1> {

?band a db1:Band;

db1:name ?band\_name;

}

SERVICE <http://localhost:8080/openrdf-sesame/repositories/db2> {

?song a db2:Song;

db2:band ?band\_name;

db2:genre ?genre.

?genre db2:title ?genre\_title.

filter (?genre\_title = "nu metal")

}

}