Работа с базами данных в платформе Thingworx

**Теоретическая часть**

Для структурированного хранения больших объёмов данных используются базы данных. Программа, позволяющая создавать и управлять базами данных, называется системой управления базами данных (СУБД). В настоящее время популярны реляционные базы данных. Реляционная (табличная) база данных представляет собой множество связанных между собой таблиц. Известные реляционные СУБД: MS Access, MySQL, MS SQL Server, Oracle и т.д.

Обычно СУБД не содержит средств создания пользовательского приложения (исключение – MS Access), то есть приложения, которое позволяет конечным пользователям легко работать с базой данных. Вместо этого, СУБД предоставляет средства получения доступа к базе данных и её изменению из пользовательских приложенийжд, написанных на различных языках программирования (C/C++, C#, Delphi, Java, PHP и т.д.).

Java Database Connectivity (JDBC, соединение с базами данных на Java) – способ подключения базы данных к приложениям, написанным на Java.

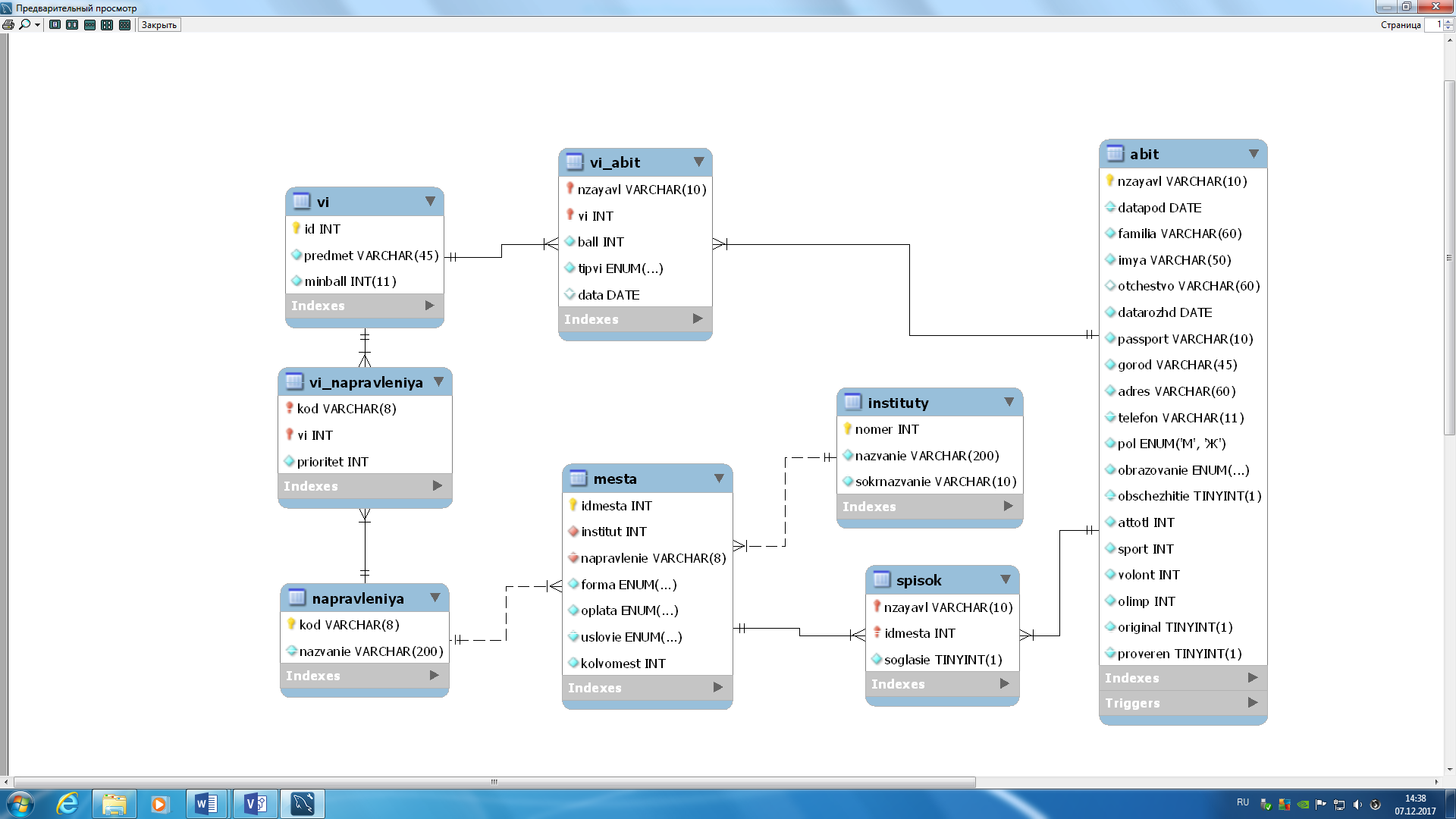
Платформа Thingworx позволяет организовать работу с любой СУБД, поддерживающей технологию JDBC. Подключение к СУБД в Thingworx проводится в два этапа:

1. Загрузка драйвера СУБД к Java в Thingworx (либо путём помещения драйвера в специальную директорию на сервере Thingworx, либо импортируя расширение, включающее данный драйвер);
2. Настройка вещи, которая будет представлять собой подключение к СУБД.

**Практическая часть**

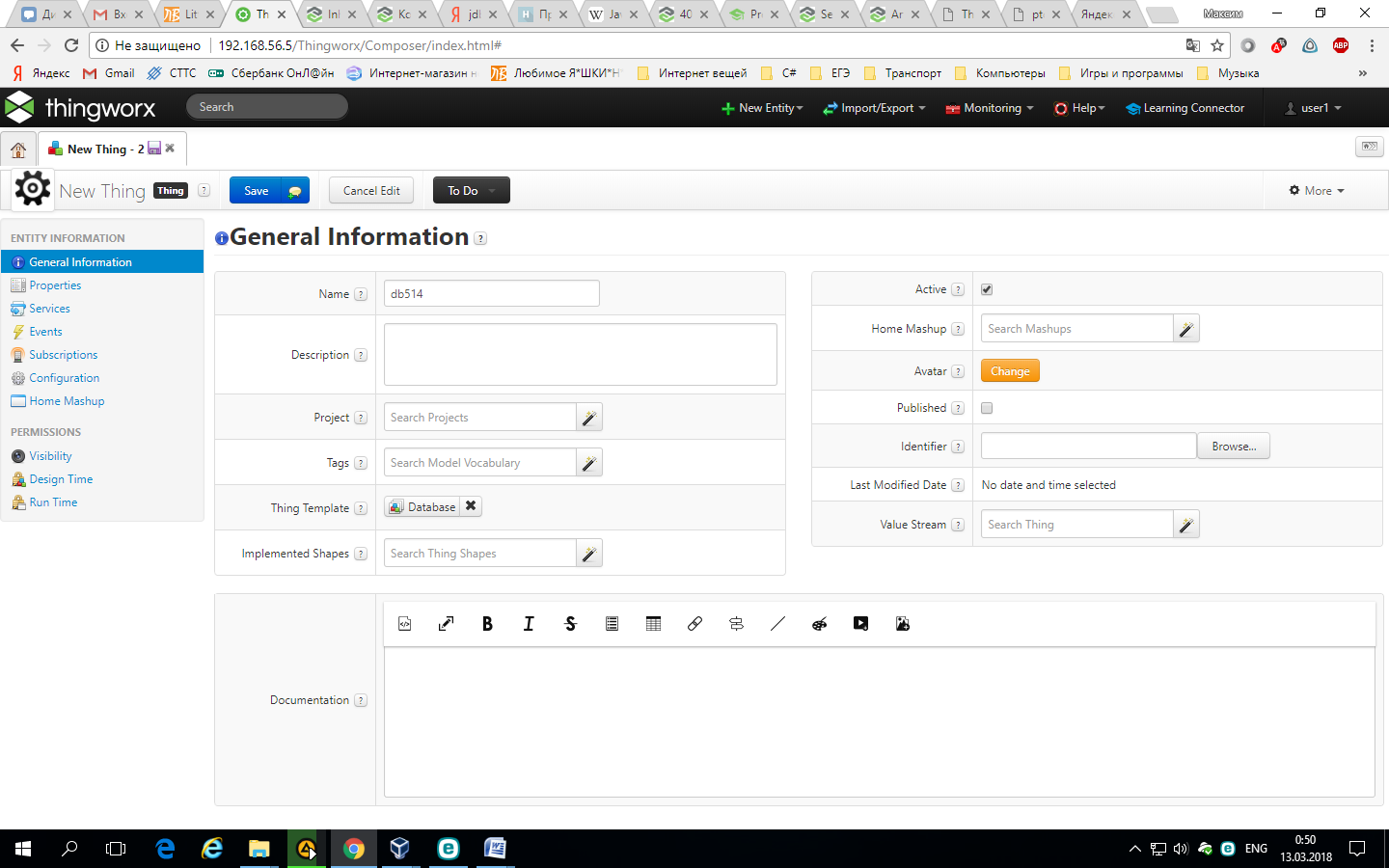
В данной работе будет организовано выполнение SQL запросов на выборку данных к базе данных guap\_abit, которая находится под управлением СУБД MySQL. Администратор сервера Thingworx заранее загрузил драйвер к СУБД MySQL в специальную директорию на сервере.

