



A Business Plan for SylvaDB
(www.testing.sylvadb.com)

Juan Luis Suárez

TFM (PEC4) Master in Multimedia Applications, 2013-2014
Professional Itinerary

Learning Consultant: Sergio Schvarstein Liuboschetz



9 January 2014

Copyright

Business Plan: Juan Luis Suárez

Software Development:

The CulturePlex Lab, University of Western Ontario

Director: Juan Luis Suárez

Technology Manager: Javier de la Rosa

Original Idea: Juan Luis Suárez, Fernando Sancho (University of Seville & The CulturePlex Lab), Javier de la Rosa

Patent Pending in the US Patent and Trademark Office: Application Number 61/859,651

Video Voiceover: David Brown

Abstract

This project presents a business plan to commercialize a graph database management system for users with little or no knowledge of programming. The name of the platform is SylvaDB and a beta version is available at www.testing.sylvadb.com. The business plan for SylvaDB has been designed following the principles of the lean start-up as presented in the lean canvas methodology and it takes advantage of the possibilities of implementing multiple iterations and pivoting as a result of users/customers feedback.

Keywords: graph database, big data, linked data, data schema, SylvaDB, commercialization, business plan

Acknowledgements

This work has been made possible thanks to the generous support provided by the Social Sciences and Humanities Research Council of Canada, the Canada Foundation for Innovation, and the Interdisciplinary Development Initiatives of the University of Western Ontario.

Research, development and innovation on SylvaDB have been conducted by the team of The CulturePlex Lab.

Table of Contents

1. Introduction	8
2. Description/Definition/Hypothesis	9
3. Objectives	10
3.1. Main Objectives	10
3.2. Secondary Objectives	10
4. Methods and tools	11
5. Schedule	12
6. Product Description	14
7. Defining the Product: List of Technical Features	15
8. System Architecture	16
9. User Guide: Getting Started with SylvaDB	19
10. Competition Analysis	41
11. Testing Assumptions (Survey of features and needs)	42
12. Lean Canvas	43
13. Marketing Plan	44
13.1. Industry	44
13.2 The Marketing Challenge	44
13.3 Situation Analysis	44
13.4 Marketing Objectives	45
13.5 Alternatives to Achieve Objectives	46
13.6 Recommendation	47
13.7 Implementation Pan	47
13.8. Unique Value Proposition	48
13.9. Positioning	48
13.10. Market Segmentation	48
13.11. Pricing Strategy	49
14. Projected Profit & Losses Statement	50
15. Conclusions	52
Annex 1. Source Code	53
Annex 2. Screen Shots	53
Annex 3. Academic article about SylvaDB	Error! Bookmark not defined.

Annex 4. Executive Summary

Error! Bookmark not defined.

Annex 6. References

54

Figures and Tables

<i>Fig. 1. General Architecture</i>	16
<i>Fig. 2 Backend</i>	17
<i>Fig. 3 Models</i>	18
<i>Fig. 4 Data Schema of a Project</i>	53
<i>Fig. 5 Screenshot of a Project's Dashboard</i>	54

Tables

<i>Table 1 Schedule</i>	13
<i>Table 2 Marketing Objectives / Phases</i>	46

1. Introduction

The most recent economic trends indicate that the huge penetration of Internet and electronic devices around the world have opened up the possibility to leverage the information resulting from digital exchanges. These digital exchanges provide a huge amount of data that we can now collect, store, analyze and interpret in real time with many different purposes. The only problem is that the access to the benefits of “Big Data” is for now limited to big organizations with the ability, funds, and expertise to exploit the flows of data generated by human beings in their daily activities.

We propose to give the possibility of dealing with Big and Linked Data to any individuals and organizations regardless their knowledge and skills in programming. We have developed a platform called SylvaDB that makes it very easy to create data schema, modify them, import and export data, query and analyze it. The objective of the project is to confer individuals and professionals the means to get the most out of their data so that they can achieve the desired solutions in areas such as medical and pharmaceutical research, biology, marketing, education, and any other field that requires dealing with Big Data in an intuitive and direct way.

2. Description/Definition/Hypothesis

The aim of this project is to develop a business plan for SylvaDBⁱ, a graph database management system easy to use even for people with no programming skills or computer sciences knowledge.

This business plan will help to define an appropriate framework for the commercialization of the product, including unique value proposition, market research, competition analysis, market positioning and segmenting, costs structure, pricing strategy, go-to-market strategy, and future technical development of the platform.

Through a rigorous implementation of such a business plan, SylvaDB will become the market leader in easy-to-use interfaces for big data and link data.

3. Objectives

3.1. Main Objectives

- Develop a business plan that will lead to a successful commercialization of SylvaDB
- Define the main parameters of a comprehensive go-to-market strategy
- Link future technical developments to the product's value chain and customer needs

3.2. Secondary Objectives

- Gain market insight into Big Data commercial needs

4. Methods and tools

For the implementation of this project and the design of a business plan for SylvaDB I will follow the principles, techniques and tools of the lean method for start-ups, including the canvas tools by leanstack.comⁱⁱ and by businessmodelgeneration.comⁱⁱⁱ, the business model analysis for entrepreneurs, and various marketing concepts and techniques related to the four P's (product, price, place, and promotions) and the five C's (consumer, company, competition, collaborators, and context) of strategic marketing management.

The lean method for start-ups was popularized by Steve Blank in his blog (<http://steveblank.com/>)^{iv} and the book "Four Steps to the Epiphany"^v and has been widely adopted in Silicon Valley and other innovation centers around the world. Blank adamantly encourages entrepreneurs in any industry sector, but specially in technological start-ups, to get out of the office and figure out what customers really want (as opposed to what engineers would like to develop) before starting any serious coding of products or services. These would reduce the number of errors related to poor business planning and modeling and limit the costs incurred by failing start-ups. Other important principles of the lean start-up method include rapid iteration, customer feedback, and testing your assumptions with real and potential customers. As part of the so-called "start-up revolution" several tools have been created to help entrepreneurs implement business plans that follow these principles. One of the most widely used in the first phases of starting up is the canvas designed by leanstack.com (see Appendixes). An app versioned for the iPad was later developed by businessmodelgeneration.com.

The business model analysis for entrepreneurs is described in a homonymous article by Thomas Eisenmann and published by the Harvard Business Review in 2012. The analysis focuses on some of the same ideas already expressed by Blank and appropriate by many start-ups, but arranges them according to the four distinctive categories under which the activities need to deliver value and earn sustainable profits can be organized: customer value proposition; technology and operations management; go-to-market plan; profit formula.

The various principles and concepts of marketing management mentioned in the introduction to this section have been staples in the literature about customer-centric marketing and can be found in sources such as "What Customers Want"^{vi}, "Understanding Digital Marketing"^{vii}, and textbooks like "Marketing Management"^{viii}.

Finally, due to its simplicity I will use the free online tool Trello (trello.com)^{ix} as my project management tool, so that I take advantage of the simple and efficient organization allowed by the three columns (To Do; Doing; Done) layout that make up a board in Trello.

5. Schedule

	October				November				December			
Tasks	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12
First Submission (Formal Proposal)												
Draft Lean Canvas		xxx										
Competition Analysis												
Market Research		xxx	xxx	xxx								
Testing Assumptions (survey of features and needs)		xxx										
Defining the Product: Complete List of Technical Features (and associated value)			xxx									
Second Submission (first partial version)												
Market Positioning					xxx							
Market Segmenting					xxx							
Unique Value Proposition Definition					xxx							
Cost Structure						xxx	xxx					
Pricing Strategy							xxx	xxx				
Third Submission (second partial version)												
Go-to-Market Plan (Channels, Campaign and Partnerships)									xxx	xxx		

Operations and Technology Plan										xxx	xxx	
Final Submission												

Table 1 Schedule

6. Product Description

SylvaDB is a graph database¹ management system designed to be used by no programmers. SylvaDB^x is based on flexible schema definitions and has been developed taking into account the need to deal with semantic information. It relies on the mathematical notion of property graph.

Sylva aims at lowering the barrier of adoption for anyone needing the power of graph databases. At the same time, it is robust and scalable enough to support collaborative large projects related to research, knowledge management, and document archiving.

In commercial terms, SylvaDB seeks to establish a strong position in the market of graph databases management systems by designing a method and a set of interfaces that will make it very easy to use as well as by occupying a strategic position in between the segments of big data and linked data.

Some of the technical advanced features of SylvaDB include:

- A controlled language to perform queries in natural language to be translated into graph queries in the back end. The system generates the grammars on the fly to adapt to the specificity of each domain as defined by the pair schema-data of each graph. It uses a LL parser.
- Easy-On-Demand deployment of new back-end instances that gives users control of their newly generated instances only through the system's interface.
- An editing system for schema modification by the own user that requires no programming knowledge, along with automated migration and evolution of schema-data through an assistant wizard.
- A graphical aid system for schema creation by the own user that requires no programming knowledge.
- A smart system for data and schema import from tabular graph data.

SylvaDB has been prototyped by The CulturePlex Lab^{xi} at the University of Western Ontario, Canada, as a result of the lab's need to use a database that would allow researchers to manipulate rich and varied data in projects related to cultural modeling, cultural networks and digital humanities. The research and prototyping phases of the project have been made possible thanks to the financial support of the Social Sciences and Humanities Research Council of Canada, the Canada Foundation for Innovation, the Government of Ontario, and the University of Western Ontario's Interdisciplinary Development Initiatives program.

The market research report for SylvaDB has been commissioned to MARS^{xii} through London's TechAlliance^{xiii} and WorldDiscoveries^{xiv}.

SylvaDB is an open source project (AGPL) that has a provisional patent pending in the US and Canada.

¹ "A graph database is a [database](#) that uses [graph structures](#) with nodes, edges, and properties to represent and store data. By definition, a graph database is any [storage system](#) that provides index-free adjacency. This means that every element contains a direct [pointer](#) to its adjacent element and no index [lookups](#) are necessary. General graph databases that can store any graph are distinct from specialized graph databases such as [triplestores](#) and [network databases](#)." (Source: Wikipedia)

7. Defining the Product: List of Technical Features

- Flexible data schemas
 - editing system for schema modification
 - graphic aid system for schema creation
- Polyglot Graph Backend
- Smart system for data and schema import from tabular graph data
- Easy-On-Demand deployment of new back-end instances
- Approximate Natural Language Query System
- Collaboration
- Visualization
- Analytics

8. System Architecture

Although the main objectives of this work are related to the writing of a business plan and, specifically, a marketing plan, for SylvaDB, it is necessary to delineate the basic components of the system architecture to better understand the links between the technological innovation provided by the product and the value added through the commercialization of such a system. These figures, along with the User Guide contained below, will help you familiarize with the platform.

