



# Connecting Looker to Denodo

Revision 20180402

## NOTE

This document is confidential and proprietary of **Denodo Technologies**.  
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2023  
Denodo Technologies Proprietary and Confidential

## Goal

This document describes how to connect Looker to a Denodo Virtual DataPort (VDP) database and build a simple dashboard over the data gathered from the database.

## Content

[Looker](#) is a web-based platform that provides data exploration and discovery business intelligence features. Looker 3.50 or later versions are needed to be able to connect to Denodo VDP using the JDBC connector.

In order to support Looker's [Symmetric Aggregates](#), a custom function (BASE64\_TO\_BASE10) has to be added to the Denodo VDP Server. This function can be found within the 'Denodo Xtrafuncs for VDP' library in the Denodo Support Site, Denodo Connects section. The library must then be imported using the Jar management option in the File menu of the Denodo VDP Administration Tool.

## Creating a database connection to a VDP database

1. Go to the Looker instance URL from any web browser.
2. Enter the corresponding credentials to log in.
3. Click on the "Admin" link located in the upper right corner.
4. Go to "Database" > "Connections".
5. Click on "New Connection".
6. Enter the parameters for the new database connection.
  - a. Name.
  - b. Set Dialect to "Denodo".
  - c. Set the Host and Port corresponding to the VDP server to create the connection.
  - d. Set Database to the name of the database to connect to.
  - e. Set the Username and Password in order to connect to the corresponding VDP database. (Note that the user has to have SELECT permissions on all views that are going to be queried from Looker),
  - f. It is also possible to specify the value for several parameters as MaxConnections, Connection Pool Timeout and additional JDBC parameters.
7. Test the connection by clicking on "Test These Settings".
8. Once the connection is successfully tested, click on "Add Connection" to save it.

You are in **Development Mode**. Exit Development Mode

looker Browse Explore ▾ Develop ▾ Q Search ● Help ▾ Admin 👤 ▾

## New Connection

**General**

Settings

Labs

Legacy Features 5

**Users**

Users

Groups

Roles

Content Access

User Attributes

**Database**

**Connections**

Queries

Persistent Derived Tables

Datagroups

**Scheduler**

Plans

**Name \***  [View Documentation](#)

The name you will use to refer to this connection in your model.

**Dialect \***

**Host:Port \***

**Database \***

**Username \***

**Password**

**Max Connections**

Max number of connections Looker will allow at one time. Must be at least 5 and no more than 100.

**Connection Pool Timeout**

The number of seconds a query will wait before timing out due to a full connection pool. Must be greater than 90.

**Additional Params**

Additional JDBC connection string parameters (advanced).