Структура базы данных guap\_abit:





1. Войдите на сервер Thingworx по адресу, логину и паролю, выданными вам.
2. Создайте новую вещь. Введите произвольное имя, шаблон – Database.



1. Перейдите в раздел Configuration, чтобы настроить подключение к СУБД MySQL.

База данных guap\_abit расположена на сервере MySQL, который находится на том же компьютере, что и сервер Thingworx.

1. Измените настройки следующим образом:

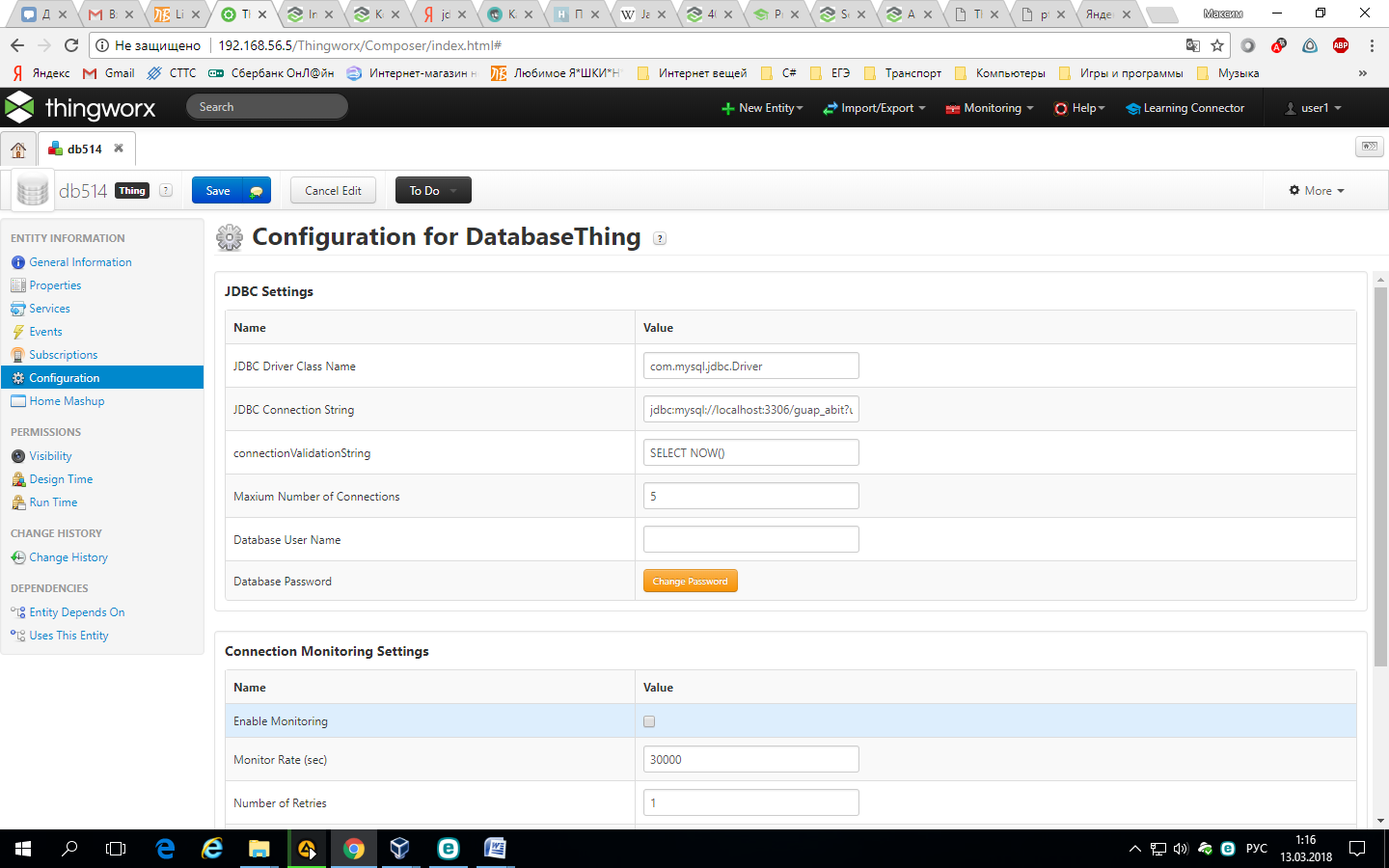
В поле JDBC Driver Class Name (название класса драйвера JDBC для нужной СУБД) напишите com.mysql.jdbc.Driver

Поле JDBC Connection String (строка подключения к СУБД) заполнить так (без пробелов, переносов и т.д.):

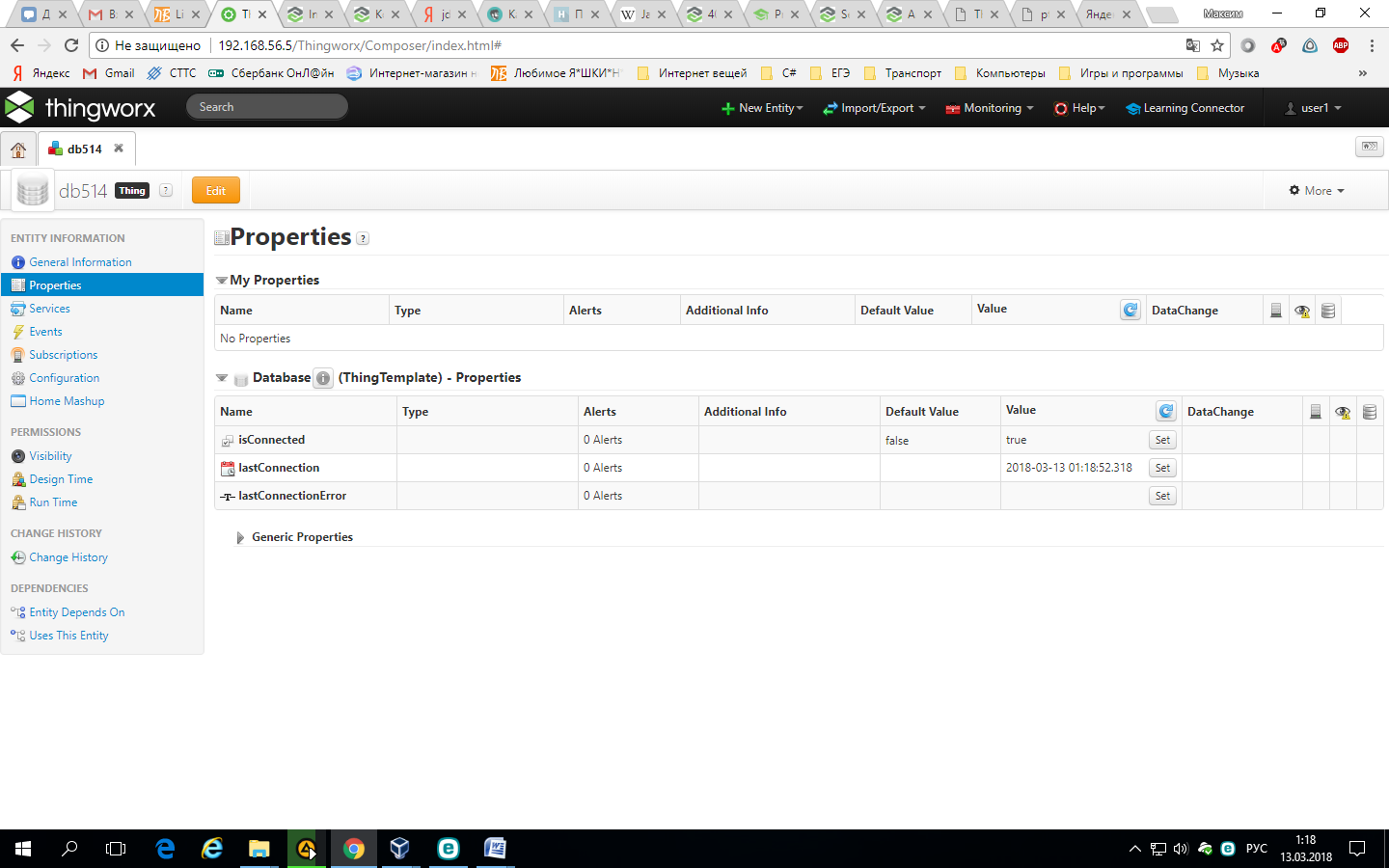
jdbc:mysql://localhost:3306/guap\_abit?user=thingworx&password=UserThingworxPassword

Данная строка обозначает, что происходит подключение к СУБД MySQL, которая расположена по адресу localhost (тот же компьютер, что и сервер Thingworx) на порту 3306 (стандартный порт для MySQL), к базе данных guap\_abit и под пользователем thingworx с паролем UserThingworxPassword.