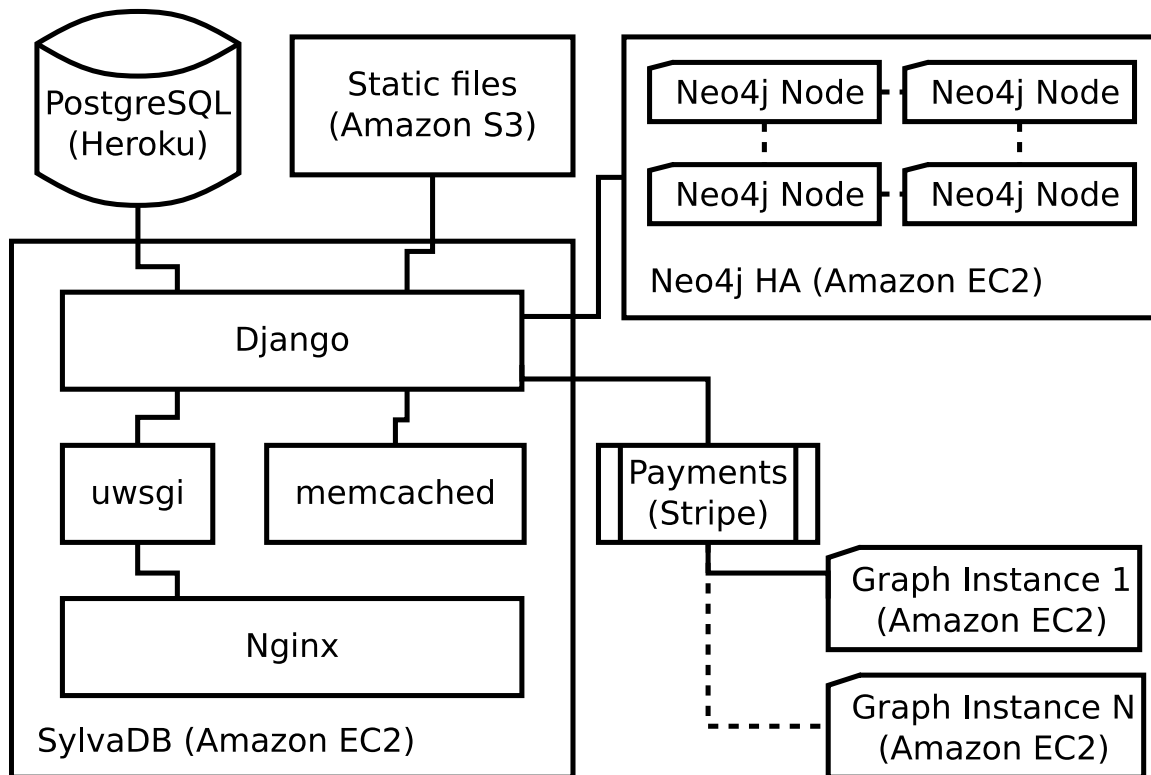


Fig. 1. General Architecture

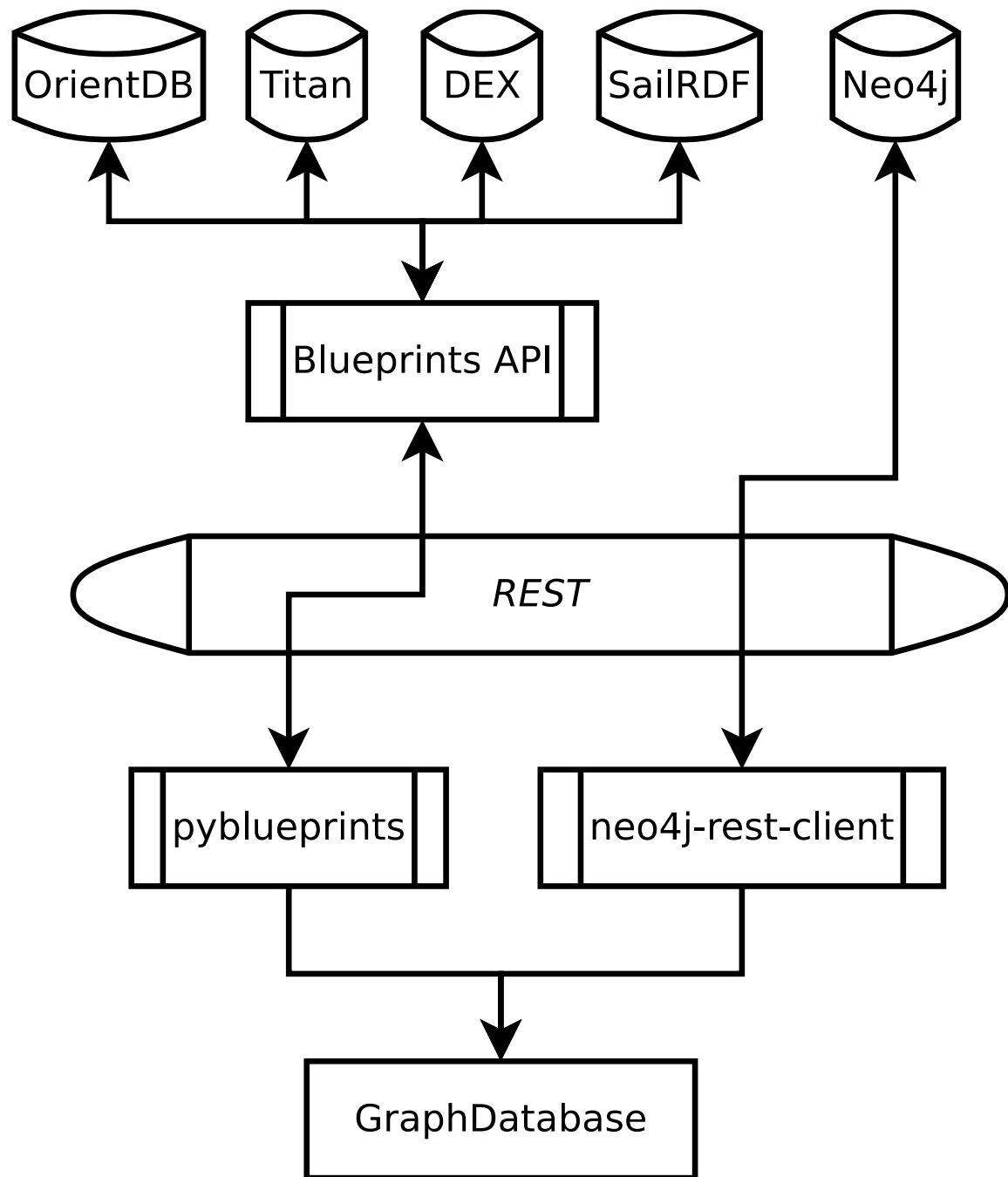


Fig. 2 Backend

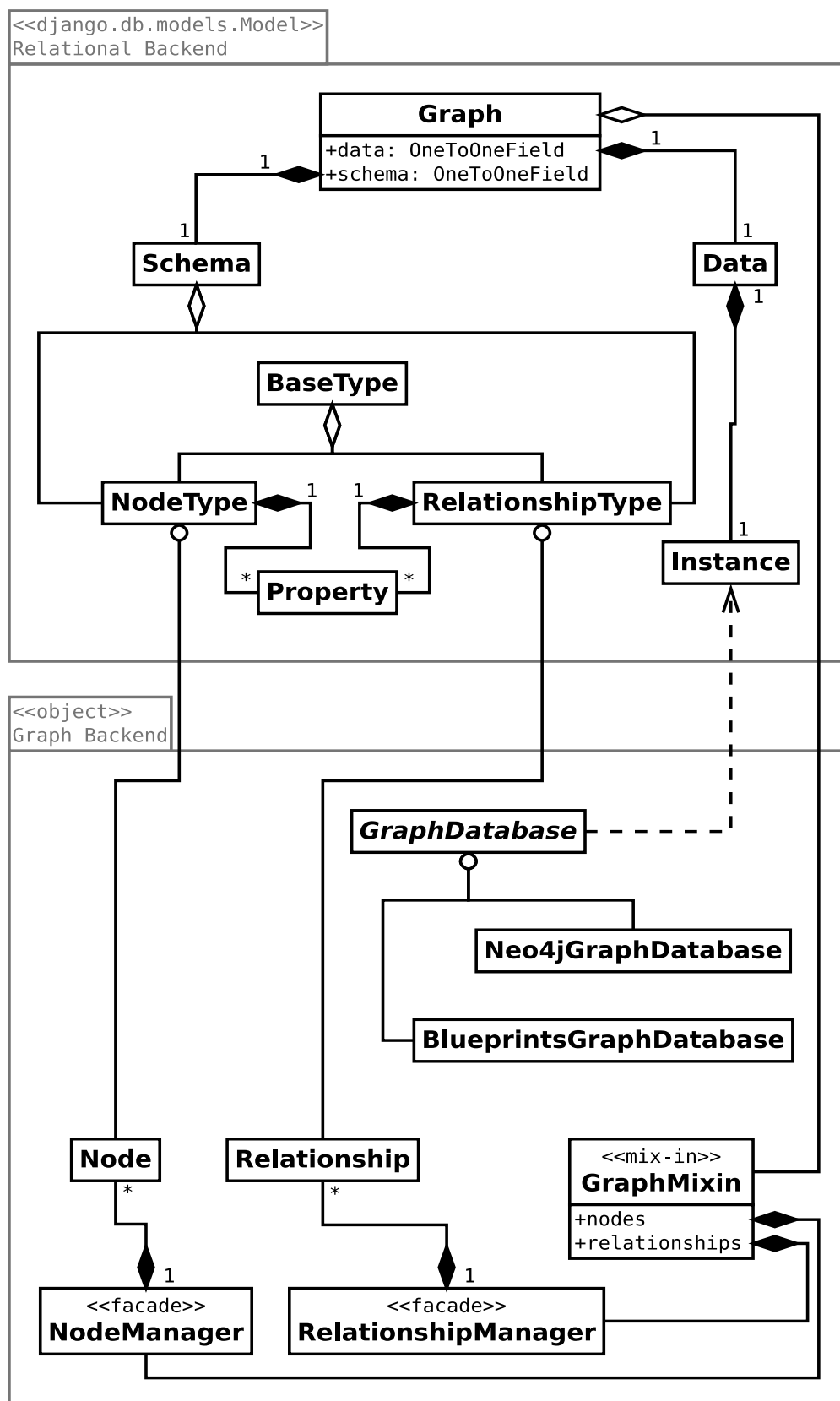


Fig. 3 Models

9. User Guide: Getting Started with SylvaDB

Getting started with SylvaDB

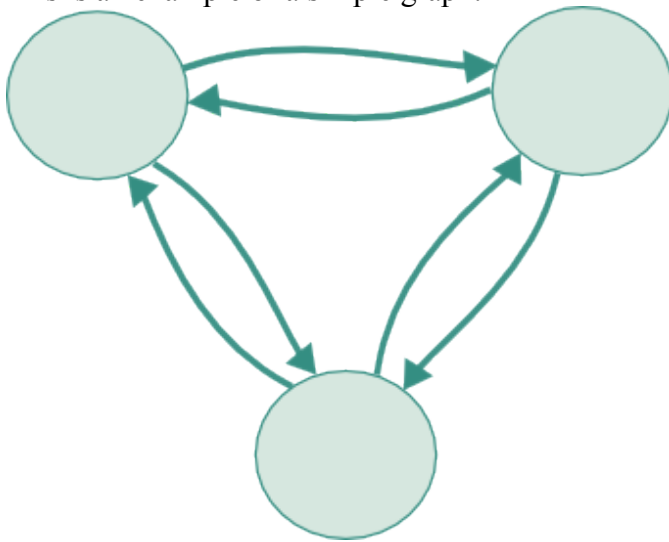
Introduction <http://testing.sylvadb.com/get-started/-introduction>

What is a graph database <http://testing.sylvadb.com/get-started/-what-is-a-graph-database>

A graph database stores data in a graph, the most generic of data structures, capable of elegantly representing any kind of data in a highly accessible way. Basically, a graph states that something is related to something else. It's a natural way of specifying relationships among a collection of items.