Can connect

**View Documentation**

You are in **Development Mode**. Exit Development Mode

looker Browse Explore ▾ Develop ▾ Q Search ● Help ▾ Admin 👤 ▾

## Connections

**General**

Settings

Labs

Legacy Features 5

**Users**

Users

Groups

Roles

Content Access

User Attributes

**Database**

**Connections**

Queries

Persistent Derived Tables

Datagroups

**Scheduler**

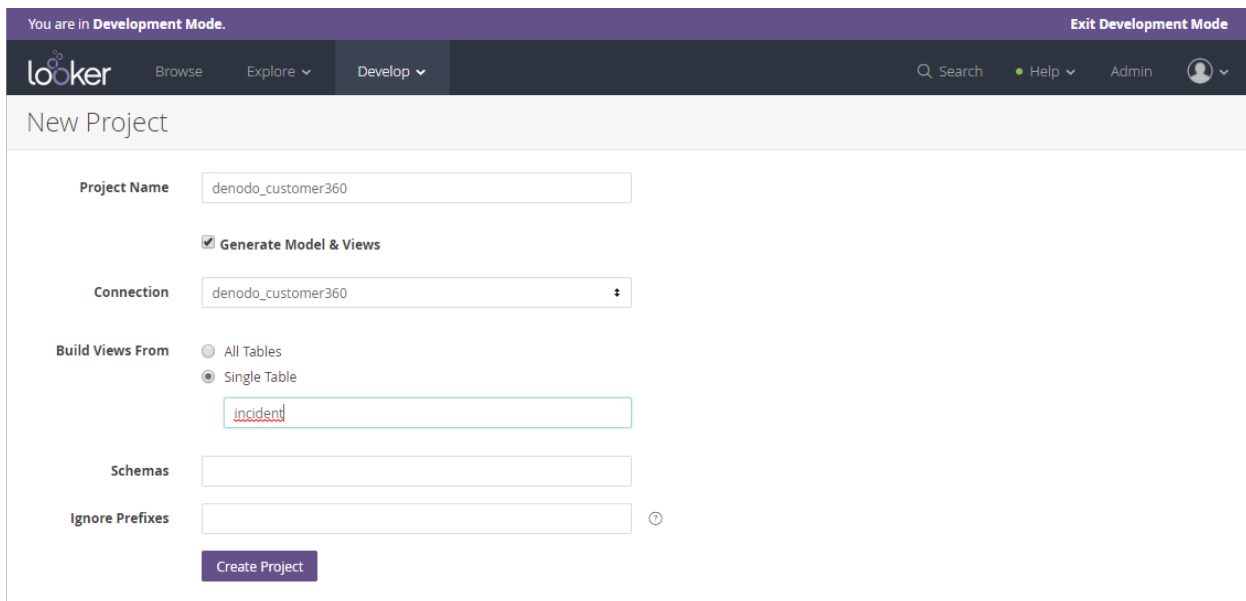
Plans

| Name                 | Host : Port | Username      | Database    | SSL | Type       | Actions  |
|----------------------|-------------|---------------|-------------|-----|------------|--|
| ● denodo_customer360 |             | admin         | customer360 | n/a | Denodo     | <input type="button" value="Test"/> <input type="button" value="Edit"/> <input type="button" value="⚙"/> |
| ● denodo_rodin_g_se  |             | jsmith        | customer360 | No  | PostgreSQL | <input type="button" value="Test"/> <input type="button" value="Edit"/> <input type="button" value="⚙"/> |
| ● thelook            |             | lookerpartner | demo_db     | No  | MySQL      | <input type="button" value="Test"/> <input type="button" value="Edit"/> <input type="button" value="⚙"/> |

## Creating a LookML project

Once the database connection is configured from Looker to the VDP database, it is possible to create a LookML project following these steps:

