**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**Факультет программной инженерии и компьютерной техники Кафедра вычислительной техники**

**КУРСОВАЯ РАБОТА «Использование NoSQL (not only SQL) баз данных» ПО ДИСЦИПЛИНЕ «СИСТЕМЫ БАЗ ДАННЫХ»**

Выполнили: Плюхин Дмитрий Алексеевич Останина Анастасия Александровна Группа: P3317

Преподаватель: Беликов Павел Андреевич

\

Санкт-Петербург

2018

Постановка задачи

Предметная область – Очень Большой Космический корабль.

Информация, хранимая в системе – данные, предназначенные для описания корабля, отправляющегося на некоторую планету не-Марс.

|  |  |
| --- | --- |
| Тип базы данных | Содержимое базы данных |
|  |  |
| Документо-ориентированная | Описание корабля, профили команды |
|  |  |
| Колоночная | История операций, бортовой журнал |
|  |  |
| Графовая | Описание взаимоотношений между |
|  | участниками экспедиции |

Описание разработанной системы

TABLE OF CONTENTS

* Home.md
* 1.-Prerequisites.md
* 2.-Entities.md
* 3.-Running-databases.md
* 4.-Filling-databases.md
* 5.-API.md
* 6.-Replication-and-sharding.md
* 7.-Ilm.md



**Home.md**

**The Document**

The document describes the main points needed for base understanding of the project. It is outlined, what is necessary for successful running the system as well as how the project is organised internally.

**The system**

Information system, containing data about activity of hypotetical spaceship, which is going to the planet not-Mars.

[Repository on github (https://github.com/ostaninanastya/space\_ship)](https://github.com/ostaninanastya/space_ship)

**Project structure**

/api contains mappers and manipulators for making it possible to interact with system via [GraphQL (http://graphql.org/learn/)](http://graphql.org/learn/) as well as web-server, written on [node.js (https://nodejs.org/en/)](https://nodejs.org/en/) in order to simplify testing

[/clustering (versions, which support clustering are located on the branches **clustering** and **ilm**) contains config files used for setting up mongodb](https://www.mongodb.com/) [(https://www.mongodb.com/),](https://www.mongodb.com/) [cassandra (http://cassandra.apache.org/)](http://cassandra.apache.org/) [and](https://www.mongodb.com/) [neo4j (https://neo4j.com/)](https://neo4j.com/) [clusters](https://www.mongodb.com/)

/connectors contains sets of functions for setting up connection to the databases using vendor's libraries

[/generation contatins data generators written in java for the databases; all of the generators use dummymaker](https://github.com/GoodforGod/dummymaker) [(https://github.com/GoodforGod/dummymaker) library](https://github.com/GoodforGod/dummymaker)

/logbook contains code which is relevant for collections from the 'logbook' section, which represent journals

/recital contains code which is relevant for collections from the 'recital' section, which represent basic info about the ship and it's command

/relations contains code which is relevant for collections from the 'relations' section, which describes relations between people on the ship and introduces few extra entities

[/store (on the branch **ilm**) contains code which is relevant for the base data container in the ilm architecture (see wiki](https://github.com/ostaninanastya/space_ship/wiki) [(https://github.com/ostaninanastya/space\_ship/wiki) to find out more about that) - for these purposes](https://github.com/ostaninanastya/space_ship/wiki) [MySQL (https://www.mysql.com/)](https://www.mysql.com/) was being used

/transporters (on the branch **ilm**) contains modules, dedicated for exchanging data between databases in the ilm architecture

**1.-Prerequisites.md**

Прежде всего, для нормальной работы всех скриптов необходимо задать в переменной окружения **SPACE\_SHIP\_HOME** путь к папке приложения.

**mongo server**

Для установки на Ubuntu:

sudo apt-key adv *--keyserver hkp://keyserver.ubuntu.com:80 --recv 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5*

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/ap sudo apt-**get** update

sudo apt-**get** install -y mongodb-org



**cassandra server**

Для установки из debian packages (**изменить версию с** **311x** **на актуальную**)

**echo** "deb http://www.apache.org/dist/cassandra/debian 311x main" | sudo tee -**a** /etc/apt/sources.**list**.**d**/cassandra.sourcescurl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key **add** - sudo apt-**get update**

sudo apt-key adv --keyserver pool.sks-keyservers.net --recv-key A278B781FE4B2BDA sudo apt-**get** install cassandra



**neo4j server**

Установка производится так:

wget --**no**-check-certificate -O - https://debian.neo4j.org/neotechnology.gpg.key | sudo apt-key **add** - **echo** 'deb http://debian.neo4j.org/repo stable/' | sudo tee /etc/apt/sources.**list**.**d**/neo4j.**list** sudo apt **update**

sudo apt install neo4j

**cassandra-driver**

Модуль для python, с помощью которого идет взаимодействие с БД

Устанавливается командой (может занять несколько минут)

pip **install** cassandra-driver

**pymongo**

Модуль для python, с помощью которого идет взаимодействие с базой данных, содержащей информацию о корабле (работающей на технологии Mongo DB) для обеспечения консистентности хранимых данных при их изменении

Устанавливается командой

pip **install** pymongo

**py2neo**

Модуль для python, с помощью которого идет взаимодействие с базой данных, содержащей информацию об организации деятельности экипажа (работающей на технологии neo4j) для обеспечения консистентности хранимых данных при их изменении

Устанавливается командой

pip **install** py2neo

**neomodel**

ODM для python, с помощью которого идет взаимодействие с neo4j

Устанавливается командой

pip **install** neomodel

**maven**

Система сборки для создания jar-архива с пользовательскими функциями для neo4j

Устанавливается например так

sudo apt-**get** install maven

**node js**

Платформа для запуска веб-сервера, с помощью которого можно можно пользоваться api из браузера

**graphene**

Библиотека, упрощающая реализацию api на GraphQL

pip3 **install** graphene

**2.-Entities.md**

**Recital**

Информация о составе корабля и его экипажа

Поскольку каждому экземпляру сущности автоматически присваивается object id, и именно он используется в дальнейшем для ссылки на экземпляры сущностей в том числе и из других баз данных, далее в описании сущностей он указываться не будет.

department - сущность для представления департамента в информационной системе



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **property** |  |  | **data type** | | **description** |  |
|  |  |  |  |  |  |  |
| name |  |  | string |  | Название департамента |  |
|  |  |  |  |  |  |  |
| hrefToVkCommunity | |  | string |  | Ссылка на сообщество вконтакте |  |
|  |  |  | |  |  |  |
| people - сущность для представления члена экипажа корабля | | | | | | |
|  |  | | |  | | |
| **property** | **data type** | | | **description** | | |
|  |  | |  |  | | |
| name | string | |  | Имя члена экипажа | | |
|  |  | |  |  | | |
| surname | string | |  | Фамилия члена экипажа | | |
|  |  | |  |  | | |
| patronymic | string | |  | Отчество члена экипажа | | |
|  |  | | |  | | |
| department | object id | | | Ссылка на департамент в котором числится член экипажа | | |
|  |  | |  |  | | |
| phoneNumber | string | |  | Номер телефона | | |
|  |  | | |  | | |
| specialization | object id | | | Ссылка на специализацию, присвоенную члену экипажа | | |
|  |  |  |  |  |  |  |



properties - материальное имущество корабля (**технические средства**, **мебель**, **средства вооружения** и т.д.)



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **property** | |  |  | **data** | | |  |  | **description** | | |  |
|  |  | **type** | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| name | |  |  | string | | |  |  | Название предмета | | |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |
| type | |  |  | object | | | |  | Ссылка на тип, к которому принадлежит элемент имущества | | |  |
|  |  | id | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| dateOfAdmission | | |  | date | | |  |  | Дата поступления на корабль, внесения в списки имущества и, соответственно, введения в эксплуатацию | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| comissioningDate | | |  | date | | |  |  | Дата прекращения эксплуатации элемента имущества, исключения из списков корабля, и, в некоторых | | |  |
|  |  |  | случаях, выброса во внешнюю среду | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |
| department | |  |  | object | | | |  | Ссылка на департамент в котором числится предмет | | |  |
|  |  | id | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | | |  |  |  |  |
| propertyTypes - типы материального имущества корабля | | | | | | | | | | | |  |
|  |  |  | | |  |  |  |  | | |  |  |
| **property** | | **data type** | | | |  |  | **description** | |  |  |  |
|  |  |  | |  |  |  |  |  | |  |  |  |
| name | | string | |  |  |  |  | Название типа имущества (**laptop**, **table**, **laser gun**, **big fragging gun**, etc) | |  |  |  |
|  |  |  | |  |  |  |  |  | |  |  |  |
| description | | string | |  |  |  |  | Описание типа имущества | |  |  |  |
|  |  |  | | |  |  | | | |  |  |  |
| specializations - специальности членов экипажа | | | | | | | | | | | |  |
|  |  |  | | |  | |  | | | | |  |
| **property** |  | **data type** | | |  |  | **description** | | | |  |  |
|  |  |  | |  |  |  |  | | | |  |  |
| name |  | string | |  |  |  | Название специальности (**electrician**, **doctor**, **first engineer**, **second engineer**, etc) | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



states - возможные состояния систем корабля



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **property** | |  | **data type** | | | |  | **description** | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| name | |  | string | |  |  |  | Название состояния (**ready**, **repair**, **working**, etc) | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| description | |  | string | |  |  |  | Описание состояния | |  |  |  |  |
|  |  |  | |  | |  |  | |  |  |  |  |  |
| systems - системы, имеющиеся на корабле | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
| **property** | | |  |  | **data** | | |  | **description** | | | |  |
|  |  | **type** | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | |  |  |  |
| name | | |  |  | string | | | | Название системы (**fuel system**, **main engine**, **spare engine**, **main thrusters**, **spare thrusters**, **power** | | | |  |
|  |  | **system**, etc) | | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | |  |  |  |  |  |
| type | | |  |  | object | | | | Тип системы корабля | | | |  |
|  |  | id | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  | | |  | |  |  |  |
| serialNumber | | |  |  | double | | | | Серийный номер, присвоенный заводом-изготовителем | | | |  |
|  |  | | |  |  |  | |  |  | |  |  |  |
| dateOfLaunch | | | |  | date | | |  | Дата запуска системы | | | |  |
|  |  | | |  |  |  | |  |  | |  |  |  |
| dateOfLastChecking | | | |  | date | | |  | Дата последней проверки системы | | | |  |
|  |  |  |  |  |  |  | | |  | |  |  |  |
| personInCharge | | | |  | object | | | | Ссылка на сущность, представляюшую человека, ответственного за обеспечение нормального режима | | | |  |
|  | id | | |  | работы системы | | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | |  |  |  |  |  |
| state | | |  |  | object | | | | Ссылка на состояние, присвоенное системе | | | |  |
|  |  | id | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | | |  | |  | | |  | |  |  |  |
| systemTypes - типы систем, имеющихся на корабле | | | | | | | | | | | | |  |
|  |  | |  | | |  | |  | | | |  |  |
| **property** | |  | **data type** | | | |  | **description** | | |  |  |  |
|  |  |  |  | |  |  |  |  | | |  |  |  |
| name | |  | string | |  |  |  | Название типа систем (**fuel**, **engine**, **thrusters**, **hydraulic**, etc) | | |  |  |  |
|  |  |  |  | |  |  |  |  | | |  |  |  |
| description | |  | string | |  |  |  | Описание типа системы корабля | | |  |  |  |
|  |  |  | | | |  |  | | | |  |  |  |
| sensors - источники данных об окружающей обстановке | | | | | | | | | | | | |  |
|  |  | |  | |  |  |  | | | | | |  |
| **property** |  |  | **data type** | |  |  | **description** | | | | |  |  |
|  |  | |  | |  |  |  | | | | |  |  |
| name |  |  | string | |  |  | Название сенсора (**MINAS\_MORGUL\_T400**, **STADDLE\_N23**, **VALMAR\_17**, etc) | | | | |  |  |
|  |  | |  | |  |  |  | | | | |  |  |
| location |  |  | object id | |  |  | Ссылка на объект, представляюший расположение сенсора | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |



boats - информационные модели судов меньшего размера, хранящихся на корабле и предназначенных для следования к местам совершения операций во внешней среде без изменения курса корабля