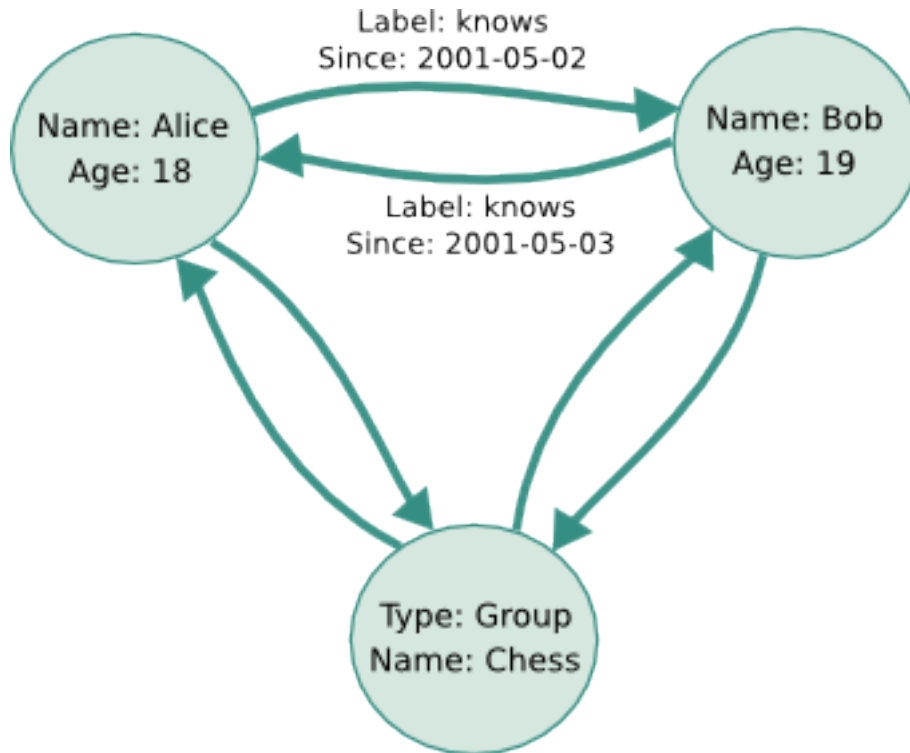
A graph uses **nodes**, **edges** and **properties** to represent the data. Typically, nodes are represented as circles and edges as arrows.

This is an example of a simple graph:



Nodes represent *entities* such as people, books, songs, or any other item you might want to keep track of. **Edges** represent the relationship between the nodes. And properties are pertinent information that relate to nodes or relationships. But in the above graph there isn't any property... yet.

This is an example of a simple graph with some properties:



In this graph there are three nodes representing two kind of entities: the entity **Person** and the entity **Group**. As you can see, each node has a set of properties with information about the entity. Also, there are six relationships between the nodes, but only two relationships have properties. Easy!

A graph is a flexible and powerful tool for describing relations between elements of any kind. For this reason, graph databases are often faster than traditional databases for associative data sets with highly complex relationships. They can scale more naturally as they do not typically require expensive database operations. Also, they are suitable to manage ad-hoc and changing data as they depend less on a rigid *schema* to describe relations between elements.

Why use SylvaDB <http://testing.sylvadb.com/get-started/-why-use-sylvadb>

SylvaDB is an easy-to-use, flexible, and scalable graph database management system that helps you collect, collaborate, visualize and query large data sets. It provides all the benefits of using a graph database. And the best thing is that you will need no programming knowledge to use it.

An overview <http://testing.sylvadb.com/get-started/-an-overview>

The Dashboard <http://testing.sylvadb.com/get-started/-the-dashboard>

This is the panel where you manage your graphs in SylvaDB. It's divided in three columns: Graphs, Collaborations and Statistics.

Dashboard

Graphs

There're no graphs yet.



New Graph

Collaborations

There're no graphs in collaboration yet.

Statistics

There're no statistics yet.

In **Graphs** you will find all the graphs you have created. **Collaborations** is for all those graphs where you are contributing as a collaborator. And **Statistics** shows some statistics about your graphs.

We haven't created any graph yet... So, let's create a new one!

Designing the graph <http://testing.sylvadb.com/get-started/ - designing-the-graph>**Creating a graph** <http://testing.sylvadb.com/get-started/ - creating-a-graph>

We are going to create a graph about music. So, first of all, we need to choose a **name** for our graph. We will call it *My music collection*. It's useful to also provide a **description**, although it's not required. Finally, we choose if our graph will be **public** or not (publicly available for other people).

Create New Graph**Name:**

My music collection

Description:

Albums, concerts, bootleg recordings and videos from my favourite bands.

Is public? ☒*It means publicly available to be browsed***Create new graph** or **Cancel**.

After clicking on *Create new graph*, we will be redirected to the Dashboard, where we can see an overview of our new graph.

Dashboard**Graphs****My music collection***"Albums, concerts, bootleg recordings and videos from my favourite bands."*By **sergio**

0 nodes

0 relationships

Strict **schema**

Public

0 collaborators

No queries yet

0 bytes on default

Some useful information is showed in this overview. For example, you can see the number of *nodes* and *relationships* of the graph, also the number of *collaborators*, and other information that we will see later, as *Strict Schema*... What does it mean? Let's see!

Creating a schema <http://testing.sylvadb.com/get-started/ - creating-a-schema>

A schema is the place where you describe how is organized your data, the structure of your database. In a graph database you use the schema to define **Types**: the kind of data you can store in your nodes. Also, to define the **allowed relationships** between the nodes.

So, clicking on **Strict schema** you will see there isn't any Type defined yet. Let's create a new Type.

After clicking on *New Type*, you will see a form where you can add a **Name** and a **Description** for your Type.

[My music collection » Schema » Type](#)

Type

Name

Description

[Save Type](#) or [Cancel](#).

You should see Types as data *entities*. For example: a book, a person, an album, etc. Here we are creating a Type to represent an **Album** entity.

Then, we define the **properties** of the Type. We need to think what are the most relevant information about a music album. For example, we would store: *Title*, *Duration*, *Number of tracks* and *Release date*. Those are what we call the **keys**. So, clicking on

[+ Add Property](#) we can get something like this:

Properties

Key

Description

☐ Use as label

Key

Description

☐ Use as label

Key

Description

☐ Use as label

Key

Description

☒ Use as label

[+ Add Property](#) or [Advanced Mode](#).

You can see that we have marked the property *Title* as the **label** of this Type. This way, we will use this property to identify each node.

Save Type

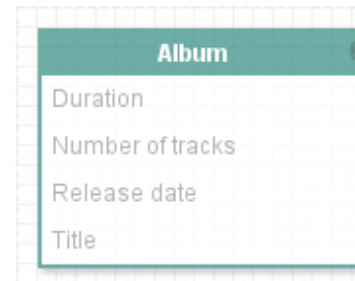
We finish clicking on **Save Type**. Then you will see an overview of our schema:

[My music collection » Schema](#)

Album

- △ **Duration**
 - △ **Number of tracks**
 - △ **Release date**
 - △ **Title**
- New **incoming** or **outgoing** allowed relationship

Diagram



Also, we want to create the Type **Artist**, so move to the bottom of the page and click *New Type*.

New Type

New allowed Relationship

We will define three properties for the Type **Artist**: *Active*, *Name* and *Size*. Note that *Name* will be the label.

Properties

Key	Description	Remove
Active	Is this artist still active?	
<input type="checkbox"/> Use as label		
Name		
<input checked="" type="checkbox"/> Use as label		
Size	Number of members	
<input type="checkbox"/> Use as label		

+ Add Property or **Advanced Mode**.

By default, properties are defined as *strings*, but sometimes we'll want to restrict the kind of data we would want to store: a *number*, a *date*, a *boolean*, ... In those cases, we can define the *type* of a property using the **Advanced Mode** tool. So, clicking on *Advanced Mode*, you will see a set of fields for each property:

Properties

Key	Data type	Default value	Order	Remove
Active	Boolean			
<input type="checkbox"/> Use as label <input type="checkbox"/> Is required?				
Description Is this artist still active?				
Name	Default			
<input checked="" type="checkbox"/> Use as label <input type="checkbox"/> Is required?				
Description 				
Size	Number			
<input type="checkbox"/> Use as label <input type="checkbox"/> Is required?				
Description Number of members				

[+ Add Property](#) or [Regular Mode](#).

We are interested in using just the **Data type** field. So, clicking on that field, you will see a list of property types. Now select the type *Boolean* for the property *Active* and *Number* for the property *Size*. This way, we are limiting the possible values of the *Active* property to be just *True* or *False* (visualized as a check box), and the possible values of the *Size* property to be just *numbers*, avoiding possible errors and mistakes when storing data.

Save Type

We finish clicking on

[My music collection » Schema](#)

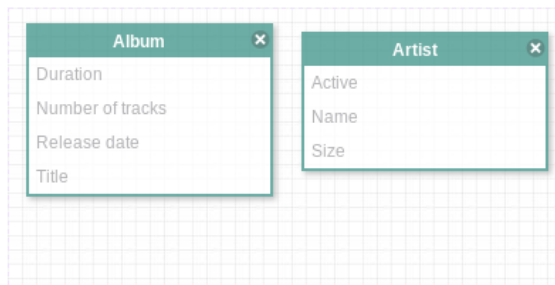
Album

- △ Duration
 - △ Number of tracks
 - △ Release date
 - △ Title
- New incoming or outgoing allowed relationship

Artist

- △ Active [Boolean]
 - △ Name
 - △ Size [Number]
- New incoming or outgoing allowed relationship

Diagram



So, it's time to change the Type **Album**, updating the types of its properties using the *Advanced Mode* tool. This time, we want to use the type *Date* for the property *Release Date* and the type *Number* for the property *Number of tracks*.

Key	Data type	Default value
Number of tracks	Number ▼	
<input type="checkbox"/> Use as label		
<input type="checkbox"/> Is required?		
Description		

Key	Data type	Default value
Release date	Date ▼	
<input type="checkbox"/> Use as label		
<input type="checkbox"/> Is required?		
Description		

Finally, repeating the steps to create also the Type **Concert**, we get the following schema:

[My music collection » Schema](#)

Album

- △ **Duration**
- △ **Number of tracks** [Number]
- △ **Release date** [Date]
- △ **Title**
- New incoming or outgoing allowed relationship

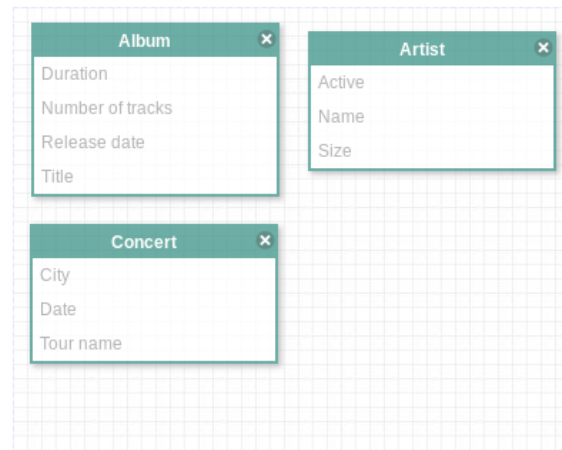
Artist

- △ **Active** [Boolean]
- △ **Name**
- △ **Size** [Number]
- New incoming or outgoing allowed relationship

Concert

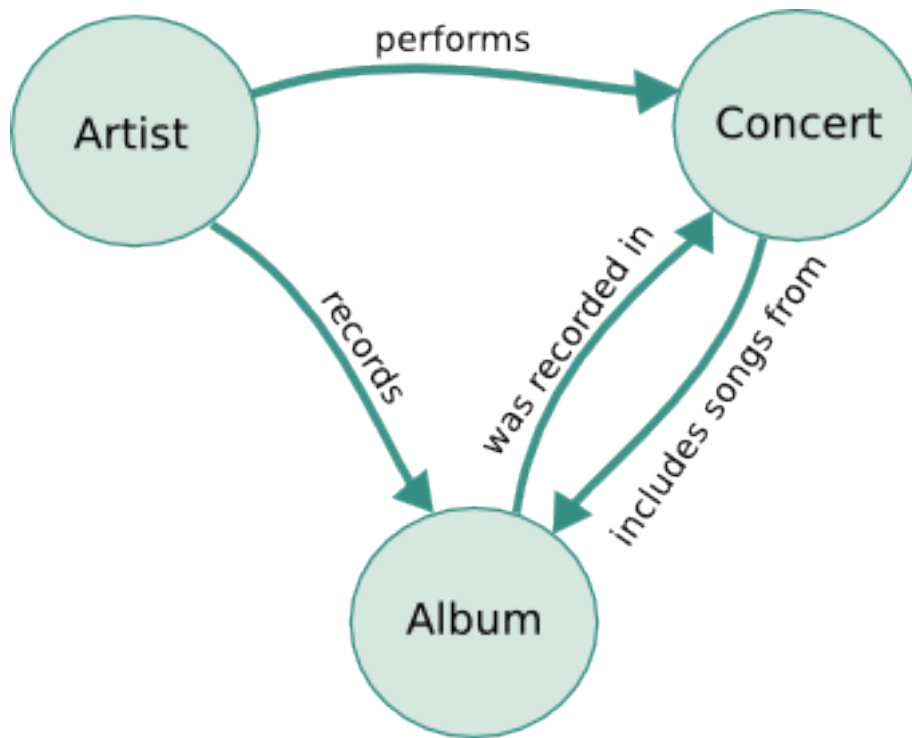
- △ **City**
- △ **Date** [Date]
- △ **Tour name**
- New incoming or outgoing allowed relationship

Diagram



We have defined three Types in our schema: **Artist**, **Album** and **Concert**, so we can already create nodes in our graph using these Types. But, this way, our graph will contain only nodes without any relation between them. And we think there is some sort of connection between these three entities, so, we need to also define **Relationships** in our schema.