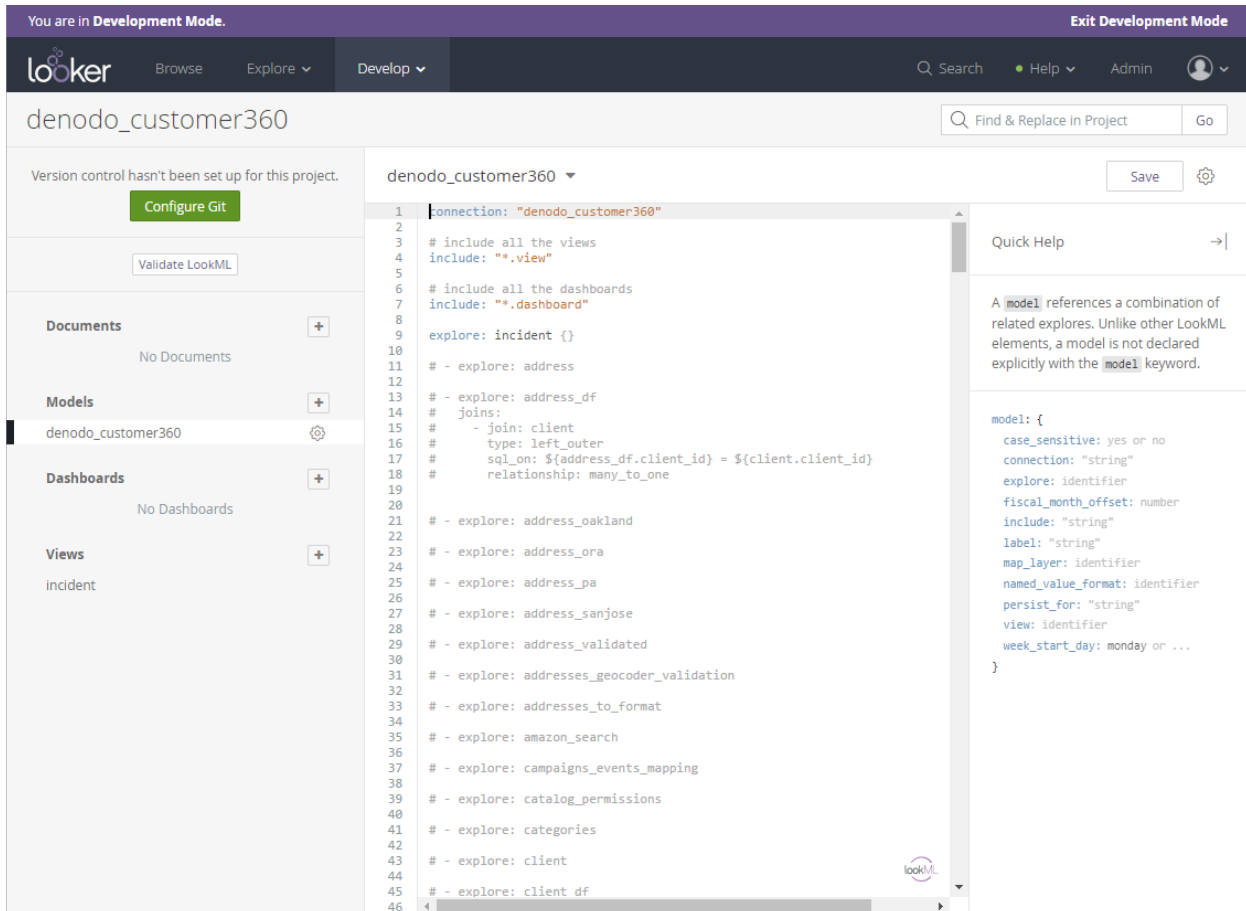
1. Click on “Develop” > “Manage LookML Projects”.
2. If not in Development mode, click on the “Develop” > “Development Mode” to toggle it to on.
3. Click on “New LookML Project”.
4. Enter the parameters for the LookML project.
  - a. Set the Project Name.
  - b. Set Connection to the connection created in the previous section.
  - c. We are going to build a project with a single view. Select the “Single table” option and enter an existing VDP view name as the Table Name.
    - i. Selecting the “Single table” option, the project will start adding only the table set up in the Table Name attribute.
    - ii. Selecting the “All tables” option, the project will start adding all the view in the database.
5. Click on “Create Project”.



The screenshot shows the 'New Project' configuration interface in Looker. At the top, it indicates 'You are in Development Mode.' and 'Exit Development Mode'. The navigation bar includes 'looker', 'Browse', 'Explore', and 'Develop'. The main content area is titled 'New Project' and contains the following fields and options:

- Project Name:** denodo\_customer360
- Generate Model & Views**
- Connection:** denodo\_customer360
- Build Views From:**
  - All Tables
  - Single Table
- Table Name:** incident
- Schemas:** (empty field)
- Ignore Prefixes:** (empty field)
- Create Project** button

After clicking on “Create Project”, Looker will generate the corresponding LookML project. Once this is done, the LookML development environment will show up.



The screenshot shows the Looker interface for a project named 'denodo\_customer360'. The main editor displays the following LookML code:

```

1 connection: "denodo_customer360"
2
3 # include all the views
4 include: "**.view"
5
6 # include all the dashboards
7 include: "**.dashboard"
8
9 explore: incident {}
10
11 # - explore: address
12
13 # - explore: address_df
14 #   joins:
15 #     - join: client
16 #       type: left_outer
17 #       sql_on: ${address_df.client_id} = ${client.client_id}
18 #       relationship: many_to_one
19
20 # - explore: address_oakland
21
22 # - explore: address_ora
23
24 # - explore: address_pa
25
26 # - explore: address_sanjose
27
28 # - explore: address_validated
29
30 # - explore: addresses_geocoder_validation
31
32 # - explore: addresses_to_format
33
34 # - explore: amazon_search
35
36 # - explore: campaigns_events_mapping
37
38 # - explore: catalog_permissions
39
40 # - explore: categories
41
42 # - explore: client
43
44 # - explore: client_df
45
46
    
```