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **property** | **data type** | | **description** |  |  |
|  |  |  |  |  |  |
| name | string | | Название судна |  |  |
|  |  |  |  |  |  |
| capacity | integer | | Количество человек, которое может принять судно |  |  |
|  |  |  |  |  |  |
| locations - возможные расположения сенсоров корабля | | | | |  |
|  |  |  |  |  |  |
| **property** | **data** |  | **description** | |  |
| **type** |  |  |
|  |  |  |  |  |
|  |  |  |  | |  |
| name | string |  | Название места расположения, которое должно давать представление о том, где находится сенсор (**top edge**, | |  |
|  | **bottom edge**, **left side**, etc) | |  |
|  |  |  |  |
|  |  |  |  |  |  |



**Logbook**

Журналы с данными о работе корабля

system\_test - предназначена для записи результатов тестирования систем корабля



|  |  |  |
| --- | --- | --- |
| **property** | **data type** | **description** |
|  |  |  |
| date | date | Дата, для которой актуально приведенное состояние системы |
|  |  |  |
| time | time | Временная отметка, для которой актуально приведенное состояние системы |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **property** | **data type** | **description** |
|  |  |  |
| system id | 12 bytes | Идентификатор системы, состояние которой было зафиксировано |
|  |  |  |
| result | int | Результат теста, может принимать значения от 0 до 100 |
|  |  |  |

control\_action - описывает команду, когда-либо выданную системе управления кораблем (например, повернуть налево или снять показания датчиков космического излучения по левому борту)



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **property** | |  | **data** | |  | **description** | | | | | |  |
|  | **type** | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |
| date | |  | date | |  | Дата, определяющая момент получения команды | | | | | |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |
| time | |  | time | |  | Временная отметка, определяющая момент получения команды | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| mac\_address | | 6 | |  |  | MAC-адрес устройства, выдавшего команду | | | | | |  |
|  | bytes | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| user\_id | | 12 | |  |  | Идентификатор пользователя, выдавшего команду | | | | | |  |
|  | bytes | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |
| command | |  | text | |  | Выданная команда (например, **get**) | | | | | |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |
| params | |  | text | |  | Параметры, уточняющие, к какому результату должна привести команда (например, **--sensor=MINAS\_MORGUL** | | | | | |  |
|  |  | **T400 --value\_name=cold\_dark\_matter\_concentration**) | | | | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |
| result | |  | text | |  | Результат работы команды (например, **23.235715395881783 TeV**) | | | | | |  |
|  |  |  |  |  |  | |  | |  |  |  |  |
| position - содержит историю перемещения корабля в космическом пространстве | | | | | | | | | | | |  |
|  |  |  |  |  | | |  |  |  |  |  |  |
| **property** | |  | **data type** | | | | |  | **description** | | |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| date | |  | date | | |  |  |  | Дата, определяющая момент получения команды | | |  |
|  |  |  |  |  | | |  |  |  |  |  |  |
| time | |  | timestamp | | | | |  | Временная отметка, определяющая момент получения команды | | |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| x | |  | double | | |  |  |  | Координата корабля по оси OX, которая направлена от Земли в сторону планеты не-Марс | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| y | |  | double | | |  |  |  | Координата корабля по оси OY, которая перпендикулярна оси OX и направлена в сторону созвездия | | |  |
|  |  |  |  | Кассиопея | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| z | |  | double | | |  |  |  | Координата корабля по оси OZ | | |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| speed | |  | double | | |  |  |  | Линейная скорость корабля | | |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| attack\_angle | |  | double | | |  |  |  | Угол между продольной осью корабля и осью OX | | |  |
|  |  |  |  |  | |  |  |  |  |  |  |  |
| direction\_angle | |  | double | | |  |  |  | Угол между проекцией продольной осью корабля на плоскость XOY и осью OX | | |  |
|  |  | |  |  | | |  | |  |  |  |  |
| sensors\_data - содержит данные о состоянии внешней среды, поставляемые сенсорами, расположенными по периметру корабля | | | | | | | | | | | |  |
|  |  |  | |  | |  |  |  | |  | |  |
| **property** |  | **data type** | | | |  |  | **description** | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| date |  | date | | | |  |  | Дата, определяющая момент получения данных от сенсора | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| time |  | time | | | |  |  | Временная отметка, определяющая момент получения данных от сенсора | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| source\_id |  | 12 bytes | | | |  |  | Идентификатор сенсора, предоставившего данные | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| event |  | text | | | |  |  | Наименование события, ставшего причиной фиксирования данных (например, **request**) | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| value\_name |  | text | | | |  |  | Название измеренной величины (например, **space\_radiation**) | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| value |  | double | | | |  |  | Значение измеренной величины | | |  |  |
|  |  |  | |  | |  |  |  | |  |  |  |
| units |  | text | | | |  |  | Единицы измерения (например, **eV** - сокрашение от электрон-вольт | | |  |  |
|  |  | | |  | | |  | | |  |  |  |
| shift\_state - хранит обстановку в области, доверенной некоторой смене, на данный момент времени | | | | | | | | | | | |  |
|  |  |  |  |  | | | |  |  |  |  |  |
| **property** | |  |  | **data** | | | |  |  | **description** | |  |
|  |  | **type** | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | |  |  |  | | | |  |  |  | |  |
| date | |  |  | date | | | |  |  | Дата, для которой характерны данные о смене | |  |
|  | |  |  |  | | | |  |  |  | |  |
| time | |  |  | time | | | |  |  | Временная отметка, для которой характерны данные о смене | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **property** | **data** | | **description** |  |
| **type** | |  |
|  |  |  |
|  |  |  |  |  |
| shift\_id | 16 bytes | | Идентификатор смены, для которой характерно зафикисированное состояние на некоторый момент |  |
| времени |  |
|  |  |  |  |
|  |  |  |  |  |
| warning\_level | text | | Уровень опасности в районе, доверенном смене (может принимать значения **lowest**, **low**, **medium**, |  |
| **heigh**, **highest**) |  |
|  |  |  |  |
|  |  |  |  |  |
| remaining\_cartridges | double | | Оставшиеся патроны (в процентах) |  |
|  |  |  |  |  |
| remaining\_air | double | | Оставшийся воздух (в процентах) |  |
|  |  |  |  |  |
| remaining\_electricity | double | | Оставшееся электричество (в процентах) |  |
|  |  |  |  |  |
| comment | text | | Необязательный комментарий от работников смены |  |
|  |  |  |  |  |
| operation\_state - содержит описание состояния операции на некоторый момент времени | | | |  |
|  |  |  |  |  |
| **property** | **data** |  | **description** |  |
| **type** |  |  |
|  |  |  |  |
|  |  |  |  |  |
| date | date |  | Дата, для которой характерны данные об операции |  |
|  |  |  |  |  |
| time | time |  | Временная отметка, для которой характерны данные об операции |  |
|  |  |  |  |  |
| boat\_id | 12 |  | Необязательный идентификатор машины, зарезервированной для выполнения операции |  |
| bytes |  |  |
|  |  |  |  |
|  |  |  |  |  |
| operation\_id | 16 |  | Идентификатор операции |  |
| bytes |  |  |
|  |  |  |  |
|  |  |  |  |  |
| operation\_status | text |  | Состояние операции на данный момент времени (например, **detaching\_from\_ship**) |  |
|  |  |  |  |  |
| distance\_to\_the\_ship | double |  | Расстояние от судна, предназначенного для операции, до корабля |  |
|  |  |  |  |  |
| zenith | double |  | Зенитный угол, задающий положение судна, предназначенного для операции, относительно корабля |  |
|  |  |  |  |  |
| azimuth | double |  | Азимутальный угол, задающий положение судна, предназначенного для операции, относительно |  |
|  | корабля |  |
|  |  |  |  |
|  |  |  |  |  |
| hydrogenium | double |  | Процентное содержание водорода в атмосфере / космической пыли вокруг команды, выполняющей |  |
|  | операцию |  |
|  |  |  |  |
|  |  |  |  |  |
| helium | double |  | Процентное содержание гелия в атмосфере / космической пыли вокруг команды, выполняющей |  |
|  | операцию |  |
|  |  |  |  |
|  |  |  |  |  |
| ... | double |  | 115 полей, содержащих данные о процентном содержании химических элементов в атмосфере / |  |
|  | космической пыли вокруг команды, выполняющей операцию |  |
|  |  |  |  |
|  |  |  |  |  |
| oganesson | double |  | Процентное содержание оганесона в атмосфере / космической пыли вокруг команды, выполняющей |  |
|  | операцию |  |
|  |  |  |  |
|  |  |  |  |  |
| comment | double |  | Необязательный комментарий от участников команды, выполняющей операцию |  |
|  |  |  |  |  |