Here you can see the relationships we want to create:



Note that we have two different relationships between *Album* and *Concert*. We start defining the relationship between *Artist* and *Album*. So, in the *Artist* section from the schema overview we saw before, we click on **outgoing**:

Artist

△ **Active** [Boolean]

△ **Name**

△ **Size** [Number]

New incoming or outgoing allowed relationship

Then, we select *Album* as the **target** of the relationship:

Allowed Relationship

Source

Source type of the allowed relationship

Name

Target

Target type of the allowed relationship

Description

Save Type Remove or Cancel.

We also want to add some property to this relationship. We want to store the *Date*:

Properties

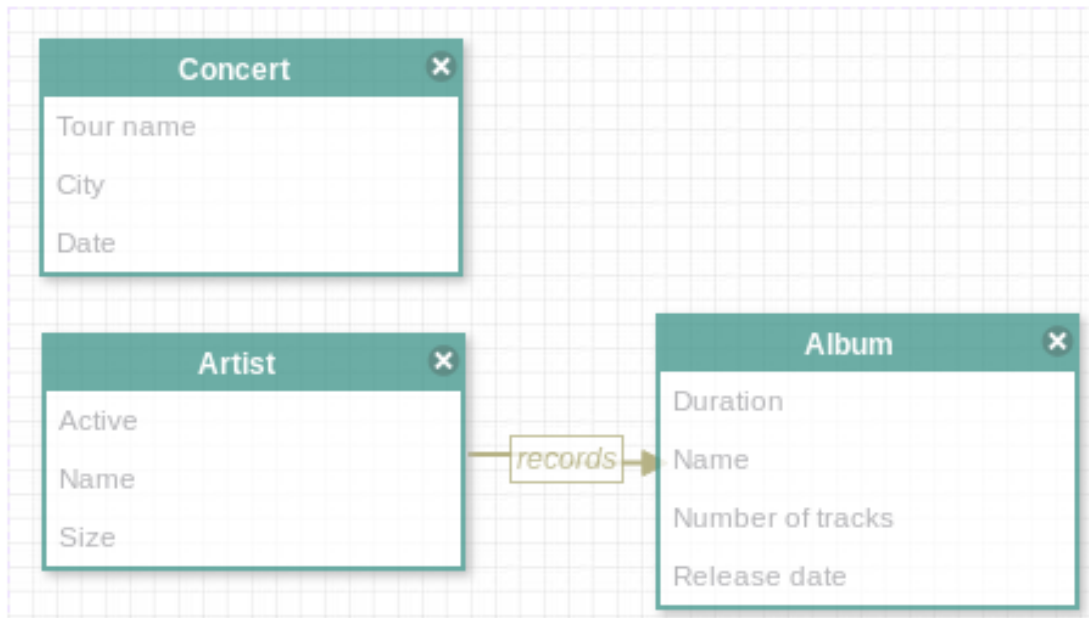
Key	Data type	Default value	Order	Remove
<input type="text" value="Date"/>	<input type="text" value="Date"/>	<input type="text"/>	<input type="text"/>	

☒ Use as label
☐ Is required?

Description

As you can see, we have marked the property *Date* as the **label** of the relationship. This way, we will use this property to identify each relationship when we visualize the graph. After clicking *Save Type*, you will see that a new relationship has been created in the Schema diagram (you can drag the boxes to see the schema better):

Diagram



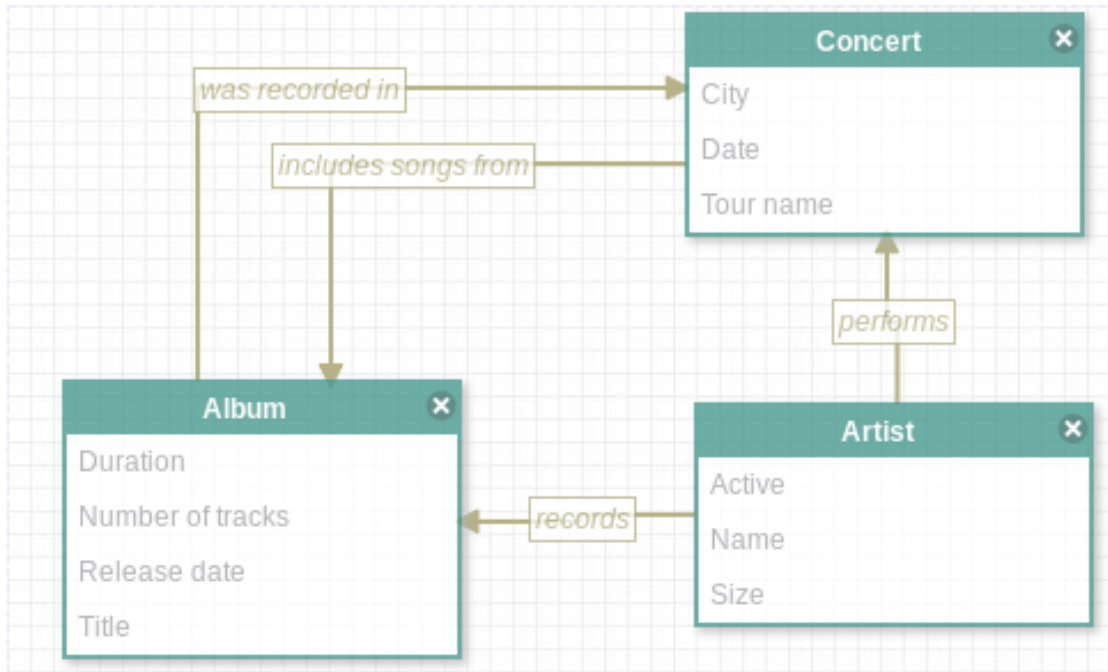
When you create a relationship in the schema between two Types, you will not see the links *incoming* nor *outgoing* anymore. So, you will need to use the link **New allowed Relationship** at the bottom of the page if you want to create more relationships using either of these Types.

New Type

New allowed Relationship

This time, we don't want to define any property in these relationships. So, repeating the process to create the rest of relationships, finally we get our database schema:

Diagram

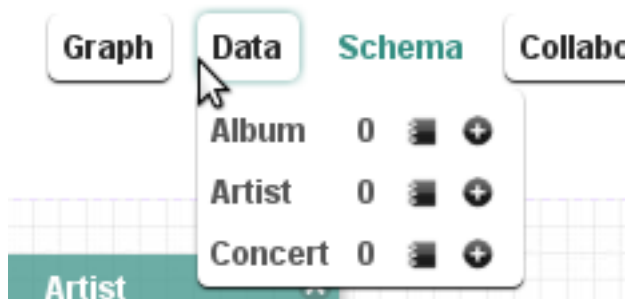


We have created a schema for the *Music Collection* graph. But, we haven't added any data yet, so in the next topics we will see how to add some **nodes** and **relationships** to the graph.

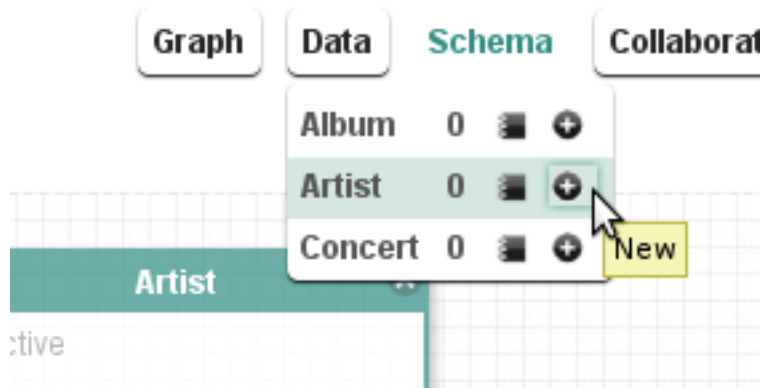
Adding data <http://testing.sylvadb.com/get-started/-adding-data>

Creating nodes <http://testing.sylvadb.com/get-started/-creating-nodes>

In order to store some data in our graph, we need to create *nodes*. So, move to the top menu at the right corner of the screen:



And add a new *Artist* clicking on the '+' symbol:



You will see a form to fill in with data about the Artist. This form is composed by four sections:

- Properties
- Relationships
- Files
- Links

The **Properties** section represent the properties we defined in the schema for the Type **Artist**. For example, we will add information about *Led Zeppelin*:

Properties

Active ☐

Is this artist still active?

Name

Led Zeppelin

Size

4

Number of members

Save Artist or **Cancel**.

We will see the rest of form sections later. So, let's save the *artist* clicking on the *save* button. You will see a data summary with the nodes for this *entity*. Currently, this list only shows one node.

You can change the order of properties listed using the field *Order* from the *Advanced Mode* tool we saw before in the Schema definition. So, let's return to the schema definition for **Artist** and activate the *Advanced Mode* tool. You simply use numbers to indicate the order:

Properties

Key	Data type	Default value	Order	Remove
Active <input type="checkbox"/> Use as label <input type="checkbox"/> Is required?	Boolean ▼		3	
Description Is this artist still active?				
Name <input checked="" type="checkbox"/> Use as label <input type="checkbox"/> Is required?	Default ▼		1	
Description (empty)				
Size <input type="checkbox"/> Use as label <input type="checkbox"/> Is required?	Number ▼		2	
Description Number of members				

[+ Add Property](#) or [Regular Mode](#).

Now the properties are listed in the order we set:

Properties

Name

Led Zeppelin

Size

4

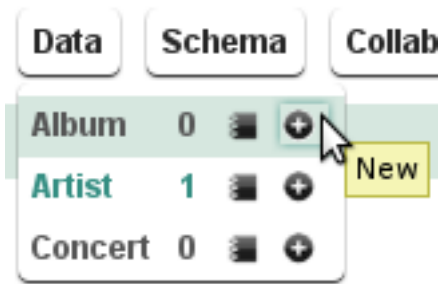
Number of members

Active ☐

Is this artist still active?

[Save Artist](#) or [Cancel](#).

Let's add some information about an **Album**. Again, move to the top menu and click on the '+' icon:



We repeat the same process filling in the properties for *Album*.

Properties

Title

The title of the album

Duration

Number of tracks

Release date

(yyyy-mm-dd)

or **Cancel.**

We also add some information about a **Concert**:

Properties

Tour name

City

Date

(yyyy-mm-dd)

or **Cancel.**

We have just created three nodes in our graph for the types **Artist**, **Album** and **Concert**.
Creating relationships [http://testing.sylvadb.com/get-started/ - creating-relationships](http://testing.sylvadb.com/get-started/-creating-relationships)

After saving the **Concert**, you will see a data summary with the nodes for this *entity*. Currently, this list only shows one node.

[My music collection](#) » [Data](#) » [Concert](#)

[Graph](#) [Data](#) [Schema](#) [Coll](#)

#	Tour name	City	Date
	Celebration Day	London, England, United Kingdom	2007-12-10

Showing 1 elements Concert.