ОБЯЗАТЕЛЬНО после заполнения поля JDBC Connection String проверьте правильность заполнения – не должно быть никаких пробелов и посторонних символов!



1. Сохраните вещь. Если всё сделано правильно, то на вкладке Properties после сохранения вещи свойство IsConnected должно принять значение true.

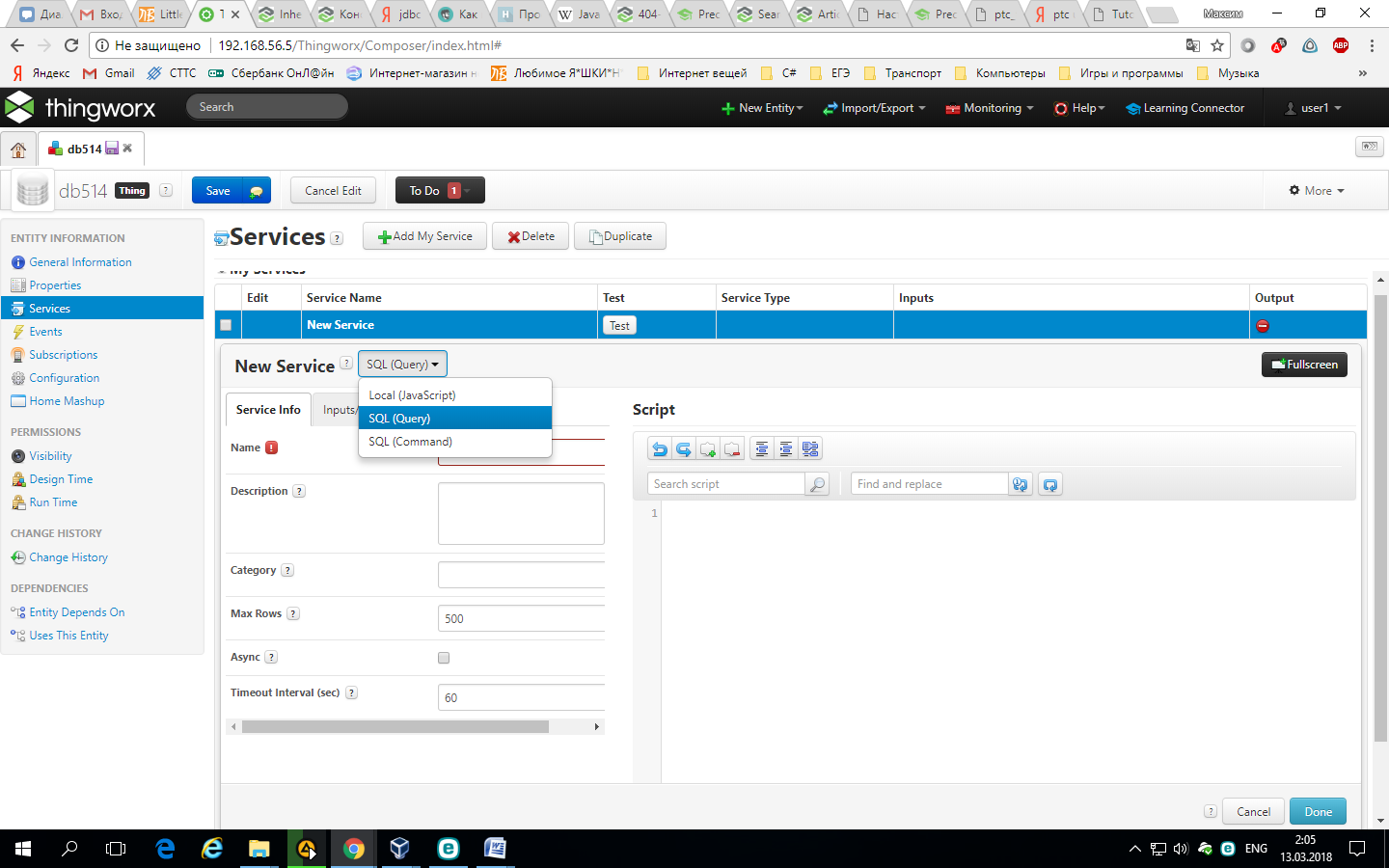


Подключение к базе данных установлено успешно. Теперь можно выполнять запросы к этой базе данных.

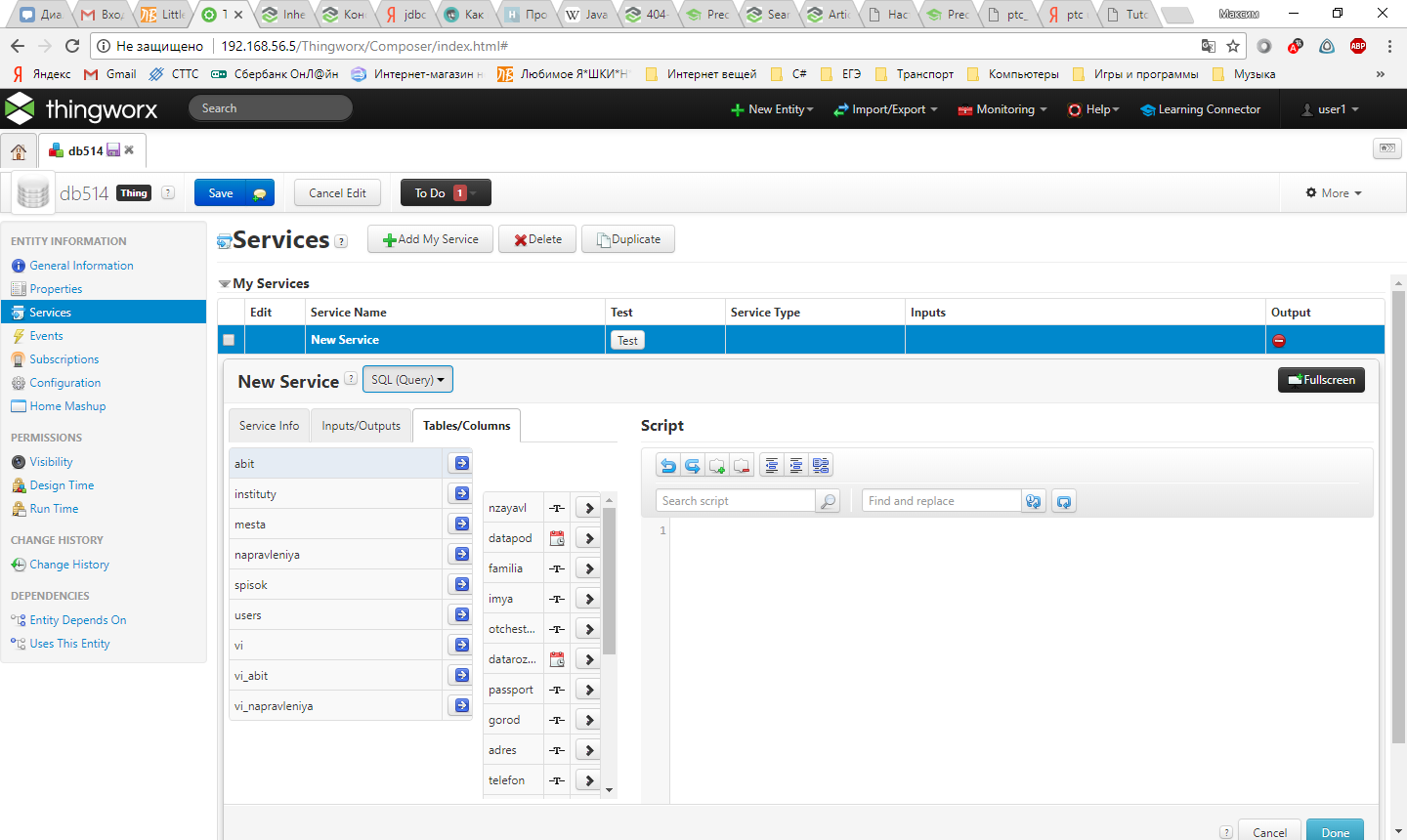
Для вещей, созданных на шаблоне Database, Thingworx позволяет создавать сервисы-запросы непосредственно на языке SQL, причём с учётом диалекта (разновидности) SQL для выбранной СУБД.