**Relations**

Информация о служебных взаимоотношениях экипажа

person - представление члена экипажа в рамках графической модели данных



|  |  |  |  |
| --- | --- | --- | --- |
| **property** | **data** | **description** |  |
| **type** |  |
|  |  |  |
|  |  |  |  |
| ident | two ints 1 | Идентификатор члена экипажа, соответствующий 12 - байтному идентификатору в базе данных, содержащей |  |
| информацию о корабле (основанной на Mongo DB) |  |
|  |  |  |
|  |  |  |  |
| controlled | reference | Связь с сущностью Department, отображающая отношение главенствования над департаментом (DIRECTOR) |  |
|  |  |  |  |
| executor | reference | Связь с сущностью Operation, отображающая отношение исполнения операции (EXECUTOR) |  |
|  |  |  |  |
| headed | reference | Связь с сущностью Operation, отображающая отношение руководства операцией (HEAD) |  |
|  |  |  |  |
| worker | reference | Связь с сущностью Shift, отображающая отношение рядового участника смены (WORKER) |  |
|  |  |  |  |
| chief | reference | Связь с сущностью Shift, отображающая отношение ответственного за смену (CHIEF) |  |
|  |  |  |  |

department - представление департамента в рамках графической модели данных



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **property** |  | **data** |  | **description** |  |
|  | **type** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| ident |  | two ints 1 |  | Идентификатор департамента |  |
| shifts |  | reference |  | Связь с сущностью Shift, отображающая отношение организации смены департаментом (INCORPORATION) |  |
|  |  |  |  |  |  |
| operations |  | reference |  | Связь с сущностью Operation, отображающая отношение организации операции департаментом |  |
|  |  | (INCORPORATION) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| controller |  | reference |  | Связь с сущностью Person, отображающая отношение руководства департаментом (DIRECTOR) |  |
|  |  |  |  |  |  |
| operation - сущность, содержащая основные сведения о планируемой или уже осуществленной операции | | | | |  |
|  |  |  |  |  |  |
| **property** |  | **data** |  | **description** |  |
|  | **type** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| ident |  | 16 bytes |  | Идентификатор операции |  |
|  |  |  |  |  |  |
| name |  | text |  | Название операции |  |
|  |  |  |  |  |  |
| start |  | datetime |  | Дата и время начала операции |  |
|  |  |  |  |  |  |
| end |  | datetime |  | Дата и время окончания операции |  |
|  |  |  |  |  |  |
| department |  | reference |  | Связь с сущностью Operation, отображающая отношение организации операции департаментом |  |
|  |  | (INCORPORATION) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| requirement |  | reference |  | Связь с сущностью Requirement, отображающая отношение использования набора требований по |  |
|  |  | специализации личного состава для операции (USER) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| persons |  | reference |  | Связь с сущностью Operation, отображающая отношение исполнения операции (EXECUTOR) |  |
|  |  |  |  |  |  |
| head |  | reference |  | Связь с сущностью Operation, отображающая отношение руководства операцией (HEAD) |  |
|  | |  | |  |  |
| shift - сущность, содержащая основные сведения о прошедшей или предстоящей смене | | | | |  |
|  |  |  |  |  |  |
| **property** |  | **data** |  | **description** |  |
|  | **type** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| ident |  | 16 bytes |  | Идентификатор смены |  |
|  |  |  |  |  |  |
| start |  | datetime |  | Дата и время начала смены |  |
|  |  |  |  |  |  |
| end |  | datetime |  | Дата и время окончания смены |  |
|  |  |  |  |  |  |
| department |  | reference |  | Связь с сущностью Operation, отображающая отношение организации операции департаментом |  |
|  |  | (INCORPORATION) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| requirement |  | reference |  | Связь с сущностью Requirement, отображающая отношение использования набора требований по |  |
|  |  | специализации личного состава для смены (USER) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| persons |  | reference |  | Связь с сущностью Operation, отображающая отношение исполнения операции (EXECUTOR) |  |
|  |  |  |  |  |  |
| head |  | reference |  | Связь с сущностью Operation, отображающая отношение руководства операцией (HEAD) |  |
|  | |  | |  |  |
| requirement - набор требований к подготовке личного состава персонала для выполнения операции или образования смены | | | | |  |
|  |  |  |  |  |  |
| **property** |  | **data** |  | **description** |  |
|  | **type** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| ident |  | 16 bytes |  | Идентификатор набора |  |
|  |  |  |  |  |  |
| name |  | text |  | Название набора |  |
|  |  |  |  |  |  |
| content |  | array of |  | Массив объектов json, каждый из которых имеет по крайней мере два поля: **ident** - идентификатор |  |
|  | jsons |  | специальности из mongo db 1 и **quantity** - количество сотрудников с такой специальностью |  |
|  |  |  |  |
| operations |  | reference |  | Связь с сущностью Operation, отображающая отношение использования операцией набора требований (USER) |  |
|  |  |  |  |  |  |
| shifts |  | reference |  | Связь с сущностью Shift, отображающая отношение использования операцией набора требований (USER) |  |
|  |  |  |  |  |  |



**Note about two ints**



Значения в базе данных neo4j, соответствующие индексам mongodb хранятся в виде пар значений целочисленного типа. Для перевода такой пары обратно в 12-байтный индекс, необходимы вычислить значение

item[0] \* 2^48 + item[1]

* интерпретировать его как 12-байтный индекс Например, если имеем mongo - идентификатор

5abfdba6ee6b7f5eec83a1ca

То можем его отобразить в целочисленном представлении

28085592993066680294893134282

Или, если разбить на два числа, то получится

(99780070403691, 140045671702986)

Обратное преобразование выглядит как

(int(math.pow(2, 48)) \* 99780070403691 + 140045671702986).to\_bytes(12, byteorder='big').hex()

Описанный механизм работает подобно механизму преобразования даты и времени из используемого ODM в число с плавающей точкой для сохранения в базе данных.

Преобразование может быть сделано при помощи пользовательской функции space\_ship.get\_hex\_ident - для этого необходимо скопировать файл neo4j\_functions-1.0.0.jar из папки relations/functions/target в папку plugins, соответствующую серверу neo4j (например, /var/lib/neo4j/plugins ). Далее функцию можно вызывать в предложениях по выборке данных, например, так:

match (n:Person) return space\_ship.get\_hex\_ident(n.ident);

Перекомпиляция исходного кода и запуск тестов делаются с помощью команды

mvn clean **package**

**3.-Running-databases.md**

**Mongo DB**

Запуск сервера

sudo service mongod start

**Logbook**

Запуск сервера

sudo service cassandra start

Выполнить команду, которая должна завершиться выводом информации о сервере с зеленым словом 'active'

sudo service cassandra **status**

Для проверки статуса кластера выполнить

sudo nodetool **status**

Должна вывести результат, начинающийся с сочетания **UN** (Up and Normal) в случае успеха

**Relations**

Выполнить команду

sudo service neo4j start

Веб-интерфейс станет доступен (стандартно) по адресу

http:*//localhost:7474*

**4.-Filling-databases.md**

**Mongo**

Находясь в папке /generation/dummyMarker , выполнить

**mvn** install

**mvn** exec:java

Для перекомпиляции и последующей генерации json файлов. Далее, перейдя в папку /recital , выполнить

**python** fill.**py**

Для заполнения базы данных mongo

**5.-API.md**

**Architecture overview**

[View in full size (https://lh3.googleusercontent.com/Fo9hQuOui-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[LCtnZf5IP4TDo4\_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u\_Nvd\_4xGjy3Wyvh7VlHsu-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[r597CM\_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

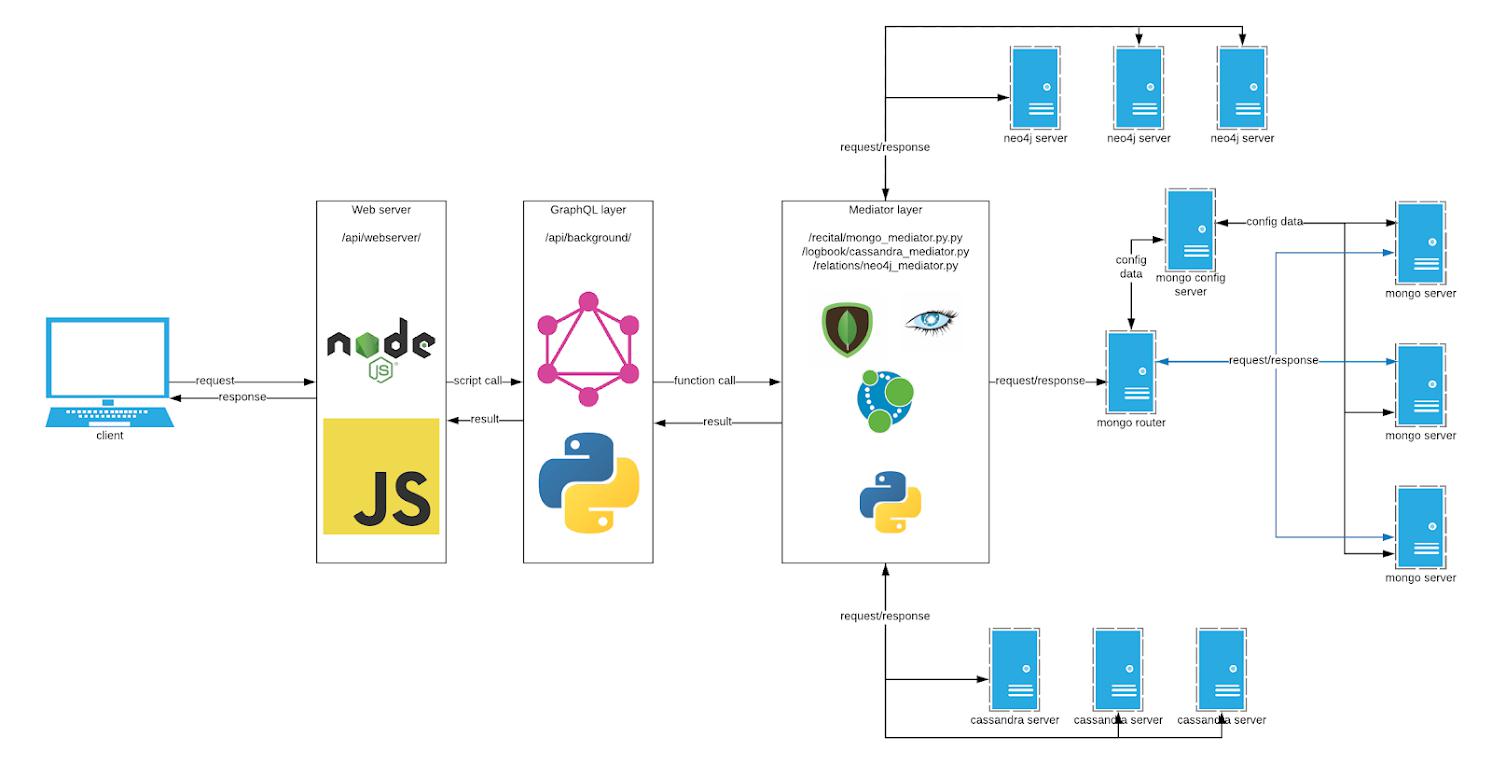
[TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[\_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[WbZbKv7p2pbHi\_uifNEf9ORo0ajzCM9Ezx\_92mFoRPX7dJ-GCrHVwNIv5\_ki8ikiVIyWs09H5cyhG1fyS6C-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)

[wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m\_bV0SV\_1VQAWLzekdOXeMVVswO=w1492-h764-no)](https://lh3.googleusercontent.com/Fo9hQuOui-LCtnZf5IP4TDo4_kyVp0arwSRpgfDoEjeyc2ptAKtnxPp2is5xZ2TKSUZpgw4ZwZPko9NxAcr4qqNLiN9R8ohaRNk8k6P0ciYDwfXR4i2u_Nvd_4xGjy3Wyvh7VlHsu-r597CM_4N40J--doOx8fQPYZdXdHlwag7jqBibXPv2y8gYXKorc7Vpm0sdMqEXLj1P1PFGxIv40iygTr-V41jKBB81OlymfR7Yawsv74YBc1-2E5F-TgaT1PHFo0jW3Qp9qTr5OxphsJ7V3kA1-qPlCKJ9jR4tYEfBgshJo9MPJSfNm-_e0kE46a9hkHy6b0puFelbk9EKNBNaojkdpZhCobxLQU34BPtgjb9R7eevIipiUOgFjSNiVd7ce0OCsmvtnEbufAEQpWDUCu8gKPtS-WbZbKv7p2pbHi_uifNEf9ORo0ajzCM9Ezx_92mFoRPX7dJ-GCrHVwNIv5_ki8ikiVIyWs09H5cyhG1fyS6C-vL9A9fAvkyW5c045W7OJegH7bDHP3b1tNVKYNV4xavjkf-A9sKwuNcwPLRlY-IJaj8L8KknzHGK-wHriU4SVfwXZ0ix18oh6lld1NTWuHtQYXqm4eZEqtO0dUWYF7x3JJ0V52D-U2IFH-Lt-m_bV0SV_1VQAWLzekdOXeMVVswO=w1492-h764-no)



**Web API**

**CRUD**



Primarily it is considered that the application is being accessed via web interface. It consists of two main parts: GET method interface and POST method interface.