Since we haven't added any relationship yet, clicking on the icon will not show any relationship. So, let's add a relationship.

Clicking on the icon the form for **Concert** will be showed again, but this time, we just want to add relationships:

All Relationships

→ **includes songs from (Album)**

Target of the relationship [Add Album](#)



+ Add includes songs from (Album)

← **performs (Artist)**

Source of the relationship [Add Artist](#)



[Toggle properties.](#)

+ Add performs

← **was recorded in (Album)**

Source of the relationship [Add Album](#)



+ Add was recorded in

Note that the little arrows show you the direction of the relationship. Also, you can see enclosed by parenthesis the *type* of the *node* at the **source** or **target** of the relationship. Adding a *target* or *source* to each relationship is really easy. Just start typing the *label* property of the node (remember you marked the label checkbox in some properties), and you will show a list with results matching the criteria.

All Relationships

→ **includes songs from (Album)**

Target of the relationship [Add Album](#)



+ **Add includes songs from (Album)**

← **performs (Artist)**

Source of the relationship [Add Artist](#)



[Toggle properties.](#)

+ **Add performs**

← **was recorded in (Album)**

Source of the relationship [Add Album](#)



+ **Add was recorded in**

We don't need to fill in the relationship **was recorded in**, so we will remove it clicking



on

Finally, we finish clicking on **Save Concert**. We can repeat the same process with the rest of nodes and relationships.

Adding multimedia <http://testing.sylvadb.com/get-started/-adding-multimedia>

In the previous form you can also add **links** and **files** to your node:

Files

Label:



File:

 No se ha sele...ingún archivo

+ **Add File**

Links

Label:



URL:

+ **Add Link**

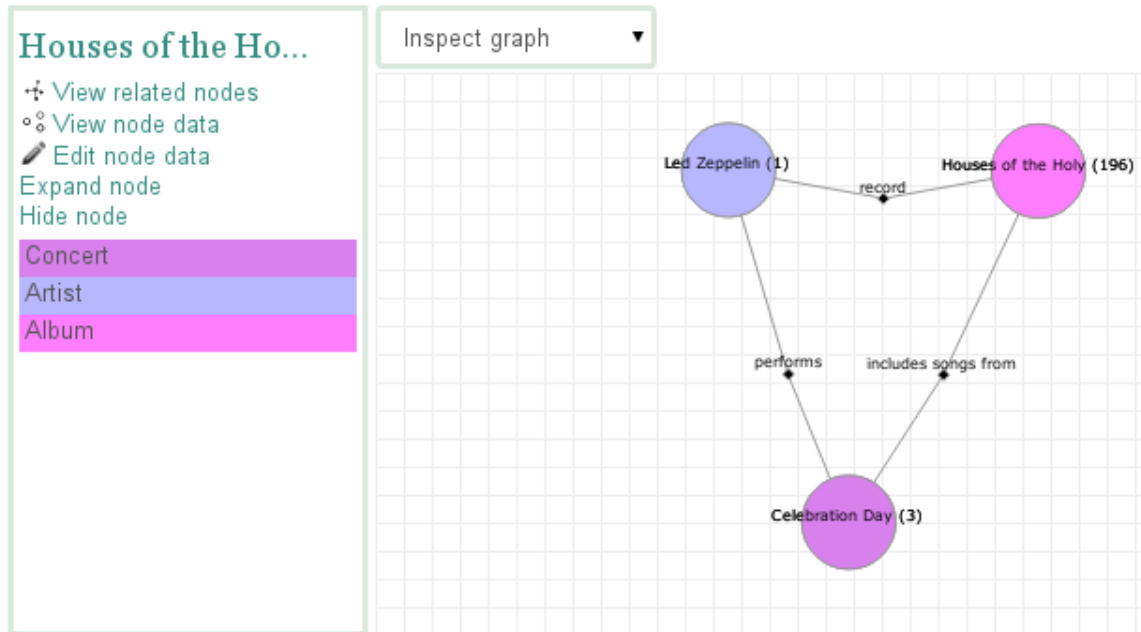
Visualizing the graph <http://testing.sylvadb.com/get-started/-visualizing-the-graph>

Graph visualization <http://testing.sylvadb.com/get-started/-graph-visualization>

Clicking on **Graph** from the top menu, you will visualize the graph. You can choose two kind of views from the select box: **inspect graph** and **whole graph**.

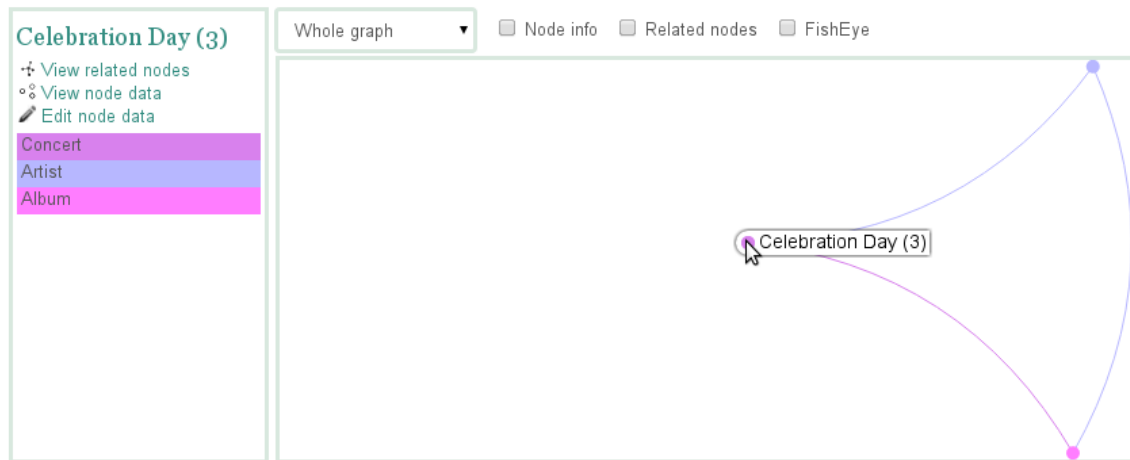
With **inspect graph** view, you can get a more detailed view of your graph. This view is useful for small graphs:

My music collection



The **whole graph** view is the best option when you want to visualize big graphs (graphs with more than 50 nodes). It also provides some useful tools to improve the analysis and visualization of your graph:

My music collection



Exporting graphs <http://testing.sylvadb.com/get-started/-exporting-graphs>

You can back up your graphs using the *Export tools*. In order to export a graph, you need to save the **graph schema** and the **graph data** (nodes and relationships). You have two ways of saving your graph data: using [GEXF \(Gephi\)](#) or using [CSV](#) as file formats.

Unfortunately, you can't save your attachments nor links at the moment.



Exporting the schema <http://testing.sylvadb.com/get-started/-exporting-schema>

You can save your graph schema using the *Export schema* tool. This tool will save your graph schema as a [JSON](#) file.

Exporting data as GEXF (Gephi) <http://testing.sylvadb.com/get-started/-exporting-gexf>

This is the preferred way of saving your graph data in SylvaDB. This tool will save your nodes and relationships as a [GEXF \(Gephi\)](#) file.

Exporting data as CSV <http://testing.sylvadb.com/get-started/-exporting-csv>

You can also save your graph data as [CSV](#) files. This tool will save your nodes and relationships in separate CSV files (one file for each node *type* and relationship *type*).

Importing graphs <http://testing.sylvadb.com/get-started/-importing-graphs>

You can import graphs in SylvaDB using the *Import tools*. You have to import the **graph schema** and optionally the **graph data**. If you want to import data, you can use two file formats: [GEXF \(Gephi\)](#) and [CSV](#). Note that you need an empty graph in order to import the schema and data.

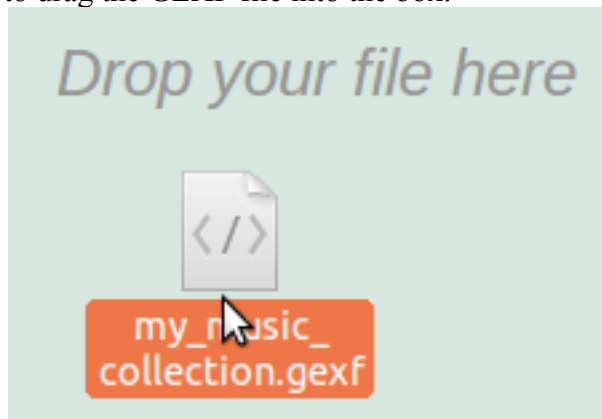
Importing the schema <http://testing.sylvadb.com/get-started/-importing-schema>

After creating a new graph, open the *Tools* menu and click *Import schema*. Then select the [JSON](#) file that holds your graph schema.



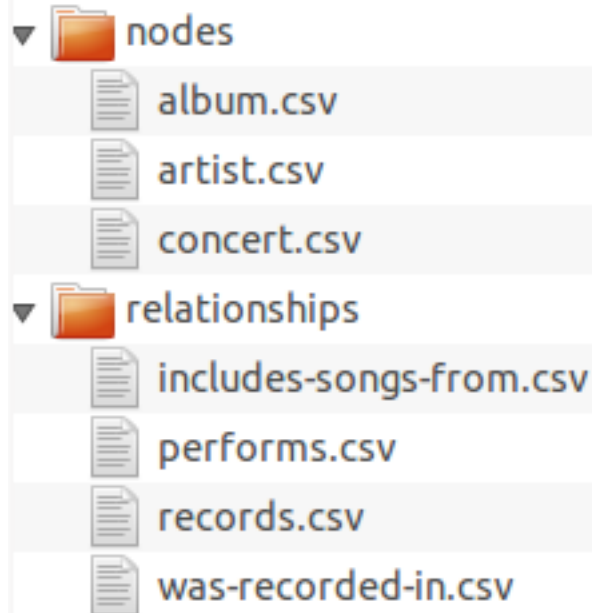
Importing from GEXF (Gephi) <http://testing.sylvadb.com/get-started/-importing-gexf>

This is the preferred way of loading nodes and relationships into your graphs. Just move to the *Tools* menu and click *Import data*. Then select *Load GEXF*. Finally, you only have to drag the GEXF file into the box.



Importing from CSV <http://testing.sylvadb.com/get-started/-importing-csv>

You can also load nodes and relationships from [CSV](#) files. Move to the *Tools* menu and click *Import data*. Then select *Load CSV*. Finally, select all your nodes (CSV files) and drag them into the box. Wait for instructions and repeat the same operation with your relationships. Remember that you need one CSV file for each node *type* and relationship *type*.



Editing CSV files <http://testing.sylvadb.com/get-started/-csv-spreadsheet>

A [CSV](#) file contains any number of **rows**, where each row consists of **fields** separated by **commas** and enclosed by **double quotes** (this is mandatory in SylvaDB). A field can contain line breaks, commas and escaped double quotes (represented as a pair of double quotes). The first row of a CSV is the **header** of the CSV and contains a list of field names. The other rows are data, and in SylvaDB they represent **nodes** or **relationships**, depending on the CSV file.

This is an example of CSV for the node *type* **Artist**:

```
"id", "type", "Active", "Name", "Size"          "1", "Artist", "False",
"Led Zeppelin", "4"
```

This is an example of CSV for the node *type* **Album**:

```
"id", "type", "Duration", "Name", "Number of tracks", "Release date"
"2", "Album", "40:58", "Houses of the Holy", "8", "1973-3-28"          "3", "Album",
"1:22:15", "Physical Graffiti", "15", "1975-2-24"
```

This is an example of CSV for the relationship *type* **Record**:

```
"source id", "target id", "label", "Date"      "1", "2", "records",
"1972-8-1"          "1", "3", "records", "1973-11-1"
```

As you can see in the examples, in the case of nodes, the CSV header must contain the field names (in this order): **id**, **type** and all the property names of the node *type*. In the case of relationships, the CSV header must contain the field names (in this order): **source id**, **target id**, **label** and all the property names of the relationship *type*. Note that **source id** and **target id** are the IDs of the *source node* and *target node* of the relationship, respectively.