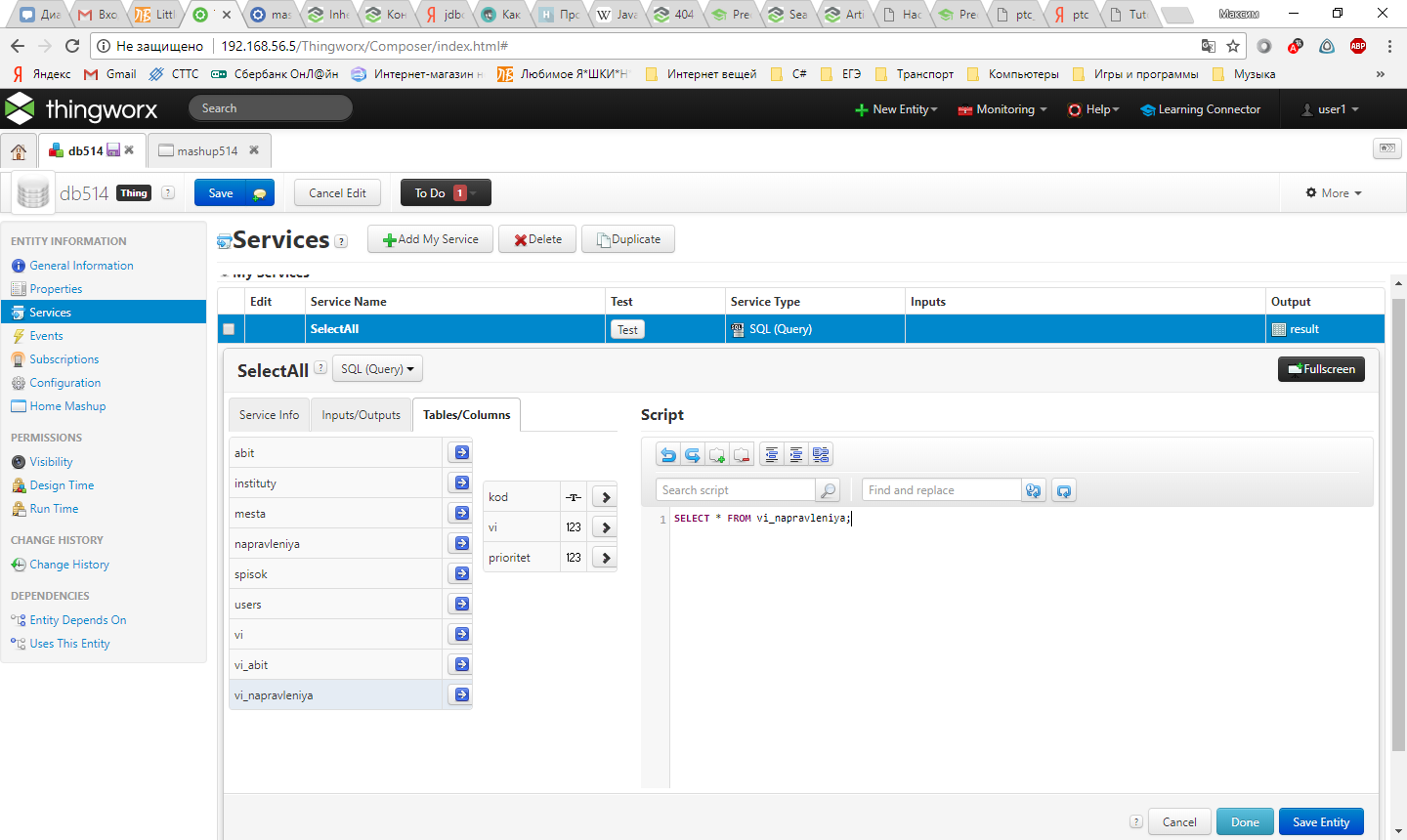
1. Создадим сервис, выводящий все записи из таблицы vi\_napravleniya.

Создайте сервис и задайте ему произвольное имя. Переключите тип сервиса с Local (JavaScript) на SQL (Query).



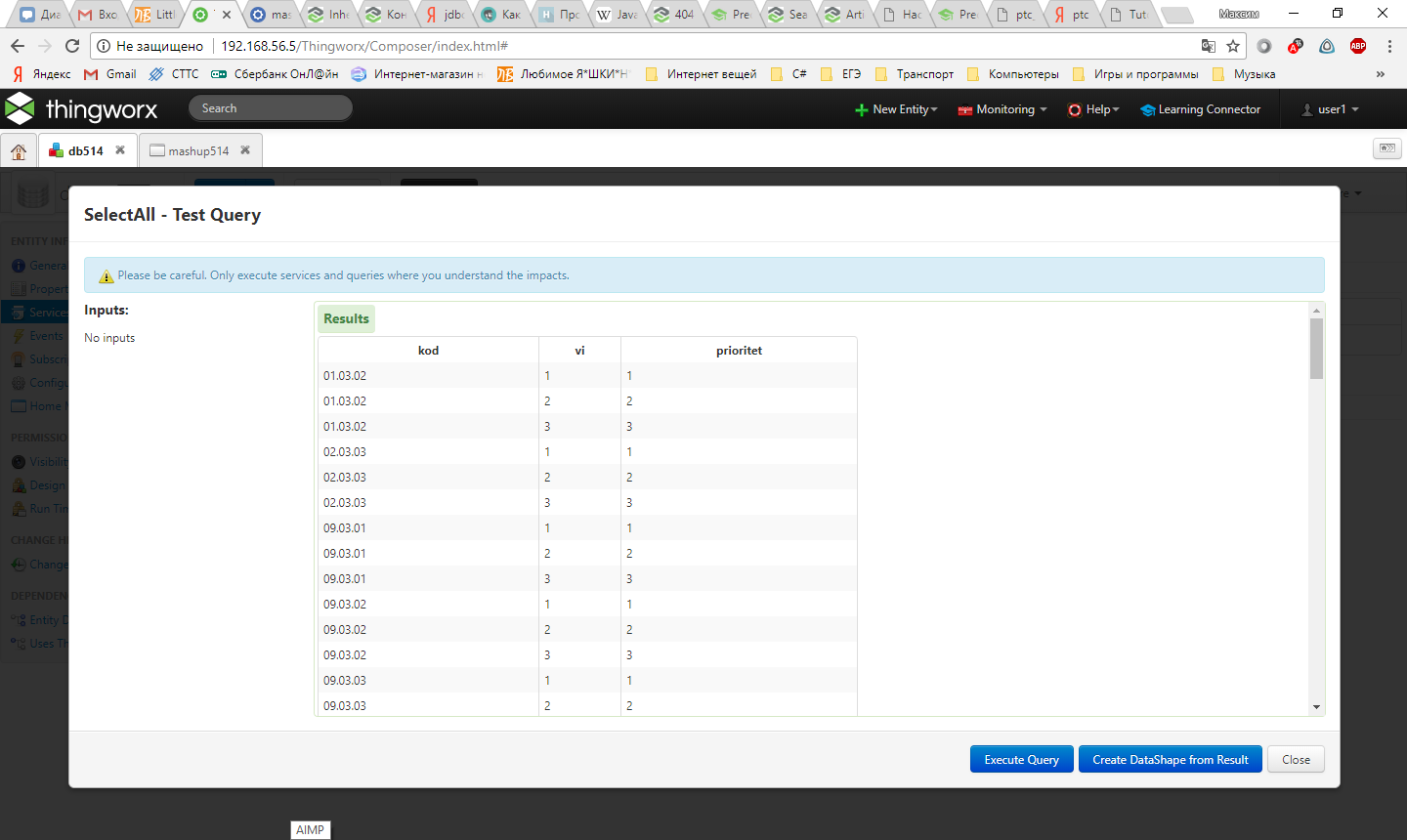
Обратите внимание, что вместо вкладок Snippets, Me, Entities появилась вкладка Tables/Columns, в которой отображаются таблицы и поля таблиц базы данных, к которой подключена вещь.