**GET**

In order to use the application via GET interface the queries of the view should be used:

http://localhost:1881/api/create/person/fields=ok&where=name:'John',surname:'White' for **creation**

http://localhost:1881/api/people/fields=name,surname&where=name:'John' for **reading**

http://localhost:1881/api/people/fields=name,surname&where=name:'John'&set=specialization:5accfcc5ee18bbcfe8035add for

**updating**

http://localhost:1881/api/delete/person/fields=ok&where=name:'John' for **deleting**

**POST**

POST queries are significantly differs from the GET queries - in contrast to seconds they use **html query body** encoded in json to represent query parameters. All the queries, described below, have to be sent to the address http://localhost:1881/api/ .

for **creation**:

{

"operation": "create",

"entity": "person",

"fields" : ["ok"],

"where": {

"name": "John",

"surname": "White"

}

}

for **reading**:

{

"operation": "read",

"entity": "requirements",

"fields" : ["name", {

"content" : [{

"specialization": ["name"]

}, "quantity"]

}]

}

for **updating**:

{

"operation": "update",

"entity": "requirements",

"fields" : ["ok"],

"where": {

"name": "programmersa"

},

"set": {

"content": "5aeddb09d678f433ec9b286c:25,5aeddb19d678f4341fc29b82:45"

}

}

for **deleting**:

{

"operation": "remove",

"entity": "requirements",

"fields" : ["ok"],

"where": {

"name": "programmersa"

}

}

**ILM**



Moreover, there is a possibility of getting content from any layer of the ilm model (see [ilm (https://github.com/ostaninanastya/space\_ship/wiki/7.-Ilm)](https://github.com/ostaninanastya/space_ship/wiki/7.-Ilm) section to find out more about that). For example, to get all people entries from the hot-cache layer, just send a request with body

{

"layer": "hot-cache",

"entity": "people"

}

to the address http://localhost:1881/data/ .

**GraphQL**

Взаимодействие с базой данных производится при помощи GraphQL, для этого можно вызывать скрипт main.py из папки /api/background, в этом случае вызов скрипта будет иметь вид

python3 test.py 'mutation Mutation{ createSensor(name:"rohan",location:"5aca531ed532b43d2c7d0f34"){ sensor{name} }}'

Далее приведен общий список объектов GraphQL, соответствующих описанным ранее сущностям базы данных. Этот список формируется сервером по адресу [http://localhost:1881/docs/ (http://localhost:1881/docs/)](http://localhost:1881/docs/)

=== Queries ===

= Entities =

**position**{

date: String

time: String

1. **Float**
2. **Float**
3. **Float** speed: **Float** attackangle: **Float** directionangle: **Float**

}

controlaction{

date: String

time: String

macaddress: String

userid: String

user: PersonMapper

command: String

params: String

result: String

}

person{

id: String

**name**: String

department: DepartmentMapper

directing: List

chiefed: List

worked: List

headed: List

executed: List

specialization: SpecializationMapper

commands: List

supervised: List

}

department{

id: String

**name**: String

director: PersonMapper

vk: String

properties: List

people: List

}

property{

id: String

**name**: String

**type**: PropertyTypeMapper

admission: String

comissioning: String

department: DepartmentMapper

}

propertytype{

id: String

**name**: String

description: String

properties: List

}

shift{

id: String

start: String

**end**: String

chief: PersonMapper

workers: List

requirements: List

}

requirement{

id: String

**name**: String

content: List

shifts: List

operations: List

}

requiremententry{

specialization: SpecializationMapper

quantity: **Int**

}

specialization{

id: String

**name**: String

people: List

}

operation{

id: String

**name**: String

start: String

**end**: String

head: PersonMapper

executors: List

requirements: List

}

system{

id: String

**name**: String

serialnumber: **Float**

launched: String

checked: String

state: SystemStateMapper

supervisor: PersonMapper

**type**: SystemTypeMapper

}

systemstate{

id: String

**name**: String

description: String

systems: List

}

systemtype{

id: String

**name**: String

description: String

systems: List

}

systemtest{

date: String

time: String

system: SystemMapper

systemid: String

result: **Int**

}

operationstate{

date: String

time: String

boatid: String

boatname: String

operationid: String

operation: OperationMapper

operationstatus: String

distancetotheship: **Float**

zenith: **Float**

azimuth: **Float**

hydrogenium: **Float**

helium: **Float**

lithium: **Float**

beryllium: **Float**

borum: **Float**

carboneum: **Float**

nitrogenium: **Float**

oxygenium: **Float**

fluorum: **Float**

neon: **Float**

natrium: **Float**

magnesium: **Float**

aluminium: **Float**

silicium: **Float**

phosphorus: **Float**

sulfur: **Float**

chlorum: **Float**

argon: **Float**

kalium: **Float**

calcium: **Float**

scandium: **Float**

titanium: **Float**

vanadium: **Float**

chromium: **Float**

manganum: **Float**

ferrum: **Float**

cobaltum: **Float**

niccolum: **Float**

cuprum: **Float**

zincum: **Float**

gallium: **Float**

germanium: **Float**

arsenicum: **Float**

selenium: **Float**

bromum: **Float**

crypton: **Float**

rubidium: **Float**

strontium: **Float**

yttrium: **Float**

zirconium: **Float**

niobium: **Float**

molybdaenum: **Float**

technetium: **Float**

ruthenium: **Float**

rhodium: **Float**

palladium: **Float**

argentum: **Float**

cadmium: **Float**

indium: **Float**

stannum: **Float**

stibium: **Float**

tellurium: **Float**

iodium: **Float**

xenon: **Float**

caesium: **Float**

barium: **Float**

lanthanum: **Float**

cerium: **Float**

praseodymium: **Float**

neodymium: **Float**

promethium: **Float**

samarium: **Float**

europium: **Float**

gadolinium: **Float**

terbium: **Float**

dysprosium: **Float**

holmium: **Float**

erbium: **Float**

thulium: **Float**

ytterbium: **Float**

lutetium: **Float**

hafnium: **Float**

tantalum: **Float**

wolframium: **Float**

rhenium: **Float**

osmium: **Float**

iridium: **Float**

platinum: **Float**

aurum: **Float**

hydrargyrum: **Float**

thallium: **Float**

plumbum: **Float**

bismuthum: **Float**

polonium: **Float**

astatum: **Float**

radon: **Float**

francium: **Float**

radium: **Float**

actinium: **Float**

thorium: **Float**

protactinium: **Float**

uranium: **Float**

neptunium: **Float**

plutonium: **Float**

americium: **Float**

curium: **Float**

berkelium: **Float**

californium: **Float**

einsteinium: **Float**

fermium: **Float**

mendelevium: **Float**

nobelium: **Float**

lawrencium: **Float**

rutherfordium: **Float**

dubnium: **Float**

seaborgium: **Float**

bohrium: **Float**

hassium: **Float**

meitnerium: **Float**

darmstadtium: **Float**

roentgenium: **Float**

copernicium: **Float**

nihonium: **Float**

flerovium: **Float**

moscovium: **Float**

livermorium: **Float**

tennessium: **Float**

oganesson: **Float**

comment: String

}

shiftstate{

date: String

time: String

shiftid: String

warninglevel: String

remainingcartridges: **Int**

remainingair: **Int**

remainingelectricity: **Int**

shift: ShiftMapper

comment: String

}

sensordata{

date: String

time: String

sourceid: String

source: SensorMapper

location: String

event: String

valuename: String

**value**: **Float**

units: String

}

sensor{

id: String

**name**: String

location: LocationMapper

}

location{

id: String

**name**: String

sensors: List

}

boat{

id: String

**name**: String

capacity: **Int**

}

= Examples =

http://localhost:1881/api/shiftstate/fields=shiftid,time,shift(id,start,**end**,chief(id))&**where**=minute:42