Since a [CSV](#) file is just a [plain text](#) file, you can open it with any simple text editor (please, avoid Microsoft Word-like programs). Although the most common way for dealing with CSV files is using a spreadsheet software like [LibreOffice Calc](#) or [Microsoft Excel](#). In the next example, we will use LibreOffice Calc.

Try opening the example for the node *type* **Album** with LibreOffice Calc. You will see a window like the following:

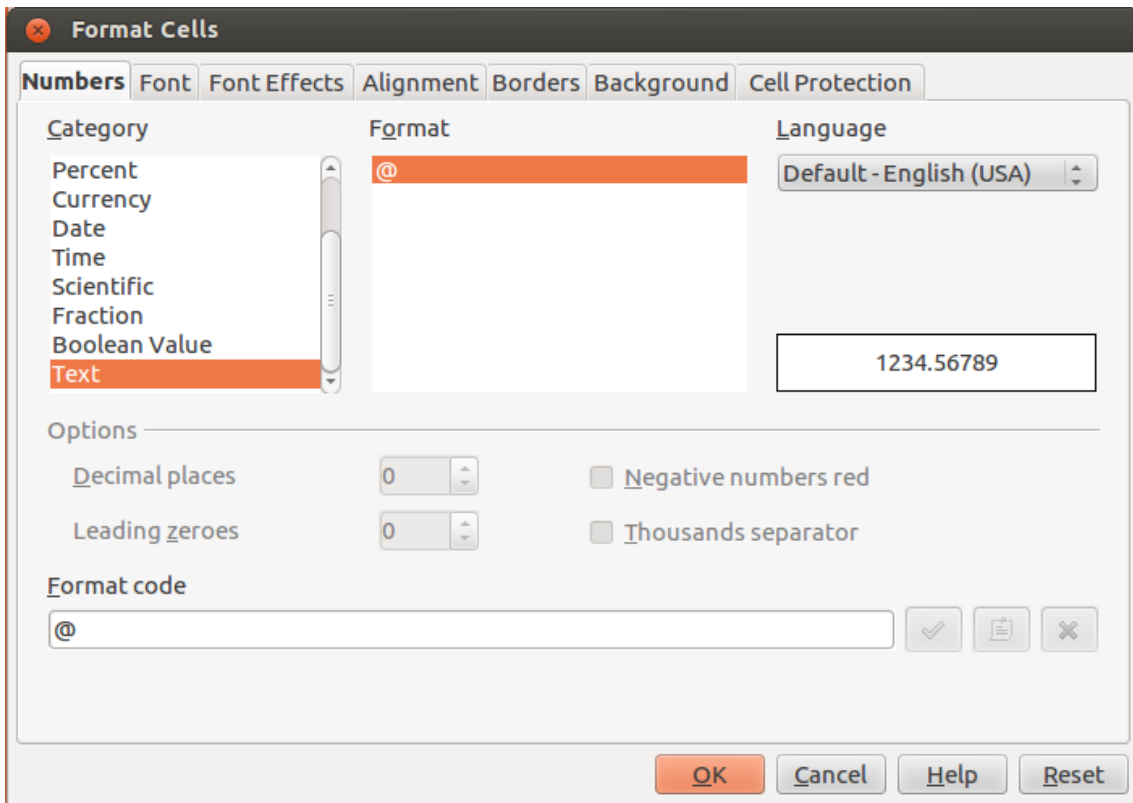
The screenshot shows the 'Import' dialog box in LibreOffice Calc. The 'Import' section has 'Character set' set to 'Unicode (UTF-8)', 'Language' set to 'Default - English (USA)', and 'From row' set to '1'. The 'Separator options' section has 'Separated by' selected, with 'Comma' checked and 'Text delimiter' set to '"'. The 'Other options' section has 'Quoted field as text' checked. The 'Fields' section shows a preview table with columns: id, type, Duration, Name, Number of tracks, Release date. The preview data is: Row 1: id, type, Duration, Name, Number of tracks, Release date; Row 2: 2, Album, 40:58, Houses of the Holy, 8, 1973-3-28; Row 3: 3, Album, 1:22:15, Physical Graffiti, 15, 1975-2-24.

Note that you must select the same options as shown in the "Separator options" and "Other options" sections from the above window. After clicking "OK", you will get your CSV imported:

	A	B	C	D	E	F
1	id	type	Duration	Name	Number of tracks	Release date
2	2	Album	40:58	Houses of the Holy	8	1973-3-28
3	3	Album	1:22:15	Physical Graffiti	15	1975-2-24

Remember that in SylvaDB is mandatory to have every field enclosed by double quotes. This way, all the fields are interpreted as text, even the numbers. Note that fields are enclosed by double quotes only after saving the spreadsheet as a CSV file. So, you will not see the fields enclosed by double quotes while working in the spreadsheet.

So, before adding data to the CSV, we must configure this sheet in order to process the numbers as text fields. Select all the rows (go to "Edit" menu and click "Select All"). Then open the "Format Cells" menu (go to "Format menu" and click "Cells..."). Finally, in the "Category" section, select "text":



After clicking "OK", numbers will be processed as text, and enclosed by double quotes when saving the spreadsheet as a CSV file.

We are adding a few more albums to this spreadsheet:

	A	B	C	D	E	F
1	id	type	Duration	Name	Number of tracks	Release date
2	2	Album	40:58	Houses of the Holy	8	1973-3-28
3	3	Album	1:22:15	Physical Graffiti	15	1975-2-24
4	4	Album	44:25	Presence	7	1976-4-31
5	5	Album	33:04	Coda	12	1982-11-19

Finally save the spreadsheet as "Text CSV" format.

Now, if you open this CSV file with a simple text editor, you will see something like the following:

```
"id","type","Duration","Name","Number of tracks","Release date"
"2","Album","40:58","Houses of the Holy","8","1973-3-28"
"3","Album","1:22:15","Physical Graffiti","15","1975-2-24"
"4","Album","44:25","Presence","7","1976-4-31"
"5","Album","33:04","Coda","12","1982-11-19"
```

You can see that all the fields have been enclosed by double quotes, even the ID numbers.

Also, if you have some embedded double quote literal inside any field, LibreOffice Calc will escape it using a pair of double quotes, so you don't have to handle this by hand.

If you plan to add some data to a blank graph using CSV files, you can get a set of *CSV templates* using the *Export CSV templates* tool. Note that this tool is only available in graphs with no data (just the schema):



10. Competition Comparison

For a graphic of the main players offering products and services in the space of NoSQL databases, please, see document attached as.

11. Testing Assumptions (Survey of features and needs)

The survey has been created in www.surveymonkey.com and it comprises 15 questions that address issues related to the users, the technology and the market potential of the proposed product. The list of questions is included below. A full version of the survey appears under Annex 5.

Questions:

1. Which sector is your company in?
2. Have you used Graph Databases before?
3. Do you know how different Graph Databases are from traditional SQL databases?
4. Graph Databases can help you manage your linked data. Do you work with highly linked data?
5. In your company/organization, are there specialists devoted to data management and analysis?
6. SylvaDB is a graph database management system that allows you to manage your data and database and you do not need to know any programming. Would this help you in your work?
7. Usually data schemes (the way you structure data) change over time, would a tool to easily change your data scheme be helpful for you?
8. To get insights from your linked data you normally do complex queries. SylvaDB offers a query system to do queries in natural language: is this an important feature for you?
9. Visualizations are an important tool for Big Data. Which of these features would you prefer to visualize your data? Please, order according to their importance.
10. Are pre-configured algorithmic measures important for your job? Please, rank the following according to your needs.
11. What range of storage better fit your data needs?
12. Which of these advanced features (machines) are more important for you? Please, order them according to their importance.
13. Which of these advanced features (system) are more important for you? Please, order them according to their importance.
14. How many collaborators work in your graphs/databases at the same time?
15. Would you like to use any other feature in your graph database?

12. Lean Canvas

The lean canvas method is a standard methodology employed in start-ups to rapidly prototype a business plan without incurring in high costs or getting into unrealistic financial estimates. It is a high level view of the possible business that help grasp in a single shot the different aspects of the business.

Product 1 - Prueba 1

PROBLEM Data schema modification Access to Big Data with no programming skills Natural language queries	SOLUTION Use SylvaDB data schema management tool SylvaDB requires basic knowledge of the problem domain and no programming skills Formulate and save your queries	UNIQUE VALUE PROPOSITION SylvaDB opens the access to Big Data to everyone who does not have programming skills HIGH-LEVEL CONCEPT SylvaDB is the BMW of Big Data at the price of a Toyota	UNFAIR ADVANTAGE Data schema management and interface (patent pending)	CUSTOMER SEGMENTS Researchers in several areas: banking, marketing, medicine, ... Companies and research institutions that want to give their researchers a way to capitalize on Big Data
EXISTING ALTERNATIVES Hire a programmer Hire a programmer Use SQL in relational databases (or hire a programmer)	KEY METRICS Number of clients Cash burn out ratio Revenue		CHANNELS Direct sales Web site Academic and professional meetings	EARLY ADOPTERS Colleagues at the faculty of Medicine at University?
COST STRUCTURE Programmers Designers Marketing team Servers Office space Administration			REVENUE STREAMS Individual subscriptions Service packages for institutions and companies Licensing	

Lean Canvas is adapted from The Business Model Canvas (BusinessModelGeneration.com) and is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.

13. Marketing Plan

13.1. Industry

Graph databases belong to the group of databases known as NoSQL, that is, databases that store and retrieve data that is modeled in means other than the tabular form of relational databases^{xv}. The reasons to use this approach have to do with simplicity of design, horizontal scaling, and connectedness of data. When data express many relationships with many complex properties as we see in social networks, relational databases are not the most effective way to deal with it. Graph data is made of nodes and edges –things and connections among things– and both nodes and edges can be assigned properties for easy storage and querying that does not lose the richness of the data. A graph database can be easily constructed from a schema drawn by hand, as depicted in this article^{xvi}.

Graph databases technology is expanding due to businesses' attempts to take advantage of the amounts of data that we are producing as a result of the revolution of the digital age. These data are characterized by volume and velocity of production, variety and complexity of data, and connectedness. These trends^{xvii} make graph databases a perfect tool to extract all value from complex and connected data related to marketing, social networks, security, transportation and many other business areas that take advantage of these type of rich data to increase efficiency and accuracy in their analysis and projections^{xviii}.

The worldwide market size for NoSQL databases is estimated to reach \$3.4 Billion by 2018 at a CAGR of 21% between 2013 and 2018 and will generate \$14 Billion in revenues over the period 2013 – 2018^{xix}. Thanks to the successful adoption of this type of technology by Facebook, Digg, and Twitter, and the development of applications in biotechnology, defense and social networking analysis it is expected that NoSQL, and specifically graph databases, turn into a source of competitive advantage in most industries that need to extract meaningful and actionable information from complex data.

At this point, the main players in the market of graph databases are Neo4J, Allegro, and Titan, among others.

13.2 The Marketing Challenge

The marketing challenge is to design a marketing plan that clearly defines the value proposition, identifies a proper market niche, positions SylvaDB in a space that makes it competitive in the long run, and creates a price structure that makes SylvaDB attractive to potential customers. In sum, the challenge is to devise a marketing plan that helps steer a startup into a viable company that adds value to its customers in a market —data analysis of complex data sets– that it is in its infancy and, therefore, not fully understood by all stakeholders.

13.3 Situation Analysis

As in the case of many startups^{xx}, SylvaDB has not received any investment to develop the product into a viable business. However, it received research funds to create a prototype that is now fully functional and open to the public in a free, beta version. As it is now, the product is totally usable and only the completion of the natural language

query system (plus minor design improvements) is scheduled in the next few months. The objective is to bootstrap the initial phases of business development through direct access to customers; the downside of this is that there is not a budget for marketing nor is there a sales force to push the product into the market.