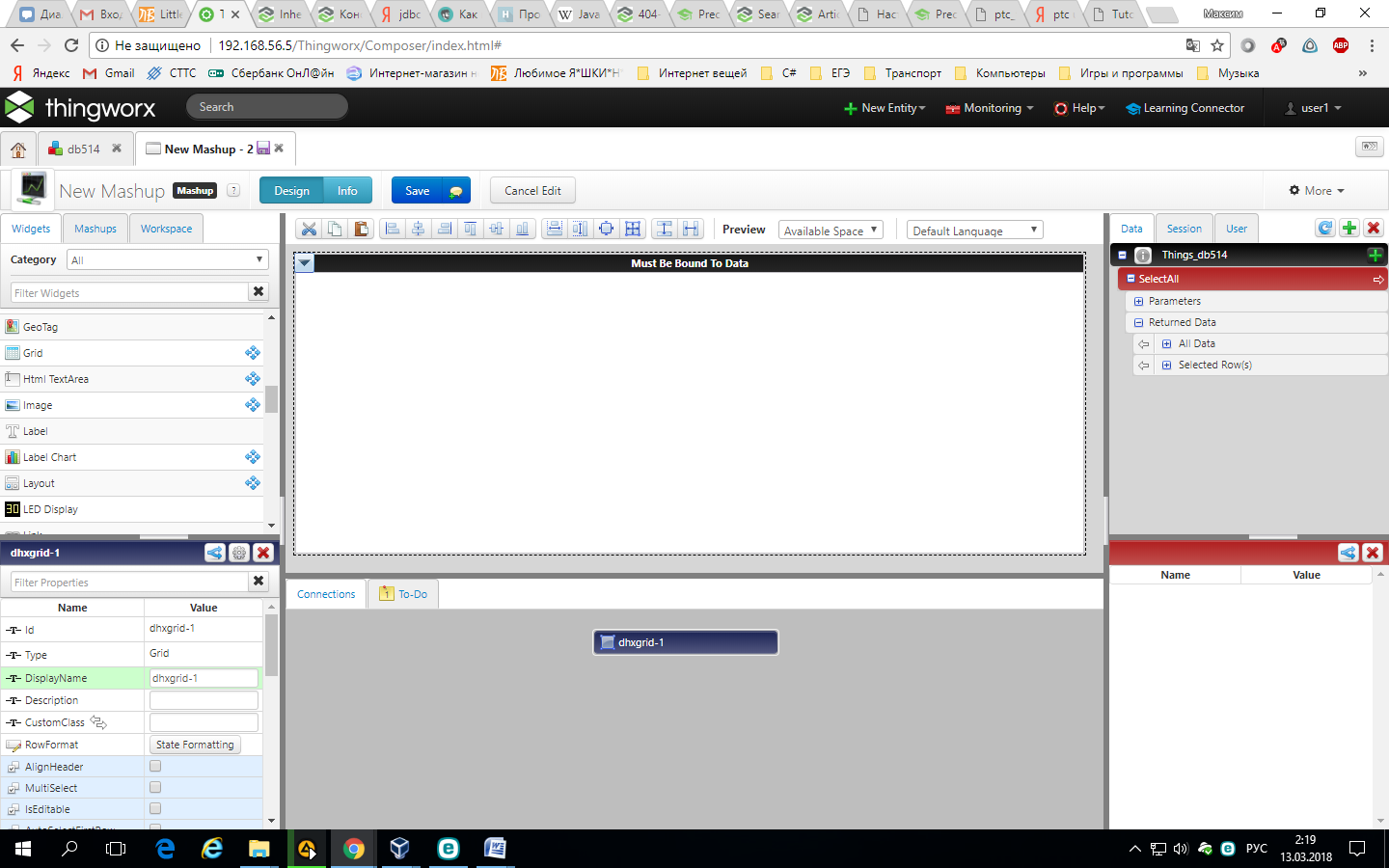
Напишите SQL код сервиса: SELECT \* FROM vi\_napravleniya;  


Сохраните сервис.

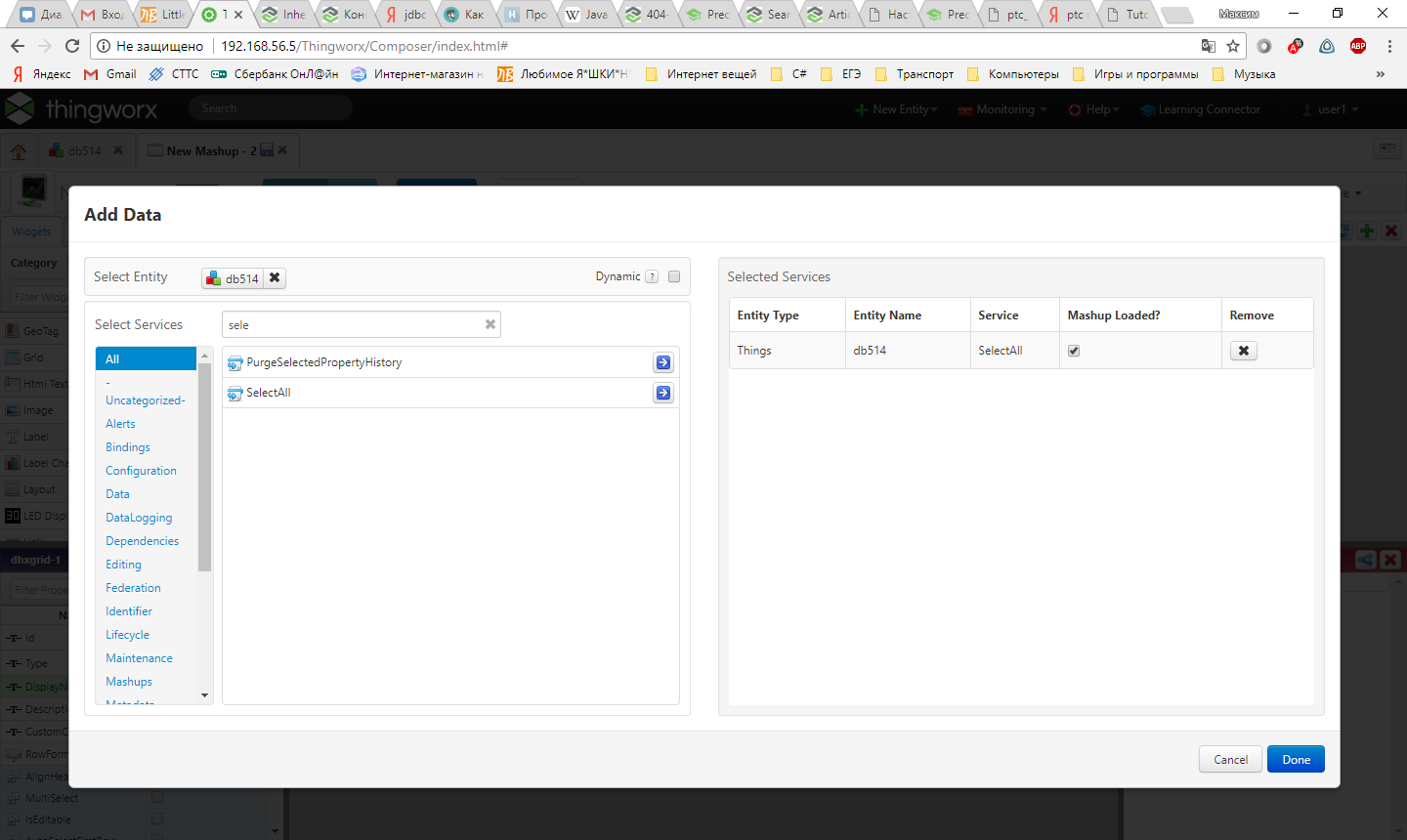
1. Сохраните вещь и протестируйте сервис. В случае успеха сервис выведет первые 500 записей таблицы vi\_napravleniya. Можно изменить максимальное число выводимых сервисом строк, отредактировав у сервиса на вкладке Service Info настройку Max Rows.