=== Mutations ===

= Entities =

create\_location {

**name**: String

}

remove\_location {

id: String

}

eradicate\_location {

id: String

}

create\_sensor {

location: String

**name**: String

}

remove\_sensor {

id: String

}

create\_department {

**name**: String

vk: String

}

remove\_department {

id: String

}

eradicate\_department {

id: String

}

create\_specialization {

**name**: String

}

remove\_specialization {

id: String

}

eradicate\_ {

id: String

}

create\_person {

department: String

**name**: String

patronymic: String

phone: String

specialization: String

surname: String

}

remove\_person {

id: String

}

eradicate\_person {

id: String

}

create\_boat {

capacity: **Int**

**name**: String

}

remove\_boat {

id: String

}

create\_systemtype {

description: String

**name**: String

}

remove\_systemtype {

id: String

}

eradicate\_systemtype {

id: String

}

create\_systemstate {

description: String

**name**: String

}

remove\_systemstate {

id: String

}

eradicate\_systemstate {

id: String

}

create\_propertytype {

description: String

**name**: String

}

remove\_propertytype {

id: String

}

eradicate\_propertytype {

id: String

}

create\_system {

checked: String

launched: String

**name**: String

serialnumber: **Float**

state: String

supervisor: String

**type**: String

}

remove\_system {

id: String

}

create\_property {

admission: String

comissioning: String

department: String

**name**: String

**type**: String

}

remove\_property {

id: String

}

create\_shift {

chief: String

department: String

**end**: String

requirements: String

start: String

workers: String

}

remove\_shift {

id: String

}

create\_operation {

**end**: String

executors: String

head: String

**name**: String

requirements: String

start: String

}

remove\_operation {

id: String

}

create\_requirement {

content: String

**name**: String

}

remove\_requirement {

id: String

}

create\_position {

attackangle: **Float**

directionangle: **Float**

speed: **Float**

timestamp: String

1. **Float**
2. **Float**
3. **Float**

}

remove\_position {

timestamp: String

}

create\_controlaction {

command: String

mac: String

params: String

result: String

timestamp: String

user: String

}

remove\_controlaction {

timestamp: String

}

create\_systemtest {

result: **Int**

system: String

timestamp: String

}

remove\_systemtest {

timestamp: String

}

create\_sensordata {

event: String

meaning: String

source: String

timestamp: String

units: String

**value**: **Float**

}

remove\_sensordata {

timestamp: String

}

create\_shiftstate {

comment: String

remainingair: **Int**

remainingcartridges: **Int**

remainingelectricity: **Int**

shift: String

timestamp: String

warninglevel: String

}

remove\_shiftstate {

timestamp: String

}

create\_operationstate {

actinium: **Float**

aluminium: **Float**

americium: **Float**

argentum: **Float**

argon: **Float**

arsenicum: **Float**

astatum: **Float**

aurum: **Float**

azimuth: **Float**

barium: **Float**

berkelium: **Float**

beryllium: **Float**

bismuthum: **Float**

boat: String

bohrium: **Float**

borum: **Float**

bromum: **Float**

cadmium: **Float**

caesium: **Float**

calcium: **Float**

californium: **Float**

carboneum: **Float**

cerium: **Float**

chlorum: **Float**

chromium: **Float**

cobaltum: **Float**

comment: String

copernicium: **Float**

crypton: **Float**

cuprum: **Float**

curium: **Float**

darmstadtium: **Float**

distancetotheship: **Float**

dubnium: **Float**

dysprosium: **Float**

einsteinium: **Float**

erbium: **Float**

europium: **Float**

fermium: **Float**

ferrum: **Float**

flerovium: **Float**

fluorum: **Float**

francium: **Float**

gadolinium: **Float**

gallium: **Float**

germanium: **Float**

hafnium: **Float**

hassium: **Float**

helium: **Float**

holmium: **Float**

hydrargyrum: **Float**

hydrogenium: **Float**

indium: **Float**

iodium: **Float**

iridium: **Float**

kalium: **Float**

lanthanum: **Float**

lawrencium: **Float**

lithium: **Float**

livermorium: **Float**

lutetium: **Float**

magnesium: **Float**

manganum: **Float**

meitnerium: **Float**

mendelevium: **Float**

molybdaenum: **Float**

moscovium: **Float**

natrium: **Float**

neodymium: **Float**

neon: **Float**

neptunium: **Float**

niccolum: **Float**

nihonium: **Float**

niobium: **Float**

nitrogenium: **Float**

nobelium: **Float**

oganesson: **Float**

operation: String

osmium: **Float**

oxygenium: **Float**

palladium: **Float**

phosphorus: **Float**

platinum: **Float**

plumbum: **Float**

plutonium: **Float**

polonium: **Float**

praseodymium: **Float**

promethium: **Float**

protactinium: **Float**

radium: **Float**

radon: **Float**

rhenium: **Float**

rhodium: **Float**

roentgenium: **Float**

rubidium: **Float**

ruthenium: **Float**

rutherfordium: **Float**

samarium: **Float**

scandium: **Float**

seaborgium: **Float**

selenium: **Float**

silicium: **Float**

stannum: **Float**

**status**: String

stibium: **Float**

strontium: **Float**

sulfur: **Float**

tantalum: **Float**

technetium: **Float**

tellurium: **Float**

tennessium: **Float**

terbium: **Float**

thallium: **Float**

thorium: **Float**

thulium: **Float**

timestamp: String

titanium: **Float**

uranium: **Float**

vanadium: **Float**

wolframium: **Float**

xenon: **Float**

ytterbium: **Float**

yttrium: **Float**

zenith: **Float**

zincum: **Float**

zirconium: **Float**

}

remove\_operationstate {

timestamp: String

}

= Examples =

http://localhost:1881/api/create/**position**/fields=ok,**position**(x)&**where**=timestamp:'2017-02-18 23:59:57',x:10.0,y:10.2,z:10



**Logbook**

* системе предусмотрена работа с содержимым базы данных при помощи API, предоставляемого разработчиком ODM cassandra driver. Однако, в соответствии с заданием была реализована возможность работы с БД на её "родном" языке.

Так, для выборки значений из базы данных можно воспользоваться функцией **select**, содержащейся в одноименном файле в папке native.

select('position', [['x', 10], ['y', 10], ['speed'], ['time']])

* том числе есть возможность исполнения скрипта с помощью командной строки

python3 native/**select**.py **position** x=10 y=10 speed **time**

python3 **native**/**select**.py **position date**='2018-04-03' **time**='20:43:03.426852000' speed

Формат результата

[{'date': Date(17624), 'time': Time(74583426852000), 'speed': 1488.0}]

Существует аналогичный скрипт для операции **update**

Пример вызова функции

update('position', [['date', '2018-04-03'], ['time', '20:43:03.426852000']], ['speed', 200]('speed', 200))

Пример вызова скрипта

python3 native/**update**.py **position date**='2018-04-03' **time**='20:43:03.426852000' 'speed->1488'

Указанные функции и скрипты предназначены для формирования запроса на языке cql и передачи его в СБД.

**6.-Replication-and-sharding.md**

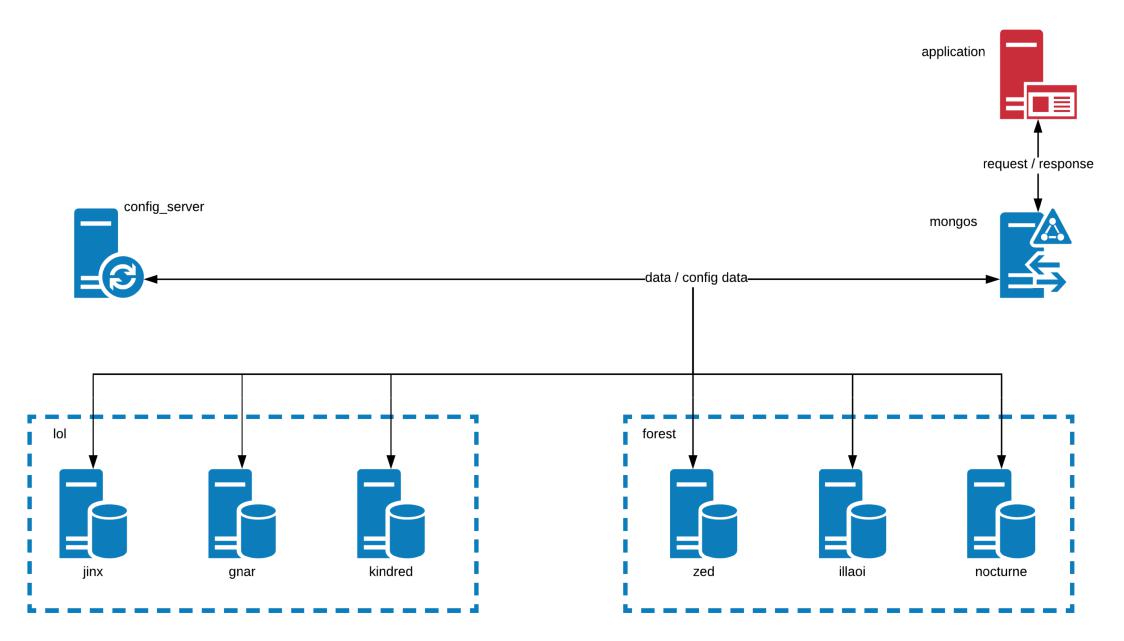
**Common**

Копии всех сконфигурированных серверов доступны [здесь (https://cloud.mail.ru/public/7U1Y/TBDcQV1Yv)](https://cloud.mail.ru/public/7U1Y/TBDcQV1Yv) (для запуска использовалась ОС **Ubuntu 17.10**,версияmongo - **3.6.4**, cassandra - **3.0.16**, neo4j - **3.3.5**).

**Mongo**

Для базы данных recital, хранящейся в mongodb, в связи с репликацией необходимо запустить 8 процессов (используются соответственно порты 1882 - 1889):

[View in full size (https://image.ibb.co/fJbOyy/mongo.png)](https://image.ibb.co/fJbOyy/mongo.png)



Три для replica set первого shard под названием 'lol' (путь к директориям, в которых хранятся файлы баз данных, указывается в соответствующих конфигурационных файлах)

mongod --config jinx\_lol\_shard\_config.yaml

mongod --config gnar\_lol\_shard\_config.yaml

mongod --config kindred\_lol\_shard\_config.yaml

Три для replica set второго shard под названием 'forest'

mongod --config zed\_forest\_shard\_config.yaml

mongod --config illaoi\_forest\_shard\_config.yaml

mongod --config nocturne\_forest\_shard\_config.yaml

Один в качестве сервера конфигурации mongos и replica sets

mongod *--config config\_server\_config.yaml*

Один в качестве драйвера (mongos)

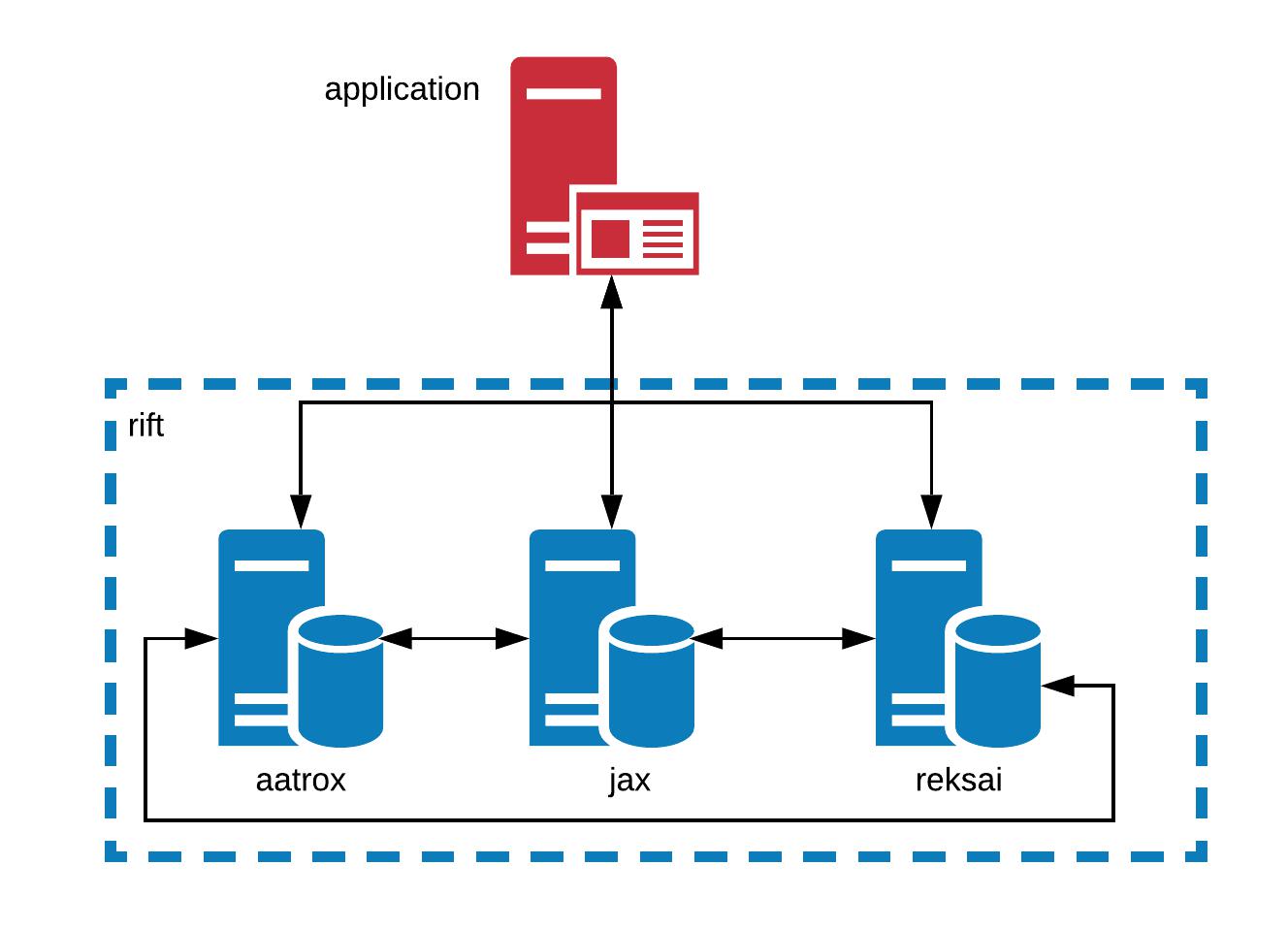
mongos *--config mongos\_config.yaml*

Необходимо также обратить внимание на необходимость инициализации всех создаваемых replica sets. Соответствующие документы, который необходимо передавать в initiate - метод, приведены в json - файлах.