SylvaDB does not aim at competing with the large players providing full deployment of graph databases. SylvaDB competes by providing an interface for the main functionalities needed to perform operations on large and complex (very connected and with many properties) datasets such as data modeling, data imports and exports, visualizations, and data queries. SylvaDB is designed so that users will never need a programmer, analyst or computer science specialist. As long as the user knows well her domain and is able to map that domain into the easy-to-use data modeling tool that SylvaDB provides, she can take full advantage of our product's tools to extract all the value from her data.

SylvaDB has a patent pending in the US and Canada, what gives an important technology protection and a competitive advantage.

13.4 Marketing Objectives

	Short-term 0-3 months	Mid-term 4-9 months	Long-term 9 months +
Marketing research	Market research report commissioned to MaRS	Targeting and segmenting: Direct interviews with managers in health care, marketing, and research in selected markets: Canada, Spain	Prepare database of companies and institutions worldwide with the size and needs that fit our service
Positioning and differentiation	Clearly define and word the value proposition in web site	Prepare sales and marketing supporting documents for pitches and correspondence with potential customers	After first few customers and market reassessment, refine the positioning and write technological and business plan for further differentiation
Branding	Define the logo, feeling and text of web site to reflect values of: research excellence, high efficiency, and best customer service Showcase partnership with	Establish list of best practices and behaviors by all team members in direct relations with customers	Loyalty strategy

	Neo4J		
Sales and distribution	Maximize the web site as marketing tool Attend specialized conferences Use freemium tier option to spread the word and as teaser for fully paid services	Develop full range of offered services with real data (Health sector) and reports to showcase potential to customers	Campaign of direct sales visits to selected potential customers. Establish clear objectives in terms of client type, scope of project, revenue, and branding
Go-to-market strategy		Secure a pilot project as test for all business aspects and to showcase it as a success story for future sales	Identify key sector and clients for 24 months plans Channels: web site, conferences and direct sales
Pricing		Establish pricing strategy for: Design and maintenance Data analysis Full deployment of instances Establish service/product mix and pricing	Establish promotion strategy for special clients or new, desirable clients/sectors

Table 2 Marketing Objectives / Phases

13.5 Alternatives to Achieve Objectives

There are two alternative routes to effectively implement a business and marketing plan for SylvaDB:

- Create a stand-alone product –platform as service– in the cloud with several price tiers that users will use out of the box. The price model here will be a basic mark up on usage and storage, plus a special fee to unlock all functionalities (for instance, limit the number of databases, relations, nodes, and data model changes that an user can implement for free). There would be a special tier for special clients who want to have the full system installed in their servers.
- Create a consulting service that leverages the flexibility of SylvaDB, especially the easiness provided to modify data models, and that would offer the following:
 - Design and maintenance of data sets: preparation of databases with variations on models of the same data sets, set the permits and access levels for team members and collaborative environments, provide basic visualizations, and a set number of pre-programmed queries for users explore further the data sets.

- Analysis of complex data sets and ensuing reports.
- Full deployment of SylvaDB system into the servers of the client, with add on technical and consulting services.

13.6 Recommendation

I favor pursuing option #2: creation of a consulting service that leverages the flexibility and functionality of SylvaDB. The reasons for this recommendation are:

- Lower startup costs, especially in marketing, sales, and servers infrastructure.
- Interviews with experts in the field (Neo4J top managers) have indicated that SylvaDB should exploit its most distinctive feature, the flexibility and easiness to change data models, and try to commercialize that. These experts have also recommended not to compete in the platform as service space because this translates into transferring the costs of renting Amazon servers (or other provider's) to the customers, not adding any value from our product, and being exposed to changes in the provider's policy and to war prices with other competitors.
- Interviews with experts in IT services for business have indicated that the average manager would not be willing to perform their own analysis even if the tools are easy to use and that in most cases they prefer to receive a short report summarizing the conclusions of the analysis.
- This option adds value from two sides: the technology itself and the knowledge that an expert group of mathematicians, big data analysts, and computer scientists bring to the various phases of the service. This intersection will help evolve both the technology and the business model as required.

13.7 Implementation Plan

The implementation plan comprises a set of actions aimed at determining in a 9-month window: the viability of business, the appropriateness of the value proposition, positioning, sales strategy, and pricing.

- Establishing main conclusions from commissioned market research report.
- Define and word value proposition in web site. Clearly define services and tiers.
- Redesign web site accordingly with least possible effort.
- Fully develop and showcase all services offered with example from health data sets.
- Promotion freemium tier as “free” marketing for the service.
- Create sales and pitch materials with all offered services and showcase results.
- Establish price structure based on value-in-use method for all services and possible combinations and promotions.
- Interviews/visits to 10 possible customers in areas of health and marketing.
- Slice the market, and likely focus on small to medium companies that are not willing or cannot afford the prices and size of services provided by big providers. Also, explore marketing and research departments with autonomy to budget part of their activities.
- Secure pilot project with real client.

Once these milestones have been accomplished and there is a positive outcome to the process, a new business plan will be designed that reevaluates seeking investment, rebuilding the team to reinforce key areas of sales, marketing, and technology, and refining a marketing plan that feeds from the data and knowledge acquired after the startup period, market trends and evolution, and effectiveness of the existing team.

13.8. Unique Value Proposition

SylvaDB opens the access to Big Data to anyone who knows the structure of the data she is working on and wants to explore different venues to analyze and visualize that data. This type of user will, in many cases, work in teams of non programming specialists, with individuals with diverse stakes in the project, and will reuse the same data, maybe with different schemes, several times. Sylva offers the flexibility both to change the scheme without affecting the integrity of the data and to allow collaboration among team members with different levels of access to the database. The fact that SylvaD works on top of Neo4J, a powerful graph database, adds to this flexibility as projects take on another dimension by using graph databases and taking advantage of the node-edge relation in storage, management, and querying the data.

13.9. Positioning

We will position SylvaDB as a B2B product that will capture the attention of analysts, marketers, researchers, and supply chain management specialists due to the ease of use and the autonomy it provides with in carrying out tasks related to the analysis of Big Data and semantic data. This positioning will take advantage of the unique value proposition that SylvaDB offers around data schema changes and data integrity as well as collaboration among teams of professionals. Thus, customization of data schema creation and evolution, deployment of special instances and tailoring of data queries to provide an out-of-the-box service to teams in specific business sectors will become the core marketing activity.

As for B2C, we will create a special account allowing low volume transactions, few databases, limited access to data schema changes, and no access to the advanced query system in order to entice new customers into trying SylvaDB and, later on, to purchase advance access.

13.10. Market Segmentation

Since SylvaDB is startup there is no record of clients or purchases that would allow for a market segmentation based on transaction patterns. Therefore, this market segmentation is completely programmatic and it will suffer changes in the months to come as a full go-to-market strategy is developed and the very reality of the market tells where the main needs and clients are.

Even so, the first segmentation comes from the distinction between direct clients (B2C) that will make use of SylvaDB from the web platform and will not require special attention or services from the business unit of the company, and those especially targeted for large projects (B2B). In this second case, the main segmentation at this point in time

comes from the legal nature of the clients, that is, between public organizations and private companies. Public or semi-public organizations requiring graph database management systems to solve specific problems in data management are in the sector of transportation (town halls and transport companies or corporations with multiple transport systems). In the private sector, the marketing team will focus on pharma and health research companies that need internal teams to develop autonomy to perform research plans as well as marketing groups that need to develop customized clustering analysis for loyalty and retention and need strong experimentation tools (different semantic data schemas) and powerful querying and visualization.

13.11. Pricing Strategy

As the main volume of business will deal with specific projects and clients varying in volume and scope, the main pricing strategy for this cases will derive from mark up pricing of 30% on total costs budgeted in the business proposal, including infrastructure costs in the cases this is needed, and with a special section devoted to service and maintenance depending on the needs of the clients.

The pricing strategy for the segment of clients using SylvaDB directly from the web platform will be based on a step strategy (three price categories) that will control the most valuable functionalities (data schema) and will make sure that no overheads are produced for an excess of computing processing or storage.

14. Projected Profit & Losses Statement

0. Hypothesis

Capital	\$4,000.00		
Incorporation costs	\$1,200.00		
Salaries	\$30,000.00		
Rent	\$600.00		
General expenses	\$1,800.00		
Project/hour price	\$540.00	Average project duration: 30	
Project/hour cost	\$270.00		
Payment to suppliers	90 days		
Inventory-WinProgress (as purchases days)	21 days		
Collections	60 days		
Taxes	20.00%		
Interest	8.00%		
Labor cost	30.00%		
Investments	Q1	Q2	Depreciation (years)
Computers	\$6,000.00		3

1. Projected P&L

	Q1	Q2	Q3	Q4
Projects contracted (in hours billed)	60	120	180	180
Sales	32,400.00 €	64,800.00 €	97,200.00 €	97,200.00 €
COGS	16,200.00 €	32,400.00 €	48,600.00 €	48,600.00 €
Gross Margin	16,200.00 €	32,400.00 €	48,600.00 €	48,600.00 €
Salaries	7,500.00 €	7,500.00 €	7,500.00 €	7,500.00 €
Corporate expense on labor	2,250.00 €	2,250.00 €	2,250.00 €	2,250.00 €
Rent	1,800.00 €	1,800.00 €	1,800.00 €	1,800.00 €
Incorporation costs	1,200.00 €			
General Operating Expenses	1,800.00 €	1,800.00 €	1,800.00 €	1,800.00 €
EBITDA	1,650.00 €	19,050.00 €	35,250.00 €	35,250.00 €
Computers depreciation	500.00 €	500.00 €	500.00 €	500.00 €
EBIT	1,150.00 €	18,550.00 €	34,750.00 €	34,750.00 €
EBT	1,150.00 €	18,550.00 €	34,750.00 €	34,750.00 €
Taxes	230.00 €	3,710.00 €	6,950.00 €	6,950.00 €

Net income	920.00 €	14,840.00 €	27,800.00 €	27,800.00
------------	----------	-------------	-------------	-----------

15. Conclusions

Developing a Business Plan for SylvaDB with special emphasis on marketing has been a very useful endeavor in order to gain an overall view of the technological innovation of the product, the market trends in databases and big data, the main players in this emerging economic area, the viable ways to position a new product in a new market, and the financial and production constraints involved in the creation of a start-up.

It has also been a tremendously interesting exercise in systematization and documentation, although these two virtues of the work can also become important hurdles as the angst to document everything can easily hide other objectives of the project such as developing an overall understanding about it or increasing the actionable knowledge on newly explored areas, such as the marketing process in this case.

Also, two main forces fight each other in a project of documentation related to a start up that is following the lean canvas method. On the one hand, documenting requires a very precise and intensive effort of attention to details and organization of information. On the other, the lean canvas method calls for a starting-up series of actions that reduce to the minimum the documentation, the code writing, and the financial estimates and reports, and favors testing the ideas with real clients out of the comfort zone of the office. Striking the appropriate balance between the two is a tricky proposition.

This Business Plan for SylvaDB has calibrated both conceptual frameworks in what has been a learning exercise in a moment of the project in which going from the prototype into the market requires a very precise analysis of the opportunities existing in this new market, and the risks of failing to positioning the product in the right niche for the right segment. This final project has helped a great deal to find this balance so that the project and the team involved solves the riddle of linking future rounds of software development and code writing to specific needs of identifiable and real customers and market segments. From this perspective, the experience has proven enriching and fulfilling.