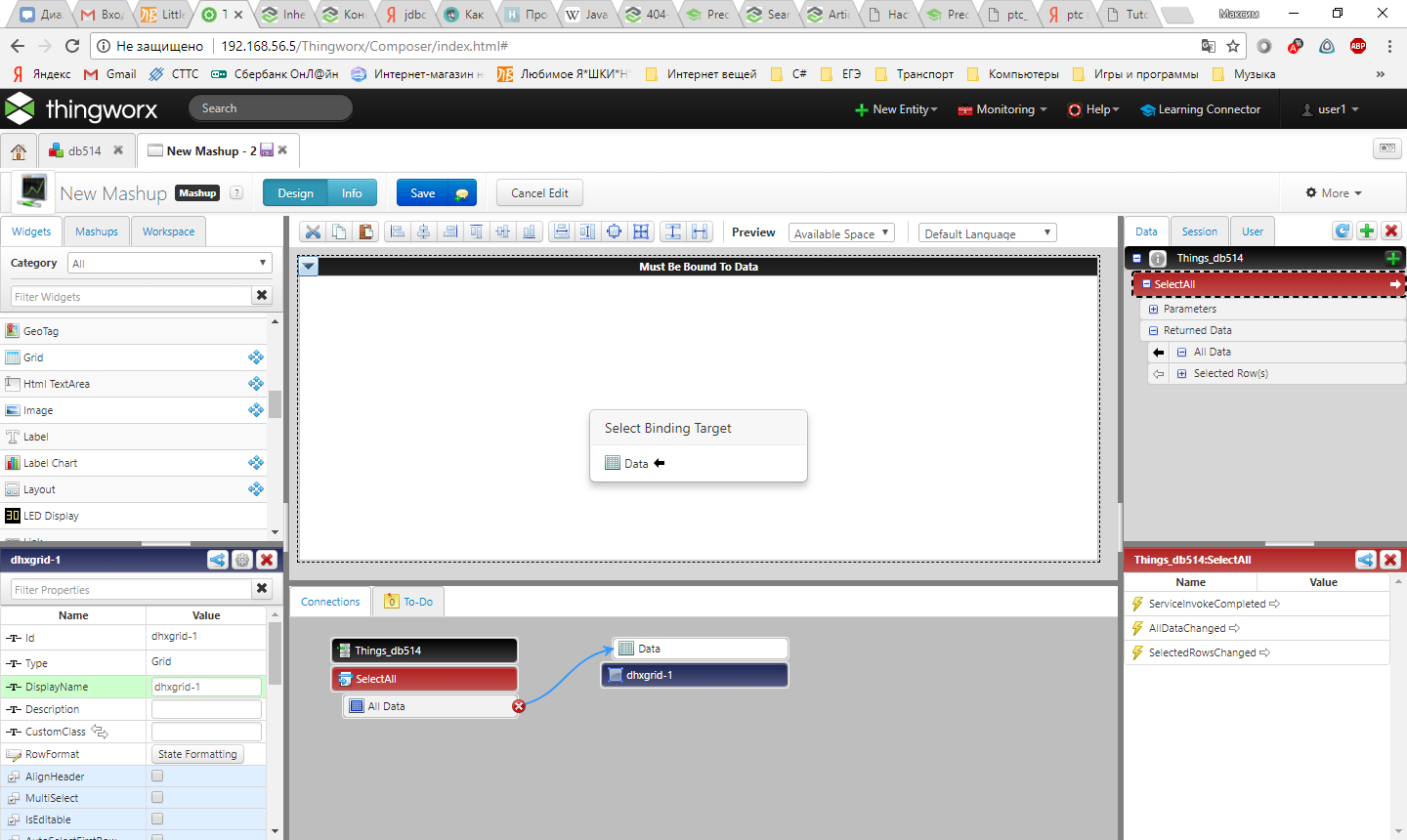
1. Создайте мэшап. Разместите на мэшапе виджет Grid. Возможно, виджет будет растянут на весь экран – это вполне нормально.



1. В правой части конструктора мэшапов найдите «+» и добавьте свою вещь и свой сервис-запрос. Этот сервис должен выполняться при загрузке мэшапа.

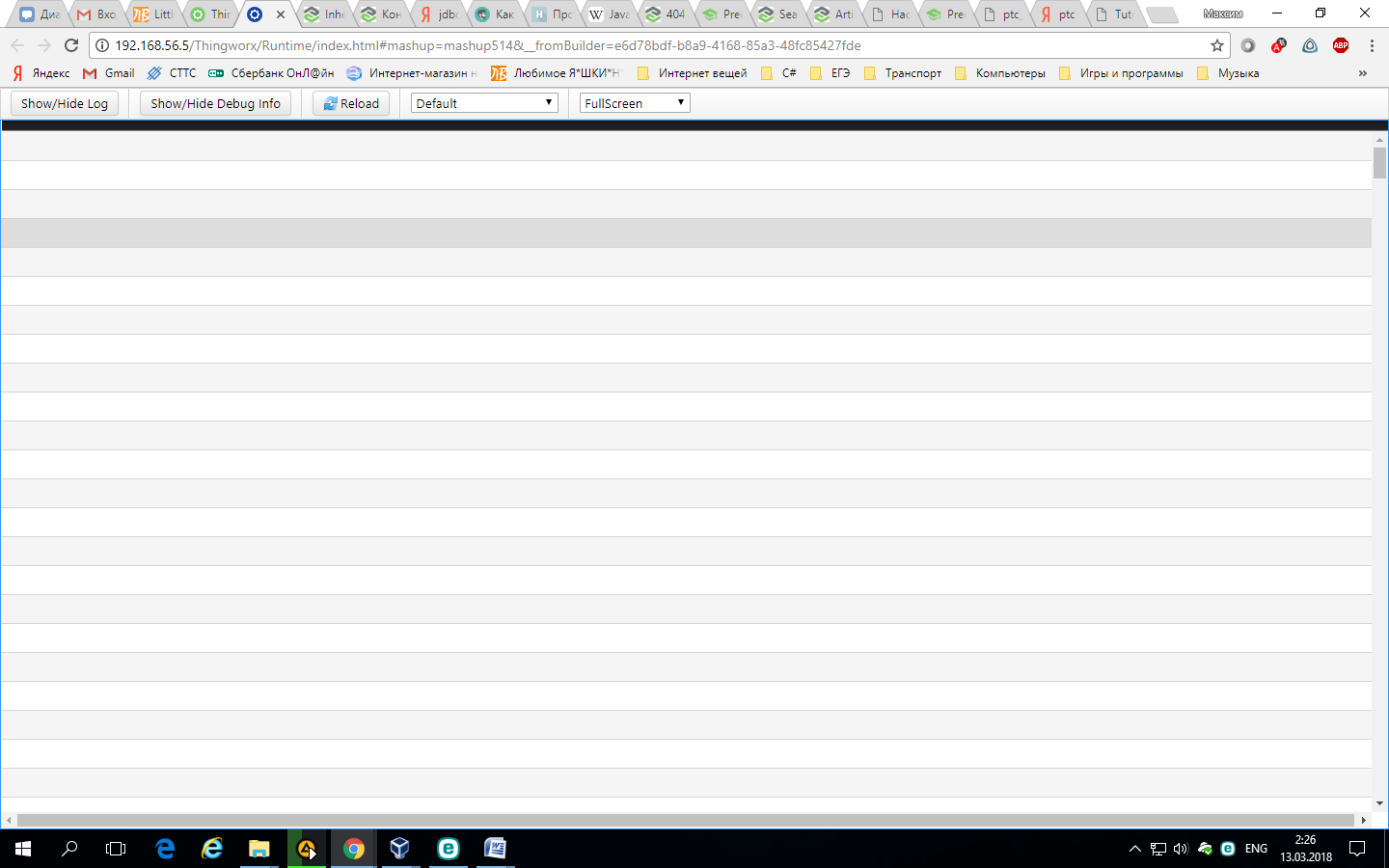


1. В правой части конструктора мэшапов появился ваш сервис. Наведите указатель мыши на слова All Data, зажмите левую кнопку мыши и перетащите на ранее размещённый виджет Grid. Отпустите левую кнопку мыши и выберите Data в появившемся меню.



Если всё будет успешно, то стрелка у All Data станет чёрной.