**Cassandra**

[Делается как тут описано http://zcourts.com/2011/07/09/setting-up-a-multi-node-cassandra-cluster-on-a-single-windows-](http://zcourts.com/2011/07/09/setting-up-a-multi-node-cassandra-cluster-on-a-single-windows-machine/" \l "sthash.b3b1vCO7.8AmRM8mX.dpbs)[machine/#sthash.b3b1vCO7.8AmRM8mX.dpbs (http://zcourts.com/2011/07/09/setting-up-a-multi-node-cassandra-cluster-on-a-single-windows-machine/#sthash.b3b1vCO7.8AmRM8mX.dpbs) но с парой нюансов, а именно:](http://zcourts.com/2011/07/09/setting-up-a-multi-node-cassandra-cluster-on-a-single-windows-machine/#sthash.b3b1vCO7.8AmRM8mX.dpbs)

[View in full size (https://image.ibb.co/crEXrJ/cassandra.png)](https://image.ibb.co/crEXrJ/cassandra.png)



**Добавить в файл hosts ( /etc/hosts ) алиасы для localhost**

127.0.0.2 localhost

127.0.0.3 localhost

127.0.0.4 localhost

**Создать папки для нодов кассандры**

cassandra/

├── aatrox

├── jax

└── reksai

**Скачать и в каждую директорию распаковать download package**

[Например отсюда http://apache-mirror.rbc.ru/pub/apache/cassandra/3.0.16/apache-cassandra-3.0.16-bin.tar.gz (http://apache-mirror.rbc.ru/pub/apache/cassandra/3.0.16/apache-cassandra-3.0.16-bin.tar.gz)](http://apache-mirror.rbc.ru/pub/apache/cassandra/3.0.16/apache-cassandra-3.0.16-bin.tar.gz)

**Базовая конфигурация**

Задать как минимум следующие поля для каждой ноды в файле config/cassandra.yaml : cluster\_name , seed\_provider - должны быть для всех одинаковы, data\_file\_directories , commitlog\_directory , saved\_caches\_directory , listen\_address , rpc\_address - должны быть для всех разные.

**Другие важные моменты конфигурации**

1. Порт jmx меняется в файле config/cassandra-env.sh , он должен быть уникальным
2. Необходимо задать минимальный и минимальный размер памяти для jvm в файле config/jvm.options (например 100M), иначе java будет доходить до 2 гигов на каждой ноде.

Создание keyspace:

**create** keyspace **if not exists** logbook **with replication** = {'class' : 'SimpleStrategy', 'replication\_factor' : 3};

Если все сделано правильно, то выполнение describe keyspaces; , выполненное на других репликах, должно содержать в том числе и только что созданное.

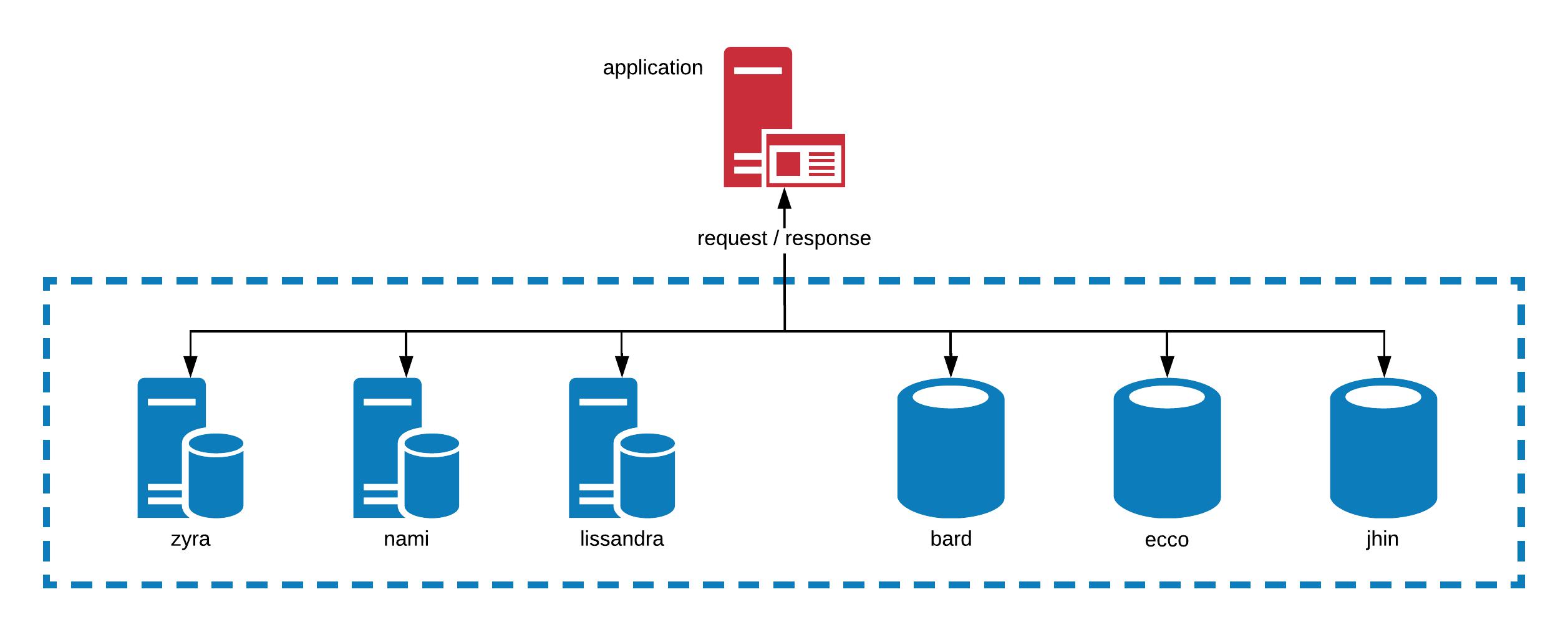
Далее можно сконфигурировать таблицы и заполнить их тестовыми данными.

**Neo4j**

**Для развертывания кластера neo4j необходима enterprise edition**

[Делается как написано здесь https://neo4j.com/docs/operations-manual/current/tutorial/local-causal-cluster/ (https://neo4j.com/docs/operations-manual/current/tutorial/local-causal-cluster/). Далее будут приведены нюансы, характерные для данного приложения.](https://neo4j.com/docs/operations-manual/current/tutorial/local-causal-cluster/)

[View in full size (https://image.ibb.co/jkgZMJ/neo4j.png)](https://image.ibb.co/jkgZMJ/neo4j.png)



**Создать папки для нодов neo4j**

neo4j/

├── **bard**

├── ekko

├── jhin

├── lissandra

├── nami

└── zyra

Будем полагать, что zyra , nami , lissandra - core nodes, а jhin , ekko , bard - replicas.

To make backup:

./neo4j-admin backup

--backup-dir=/media/zeionara/8ef5f608-07b4-4216-88ce-1c255f8edaab/zeionara/neo4j/backups --name=28.04.2018

--from=127.0.0.12:6363

--cc-report-dir=/media/zeionara/8ef5f608-07b4-4216-88ce-1c255f8edaab/zeionara/neo4j/backups

**7.-Ilm.md**

**General description**

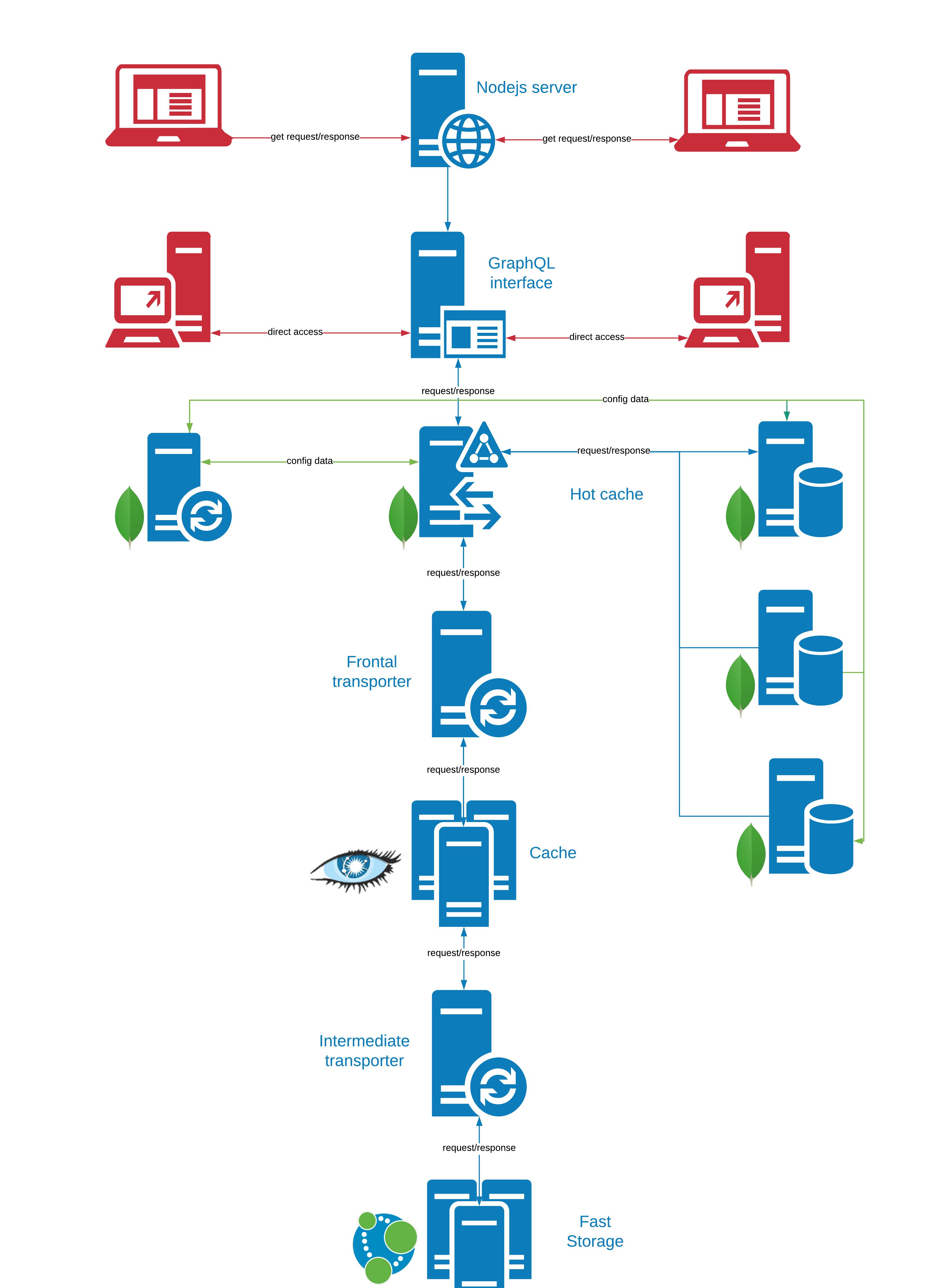
The ilm (which stands for **information lifecycle management**) model is a result of rethinking and restructuring the project (it is the main cause, why there is an standalone branch for the feature). So, in the old model the data kept in the database has been splitted up into three parts, which matches three dbms vendors depending on entity's functional dedication - some of them are located in mongo, the most basic data, collections, which represent journals and logs are located in cassandra in order to get statistics faster and connections between base entities are represented in neo4j, which gives a convenient way to visualize relations between objects.