Annex 1. Source Code
Available at <https://github.com/CulturePlex/Sylva>

Annex 1. Source Code
Available at <https://github.com/CulturePlex/Sylva>

Annex 2. Screen Shots

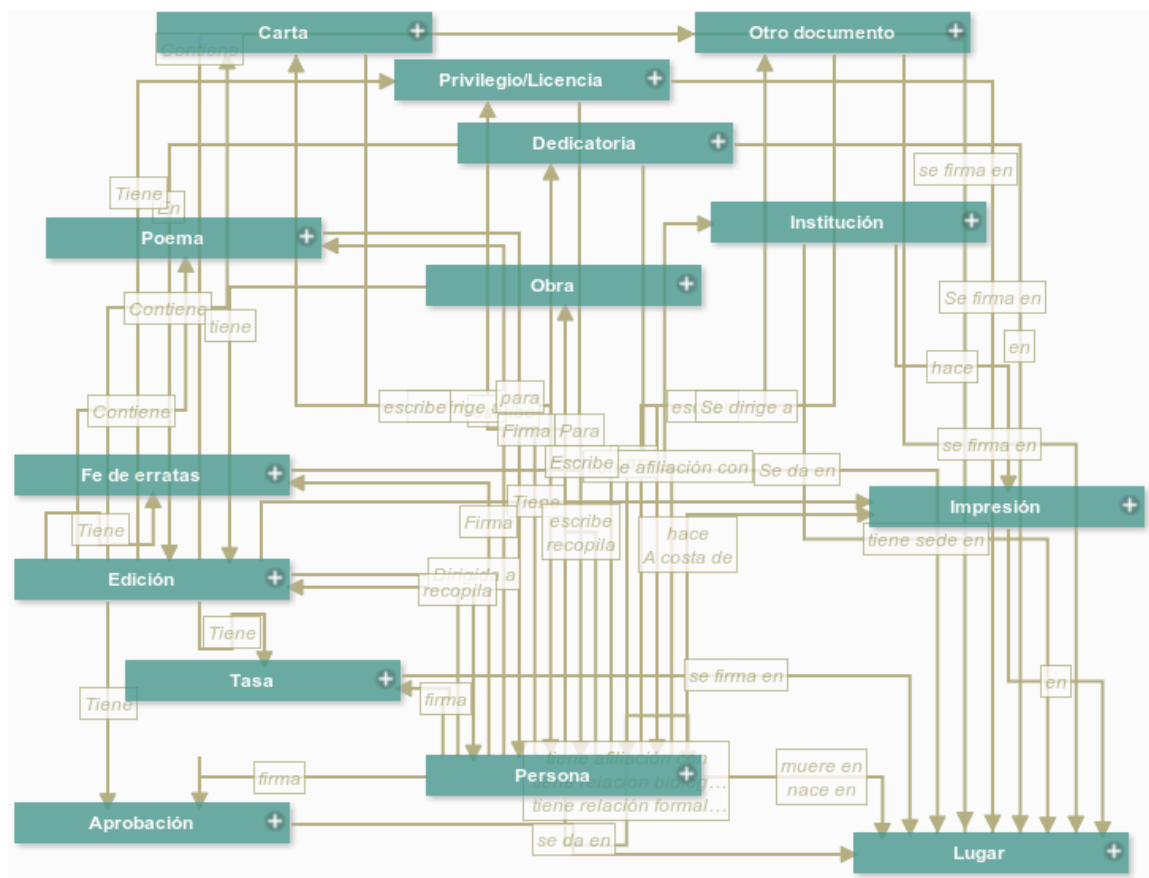


Fig. 4 Data Schema of a Project

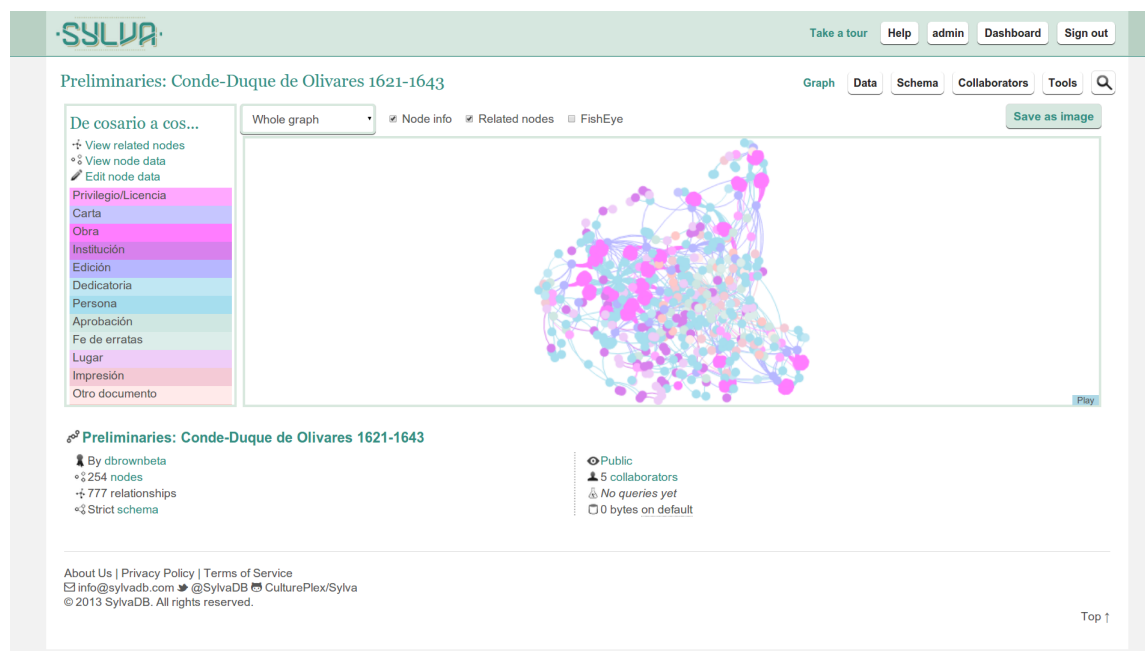


Fig. 5 Screenshot of a Project's Dashboard

Annex 3. Competition Comparison

See attached document

Annex 4. Executive Summary

This document summarizes the main aspects of SylvaDB, a graph database management system that is tailored to users with little or no knowledge of programming but whose professional activities require daily contact with messy data for the purposes of collecting, defining, curating, analyzing and visualizing the data. The focus of the document has been on highlighting the innovative aspects of the system as well as the unique value proposition that SylvaDB brings to the market following the Lean Canvas methodology of start-ups. Finally, the document explores several commercialization paths.

Annex 5. References

ⁱ See Appendixes for a graphic explaining the system.

ⁱⁱ leanstack.com

ⁱⁱⁱ businessmodelgeneration.com

^{iv} <http://steveblank.com>

^v Blank, Steve. *Four Steps to the Epiphany. Successful Strategies for Products that Win*. K&S Ranch Consulting, 2005. 3rd edition.

^{vi} Ulwik, Anthony. *What Customers Want. Using Outcome-Driven Innovation to Create Breakthrough Products and Services*. McGraw Hill, 2005.

^{vii} Ryan, Damian, Calvin Jones. *Understanding Digital Marketing. Marketing Strategies for Engaging the Digital Generation*. Kogan Page, 2012. 2nd edition.

^{viii} Kotler, Philip et al. *Marketing Management*. Pearson, 2012, 2nd ed.

^{ix} www.trello.com

^x De la Rosa, Javier, Juan Luis Suárez, Fernando Sancho. SylvaDB: A Polyglot and Multi-backend Graph Database Management System. *Proceedings of the 2nd International Conference on Data Technologies and Applications*, pages 285-292 DOI: 10.5220/0004475702850292. See Appendixes for details.

^{xi} <http://cultureplex.ca>

^{xii} <http://www.marsdd.com/>

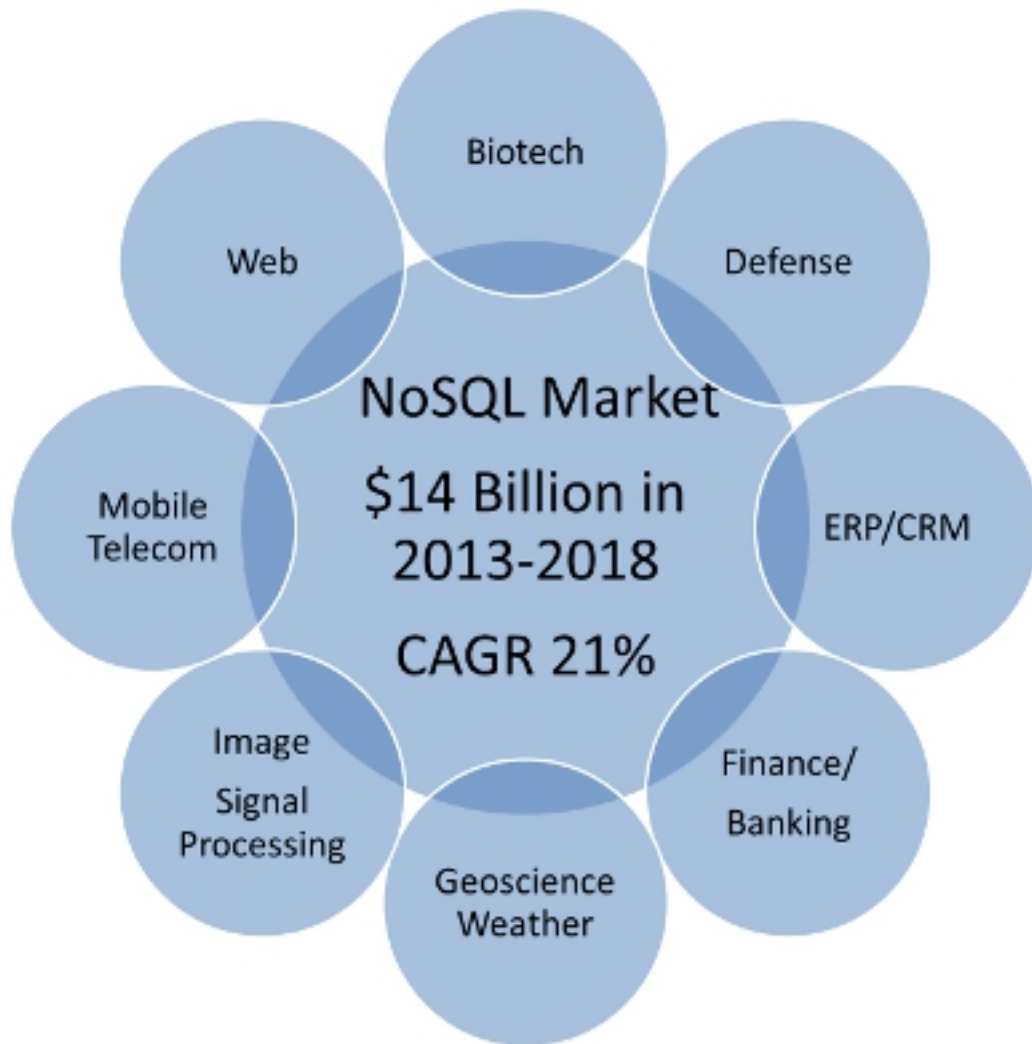
^{xiii} <http://www.techalliance.ca/>

^{xiv} <http://worlddiscoveries.ca/>

^{xv} <http://en.wikipedia.org/wiki/NoSQL>

^{xvi} <http://www.computerweekly.com/feature/Whiteboard-it-the-power-of-graph-databases>

^{xvii} <http://www.publictechnology.net/news/it-trends-gartners-2013-predictions-information-management/37616>



^{xviii} “Gartner recently named Strategic Big Data one of the top 10 strategic trends for 2013 and pointed to five specific types of data –all of them graphs– as the ‘five richest big data sources on the Web’: the social graph, intent graph, consumption graph, interest graph, and mobile graph. Add this to the strategic data sources inside one’s organization that are inherently graphs, and one becomes surprised at how pervasive graphs are.” “NoSQL, Big Data and Graphs” at www.neotechnology.com

^{xix} <http://www.marketresearchmedia.com/?p=568>

^{xx} Thomas Eisenmann. *Business Model Analysis for Entrepreneurs*. HBS 9-812-096.