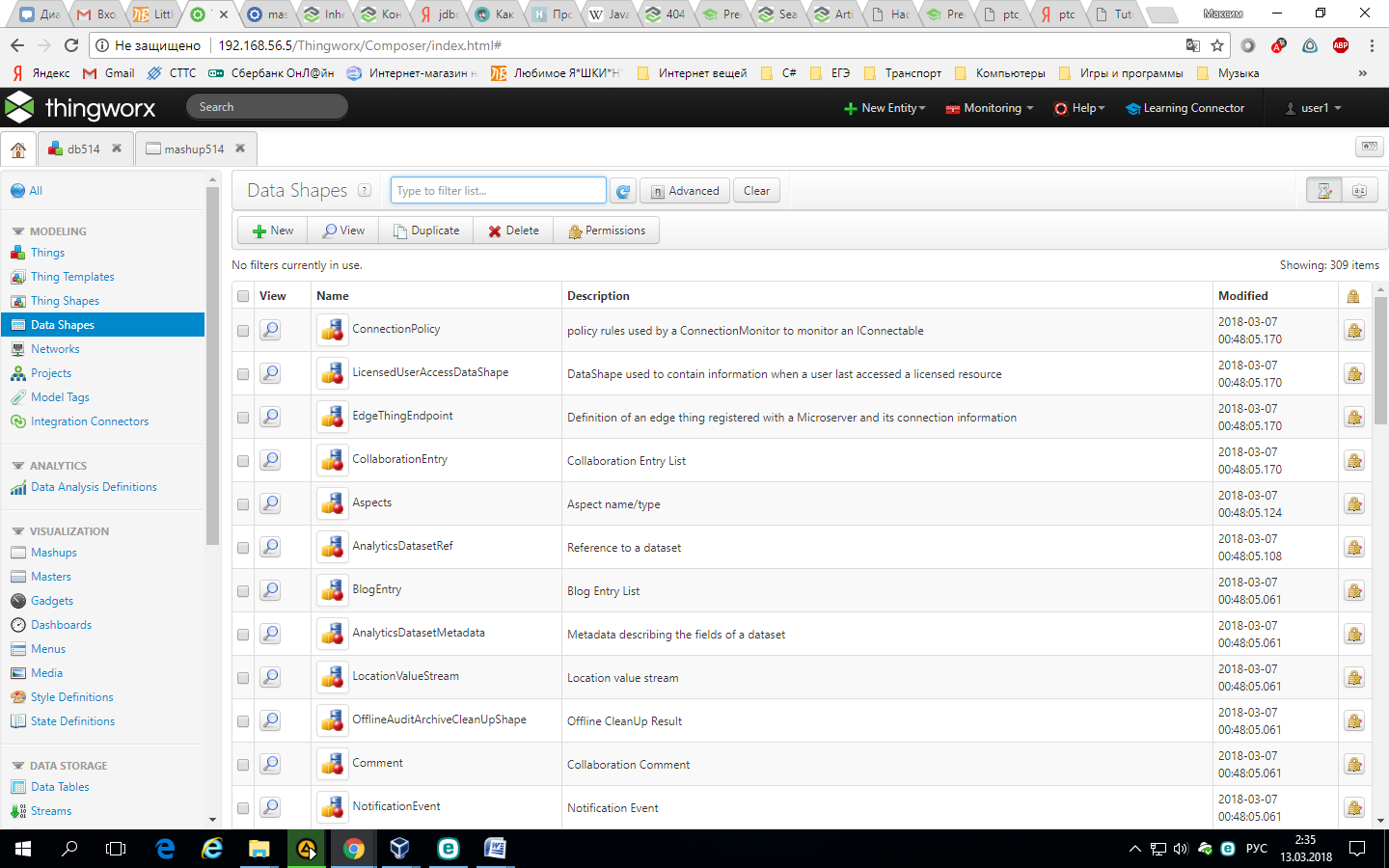
1. Перейдите в конструкторе мэшапов в раздел Info, введите имя мэшапа и сохраните его. Протестируйте мэшап.



Видно, что результат не соответствует нашим ожиданиям.

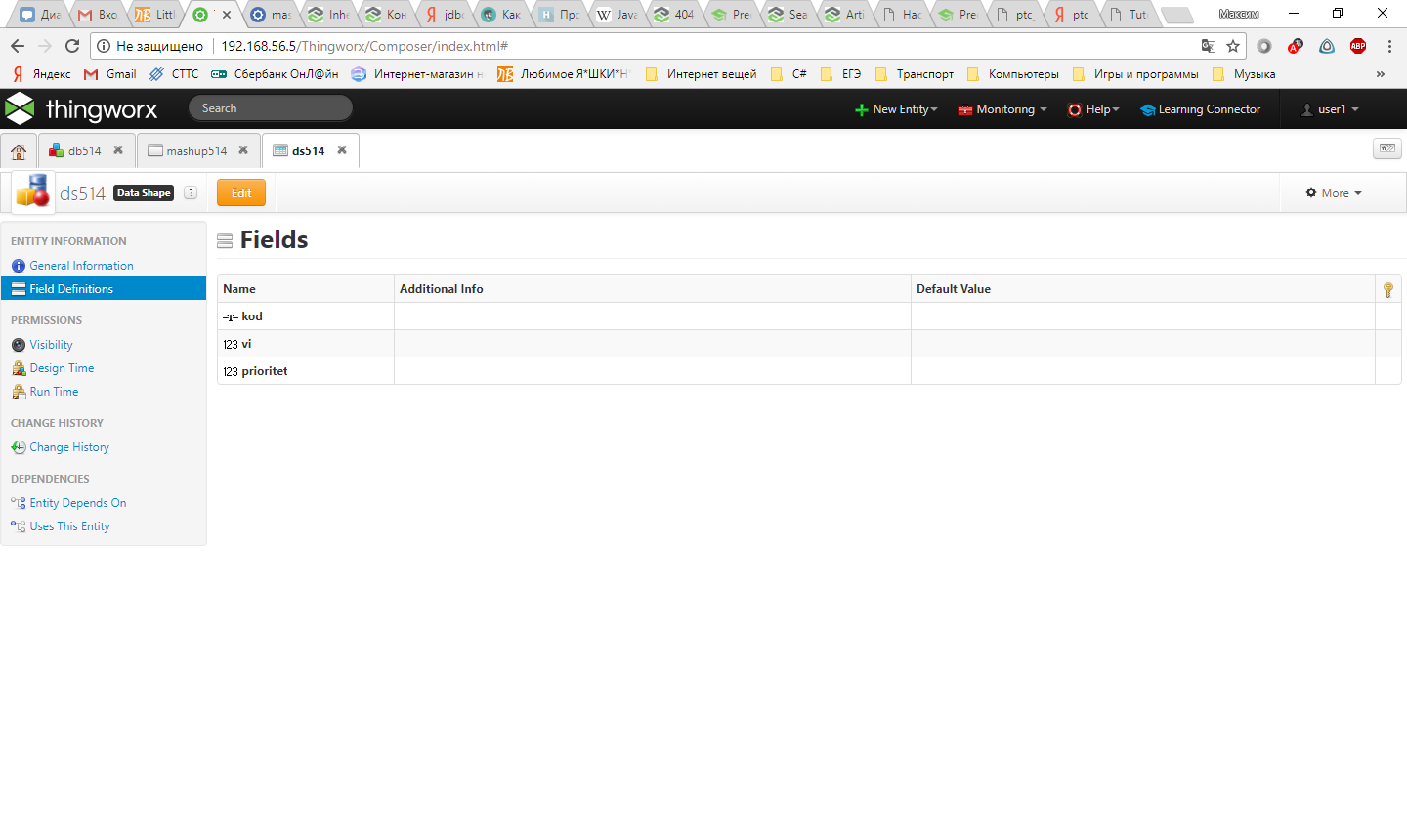
Для корректной работы мэшапа нужно вручную написать перечень полей таблицы с указанием их типов.

1. Перейдите на главную страницу Thingworx («домик») и зайдите в раздел Data Shapes.