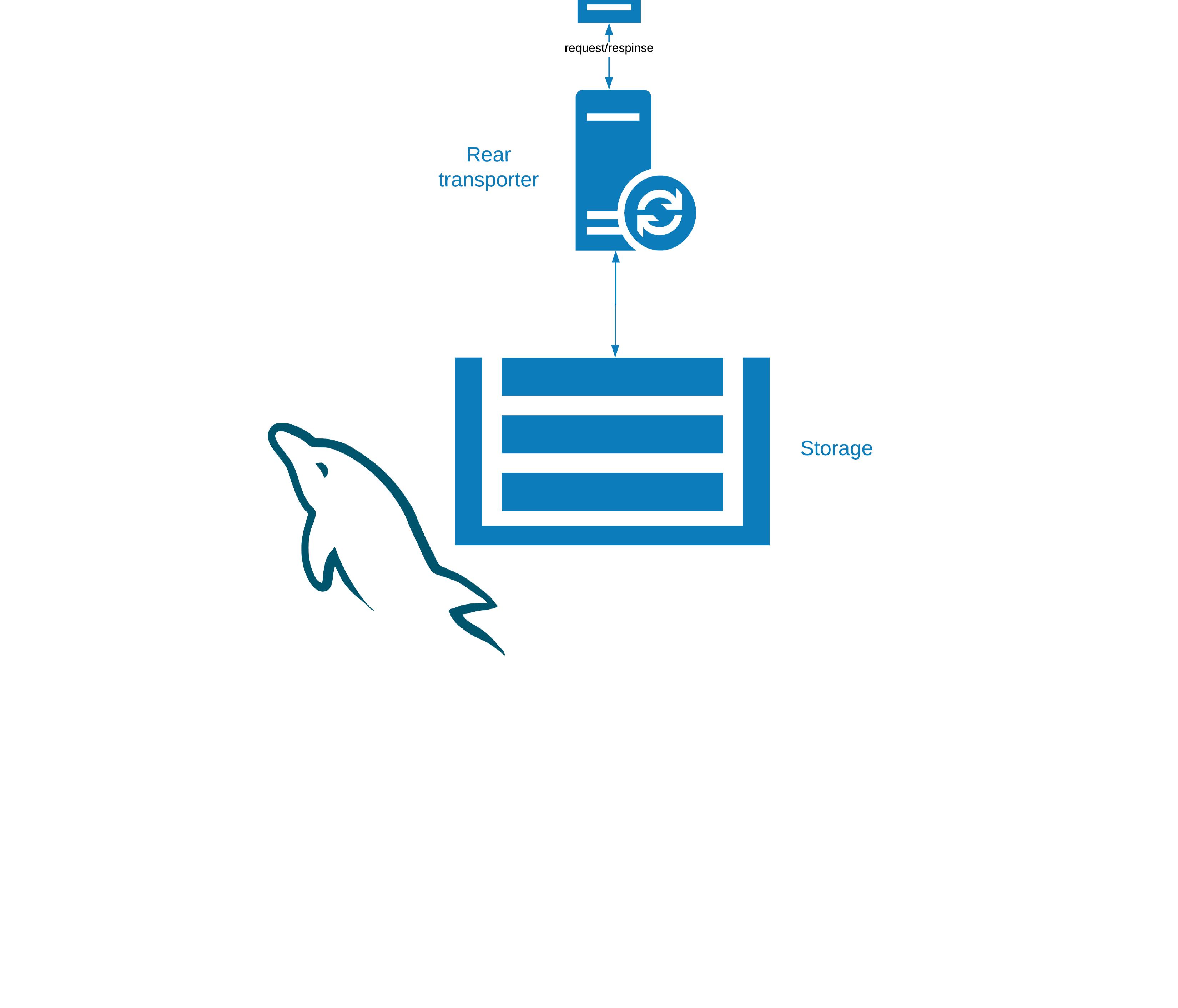
When building the ilm model an idea about keeping the most recent and the most needed data near to user in order to increase speed of fetching required data has been considered rather than functional dividing of data into several blocks. As a result each dbms from three vendors became a **layer** of the model, which might be primarily described by distance to the end-user. So, each dbms also came to containing schemas for every entity type

because now each entry might be occured on any layer of the model. Moreover, an additional external dbms has been taken to represent the lowest layer of the model. Because of low importance of required writing latency and speed as well as high requirements to the reliability, **RDBMS MySQL** has been used.

**Architecture overview**

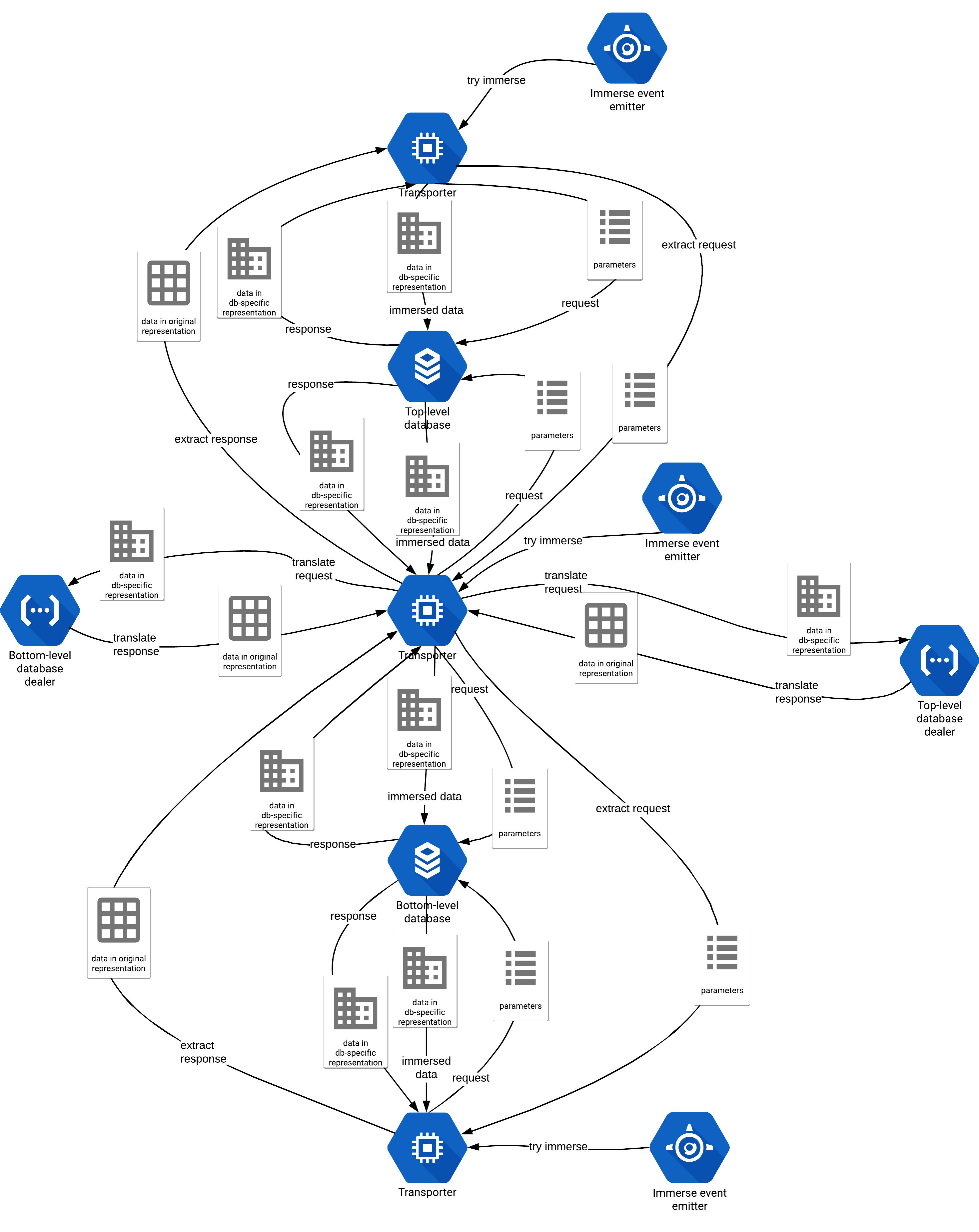
[View in full size (https://image.ibb.co/hFFk17/Blank\_Diagram\_Page\_1.png)](https://image.ibb.co/hFFk17/Blank_Diagram_Page_1.png)





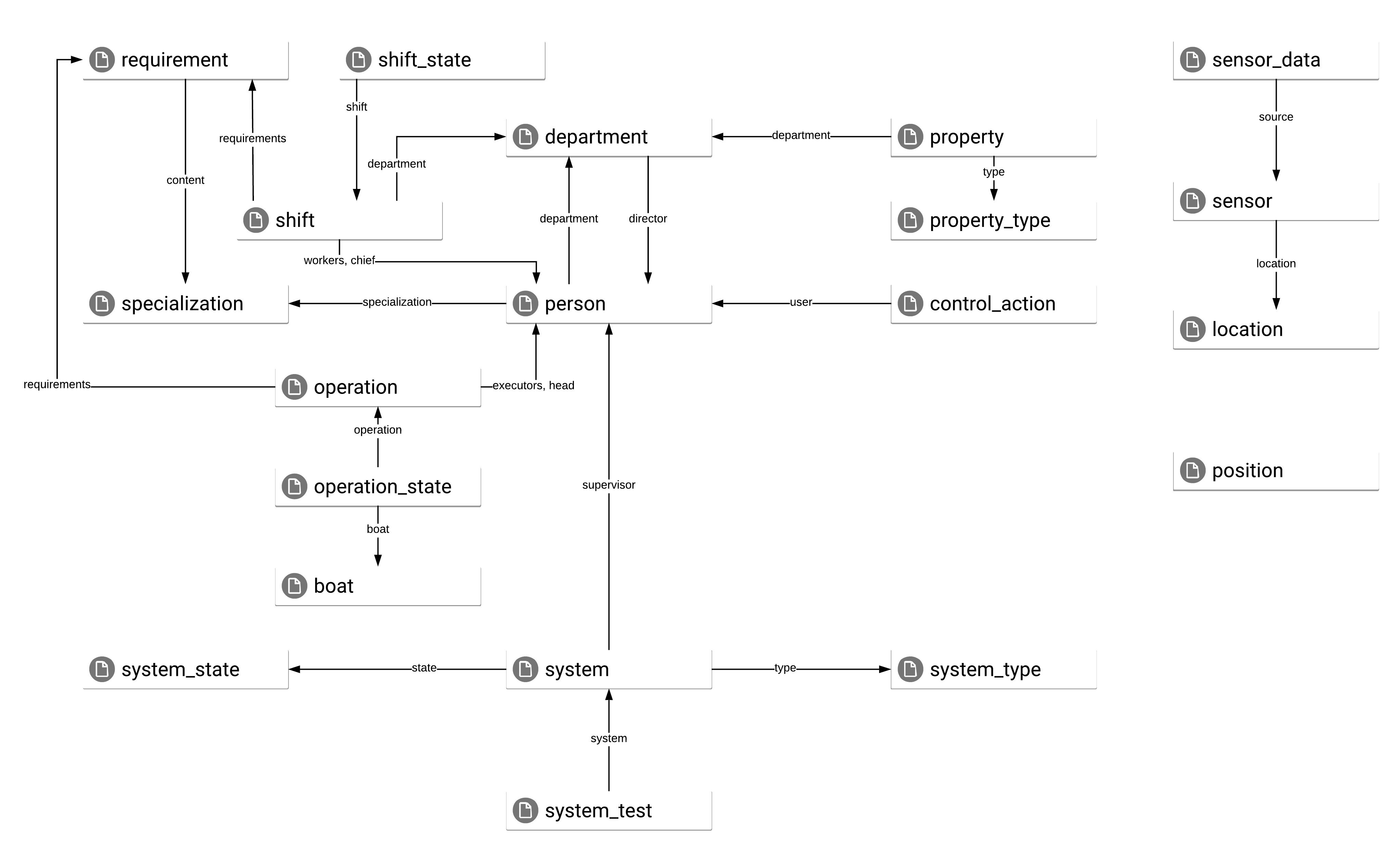
**Transporter's interface overview**

[View in full size (https://image.ibb.co/gHsVrJ/System\_of\_Heroes\_New\_Page.png)](https://image.ibb.co/gHsVrJ/System_of_Heroes_New_Page.png)