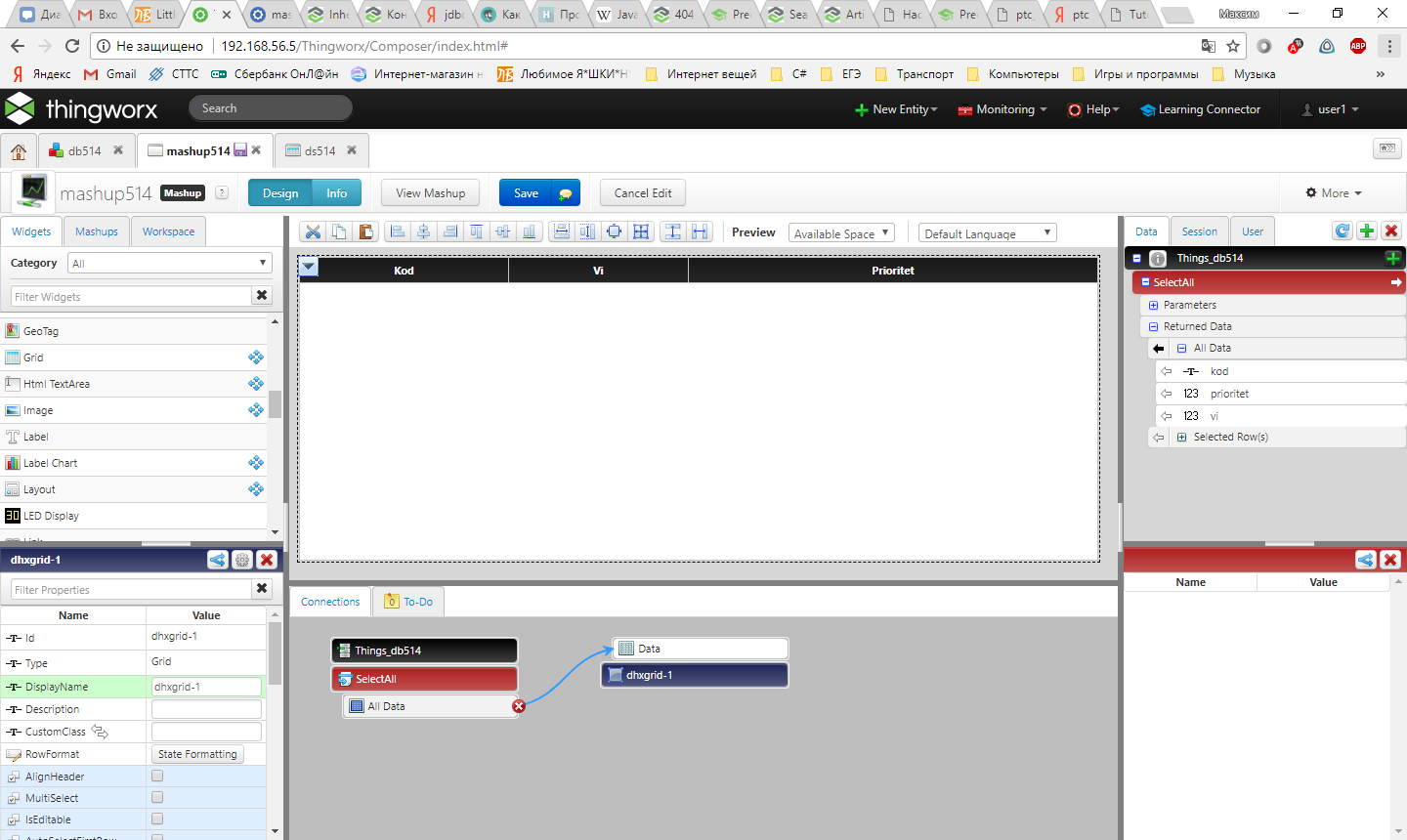


1. Создайте новый Data Shape. Задайте произвольное имя. Перейдите на вкладку Field Definitions. Добавьте три поля:

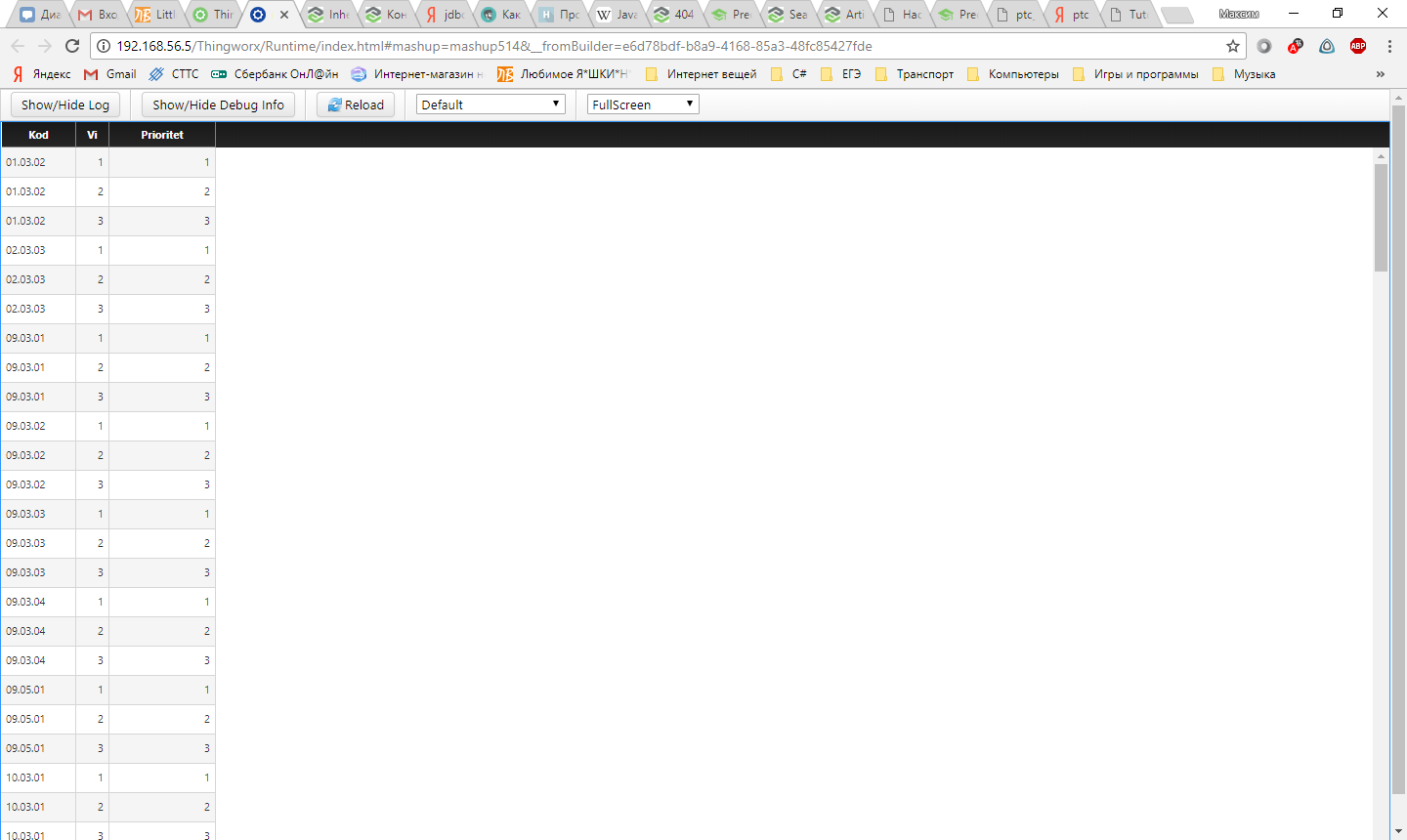
|  |  |
| --- | --- |
| Название поля | Тип |
| kod | STRING |
| vi | INTEGER |
| prioritet | INTEGER |



1. Сохраните Data Shape.
2. Откройте свой сервис-запрос и на вкладке Inputs/Outputs в разделе Outputs в поле Data Shape выберите свой только что созданный Data Shape.
3. Сохраните сервис и сохраните вещь.
4. Откройте снова мэшап в режиме редактирования. Снова перетащите All Data с правой части конструктора мэшапов на виджет Grid. Соглашайтесь на появившееся сообщение. В итоге должны отобразиться названия столбцов.



1. Сохраните мэшап и протестируйте его.



В целом, теперь мы выводим данные пользователю.

1. Создайте сервисы для выполнения следующих SQL-запросов (выводить на мэшапы не нужно) и протестируйте сервисы. Попытайтесь понять, что делает каждый запрос.

Запрос 2.

SELECT \* FROM abit;

Запрос 3.

SELECT \* FROM vi\_napravleniya WHERE kod='09.03.04';

Запрос 4.

SELECT COUNT(\*) FROM napravleniya;

Запрос 5.

SELECT nazvanie FROM instituty WHERE nomer<5;

Запрос 6.

SELECT datarozhd FROM abit ORDER BY datarozhd DESC;

Запрос 7.

SELECT SUM(kolvomest) FROM mesta WHERE institut=5 AND oplata='Бюджет';

Запрос 8.

SELECT vi, AVG(ball) FROM vi\_abit GROUP BY vi;

Запрос 9. Запрос выведет коды направлений, по которым суммарное количество мест более 50.

SELECT napravlenie FROM mesta GROUP BY napravlenie HAVING SUM(kolvomest)>50;

Запрос 10. Запрос выведет все возможные варианты мест на направление «Программная инженерия».

SELECT mesta.\* FROM mesta, napravleniya WHERE mesta.napravlenie=napravleniya.kod AND napravleniya.nazvanie='Программная инженерия';

Попробуйте составить свои запросы к данной базе данных.