**New ER model**

[View in full size (https://image.ibb.co/cW6owJ/er.png)](https://image.ibb.co/cW6owJ/er.png)



It might be noticed that now every object has a set of common fields - \_id , \_\_accessed\_\_ , \_\_created\_\_ , \_\_gaps\_\_ and \_\_cause\_\_ .

\_id represents a unique key, which can be used for addressing an entry as well as for building shard key (e.g.

ObjectId("00000000000000000000000c") )

\_\_accessed\_\_ keeps the moment, when the object have been accessed last time (e.g. ISODate("2016-09-18T16:00:00Z") ).

\_\_created\_\_ contains timestamp which points to the moment of entry creation (e.g. ISODate("2016-09-18T16:00:00Z") ).

\_\_gaps\_\_ contains distances (in seconds) between timestamps of accessing an entry (e.g. [5, 60, 3600] which means that the entry has been

accessed at moments \_\_accessed\_\_ , \_\_accessed\_\_ - 3600 , \_\_accessed\_\_ - 3660 , \_\_accessed\_\_ - 3665 ).

\_\_cause\_\_ is a cause of moving an entry from upper level to a lower one (it is equal to null if an object has came from bottom in other cases it is a text value).

**boats**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| capacity | int |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**departments**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| vk | text |
|  |  |
| director | blob |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**property\_types**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| description | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**system\_states**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| description | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**system\_types**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| description | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**specializations**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**locations**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**sensors**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| location | blob |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**systems**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| type | blob |
|  |  |
| serial\_number | float |
|  |  |
| launched | timestamp |
|  |  |
| checked | timestamp |
|  |  |
| supervisor | blob |
|  |  |
| state | blob |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**people**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| surname | text |
|  |  |
| patronymic | text |
|  |  |
| department | blob |
|  |  |
| phone | text |
|  |  |
| specialization | blob |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**properties**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| type | blob |
|  |  |
| admission | timestamp |
|  |  |
| comissioning | timestamp |
|  |  |
| department | blob |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**shifts**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| start | timestamp |
|  |  |
| end | timestamp |
|  |  |
| department | blob |
|  |  |
| chief | blob |
|  |  |
| workers | list of blob |
|  |  |
| requirements | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**operations**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| start | timestamp |
|  |  |
| end | timestamp |
|  |  |
| head | blob |
|  |  |
| executors | text |
|  |  |
| requirements | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**requirements**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| name | text |
|  |  |
| content | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**system\_tests**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| system | blob |
|  |  |
| result | int |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**control\_actions**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| mac\_address | blob |
|  |  |
| user | blob |
|  |  |
| command | text |
|  |  |
| params | text |
|  |  |
| result | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**positions**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| x | float |
|  |  |
| y | float |
|  |  |
| z | float |
|  |  |
| speed | float |
|  |  |
| attack\_angle | float |
|  |  |
| direction\_angle | float |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**sensor\_data**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| source | blob |
|  |  |
| event | text |
|  |  |
| meaning | text |
|  |  |
| value | float |
|  |  |
| units | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**shift\_states**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| shift | blob |
|  |  |
| warning\_level | text |
|  |  |
| \_cartridges | int |
|  |  |
| air | int |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| electricity | int |
|  |  |
| comment | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |

**operation\_states**



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| \_id | blob |
|  |  |
| timestamp | timestamp |
|  |  |
| boat | blob |
|  |  |
| operation | blob |
|  |  |
| status | text |
|  |  |
| distance | float |
|  |  |
| zenith | float |
|  |  |
| azimuth | float |
|  |  |
| hydrogenium | float |
|  |  |
| helium | float |
|  |  |
| lithium | float |
|  |  |
| beryllium | float |
|  |  |
| borum | float |
|  |  |
| carboneum | float |
|  |  |
| nitrogenium | float |
|  |  |
| oxygenium | float |
|  |  |
| fluorum | float |
|  |  |
| neon | float |
|  |  |
| natrium | float |
|  |  |
| magnesium | float |
|  |  |
| aluminium | float |
|  |  |
| silicium | float |
|  |  |
| phosphorus | float |
|  |  |
| sulfur | float |
|  |  |
| chlorum | float |
|  |  |
| argon | float |
|  |  |
| kalium | float |
|  |  |
| calcium | float |
|  |  |
| scandium | float |
|  |  |
| titanium | float |
|  |  |
| vanadium | float |
|  |  |
| chromium | float |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| manganum | float |
|  |  |
| ferrum | float |
|  |  |
| cobaltum | float |
|  |  |
| niccolum | float |
|  |  |
| cuprum | float |
|  |  |
| zincum | float |
|  |  |
| gallium | float |
|  |  |
| germanium | float |
|  |  |
| arsenicum | float |
|  |  |
| selenium | float |
|  |  |
| bromum | float |
|  |  |
| crypton | float |
|  |  |
| \_rubidium | float |
|  |  |
| strontium | float |
|  |  |
| yttrium | float |
|  |  |
| zirconium | float |
|  |  |
| niobium | float |
|  |  |
| molybdaenum | float |
|  |  |
| technetium | float |
|  |  |
| ruthenium | float |
|  |  |
| rhodium | float |
|  |  |
| palladium | float |
|  |  |
| argentum | float |
|  |  |
| cadmium | float |
|  |  |
| indium | float |
|  |  |
| stannum | float |
|  |  |
| stibium | float |
|  |  |
| tellurium | float |
|  |  |
| iodium | float |
|  |  |
| xenon | float |
|  |  |
| caesium | float |
|  |  |
| barium | float |
|  |  |
| lanthanum | float |
|  |  |
| cerium | float |
|  |  |
| praseodymium | float |
|  |  |
| neodymium | float |
|  |  |
| promethium | float |
|  |  |
| samarium | float |
|  |  |
| europium | float |
|  |  |
| gadolinium | float |
|  |  |
| terbium | float |
|  |  |

|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| dysprosium | float |
|  |  |
| holmium | float |
|  |  |
| erbium | float |
|  |  |
| thulium | float |
|  |  |
| ytterbium | float |
|  |  |
| lutetium | float |
|  |  |
| hafnium | float |
|  |  |
| tantalum | float |
|  |  |
| wolframium | float |
|  |  |
| rhenium | float |
|  |  |
| osmium | float |
|  |  |
| \_iridium | float |
|  |  |
| platinum | float |
|  |  |
| aurum | float |
|  |  |
| hydrargyrum | float |
|  |  |
| thallium | float |
|  |  |
| plumbum | float |
|  |  |
| bismuthum | float |
|  |  |
| polonium | float |
|  |  |
| astatum | float |
|  |  |
| radon | float |
|  |  |
| francium | float |
|  |  |
| radium | float |
|  |  |
| actinium | float |
|  |  |
| thorium | float |
|  |  |
| protactinium | float |
|  |  |
| uranium | float |
|  |  |
| neptunium | float |
|  |  |
| plutonium | float |
|  |  |
| americium | float |
|  |  |
| curium | float |
|  |  |
| berkelium | float |
|  |  |
| californium | float |
|  |  |
| einsteinium | float |
|  |  |
| fermium | float |
|  |  |
| mendelevium | float |
|  |  |
| nobelium | float |
|  |  |
| lawrencium | float |
|  |  |
| rutherfordium | float |
|  |  |
| dubnium | float |
|  |  |
| seaborgium | float |
|  |  |



|  |  |
| --- | --- |
| **property** | **data type** |
|  |  |
| bohrium | float |
|  |  |
| hassium | float |
|  |  |
| meitnerium | float |
|  |  |
| darmstadtium | float |
|  |  |
| roentgenium | float |
|  |  |
| copernicium | float |
|  |  |
| nihonium | float |
|  |  |
| flerovium | float |
|  |  |
| moscovium | float |
|  |  |
| livermorium | float |
|  |  |
| tennessium | float |
|  |  |
| oganesson | float |
|  |  |
| comment | text |
|  |  |
| \_\_accessed\_\_ | timestamp |
|  |  |
| \_\_created\_\_ | timestamp |
|  |  |
| \_\_gaps\_\_ | list of numbers |
|  |  |
| \_\_cause\_\_ | text |
|  |  |



Заключение

Итак, в результате выполнения курсовой работы было произведено ознакомление с основными ключевыми типами баз данных, которые концептуально отличаются от реляционных, хранящих данные в виде взаимосвязанных таблиц.

Были сделаны выводы об области применения отдельных типов баз данных в производственной деятельности, а именно, документо-ориентированная модель наиболее универсальная и обладает гораздо большей гибкостью по сравнению с общепринятыми SQL базами данных. Колоночная база данных в высшей степени предназначена для ведения разного рода логов и журналов, с целью дальнейшей группировки данных по столбцам при помощи пользовательских или встроенных агрегирующих фунций, именно такие базы данных ориентированы на режимы работы типа OLAP. Третий тип баз данных, графовые, выделяются наибольшей наглядностью и простотой построения моделей данных, что роднит принципы организации таких баз данных с объектно-ориентированной парадигмой разработки ПО.

Однако, как было показано во второй половине курсовой работы, все типы баз данных могут хранить в себе любые данные вне зависимости от области их применения, хотя в таком случае показатели быстродействия и удобства применения могут оказаться гораздо ниже.

Также выполнение курсовой работы способствовало ознакомлению с программными продуктами, упрощающими взаимодействие с базами данных при помощи таких языков программирования, как Java и Python. Была на практике реализована возможность создания пользовательских функций, которые могли бы быть вызваны тем же способом, что и встроенные.

Помимо всего прочего, были написаны скрипты для генерации тестовых данных при помощи сторонней библиотеки. Было измерено и оценено время заполнения базы данных большими объемами данных, которое, как оказалось, сильно зависит от используемой технологии.