The right-hand sidebar contains a 'Quick Help' section with the following text:

A `model` references a combination of related explores. Unlike other LookML elements, a model is not declared explicitly with the `model` keyword.

```

model: {
  case_sensitive: yes or no
  connection: "string"
  explore: identifier
  fiscal_month_offset: number
  include: "string"
  label: "string"
  map_layer: identifier
  named_value_format: identifier
  persist_for: "string"
  view: identifier
  week_start_day: monday or ...
}
    
```

## Components of a LookML Project

- **Project:** A directory containing LookML files that corresponds to a single database connection.
- **Explore:** An isolated query-building environment that appears on the Explore drop-down list in Looker. An explore is comprised of a base view and optionally several joined views.
- **Model:** A collection of explores, manifested as a single model file. Many models can live in the same project, and can share view files. Permissions can be set at a model level.
- **View:** A table of data. A view file points to a material or derived table of data and contains definitions of all the columns within, manifested as dimensions. A view file also contains definitions of measures (aggregates) built off of those dimensions.
- **Dimension:** An attribute (column) by which you can group/slice the data. A dimension can be material or derived.
- **Measure:** An aggregation over the data, which can be grouped by a dimension or set of dimensions. Examples are SUMs, COUNTs, MINs, AVGs, etc.

## Adding Explorers

As it can be seen in the last image showing the LookML of the previously created project, there is only one explorer, "incident". We have obtained just this one explorer due to the selection of the "Single table" option and setting "incident" as table name during the creation of this sample project.

To add more explorers to the project it is possible to manually modify the LookML content of the project in the same way it is shown in the comment lines of the LookML code.

- Adding a simple explorer for a single view:

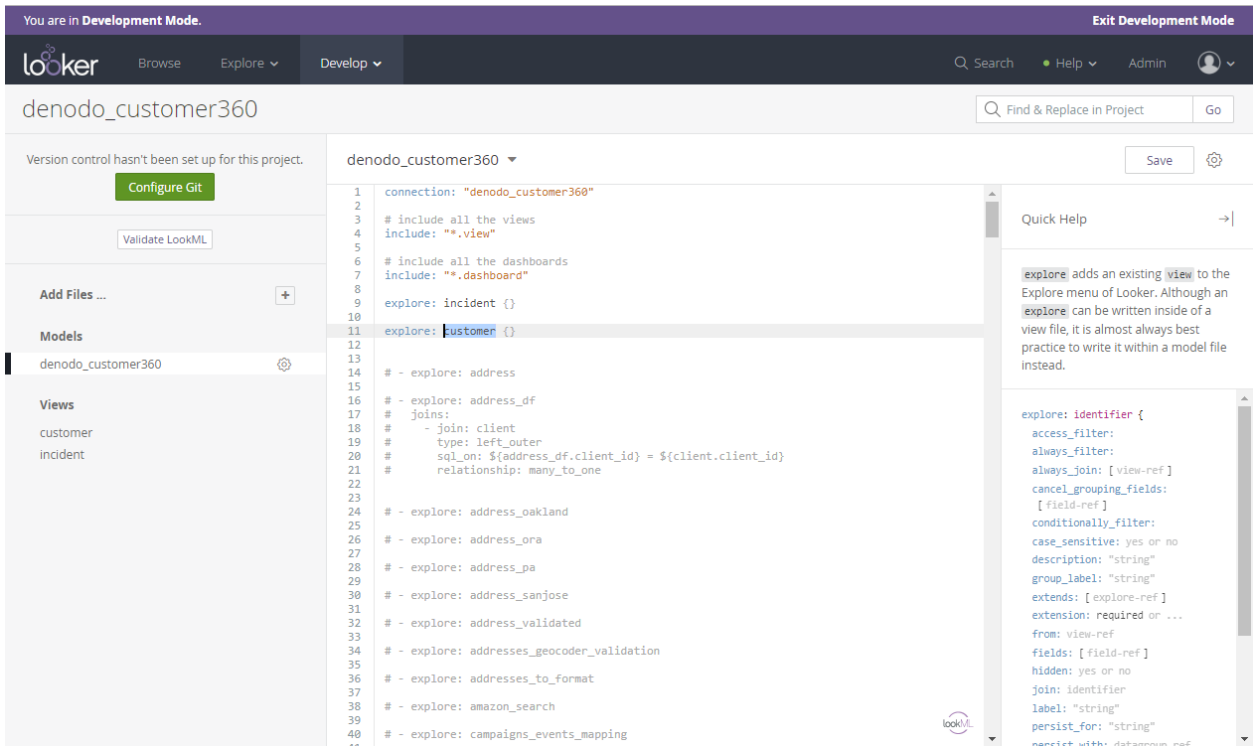
```
explore: <view_name> {}
```

- Adding an explorer joining views:

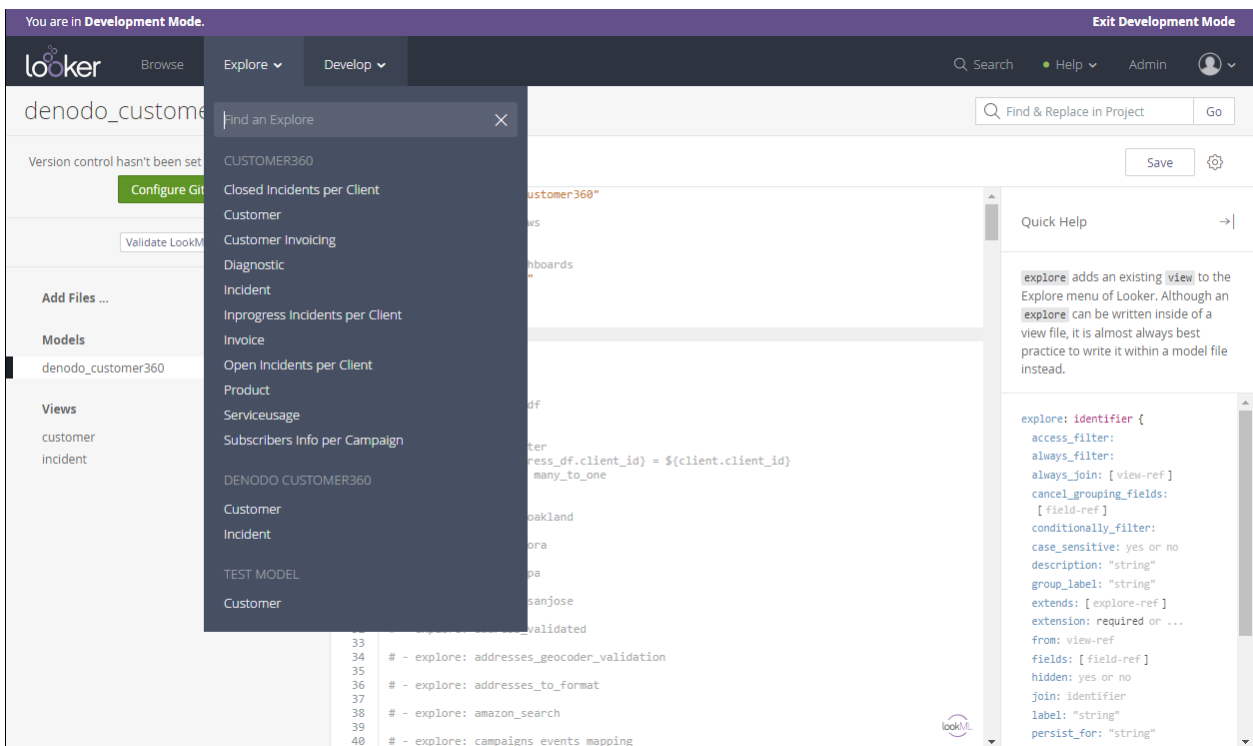
```
explore: <view_name_1> {  
  join: <view_name_2> {  
    # Desired join parameters  
    type: <join_type>  
    wql_on: <join_condition>  
    relationship: <relationship_between_views>  
  }  
}
```

After editing the LookML click on the "Save" button to the changes take effect.

To know more about how to define explorers take a look to the following article: [Explore Parameters](#).

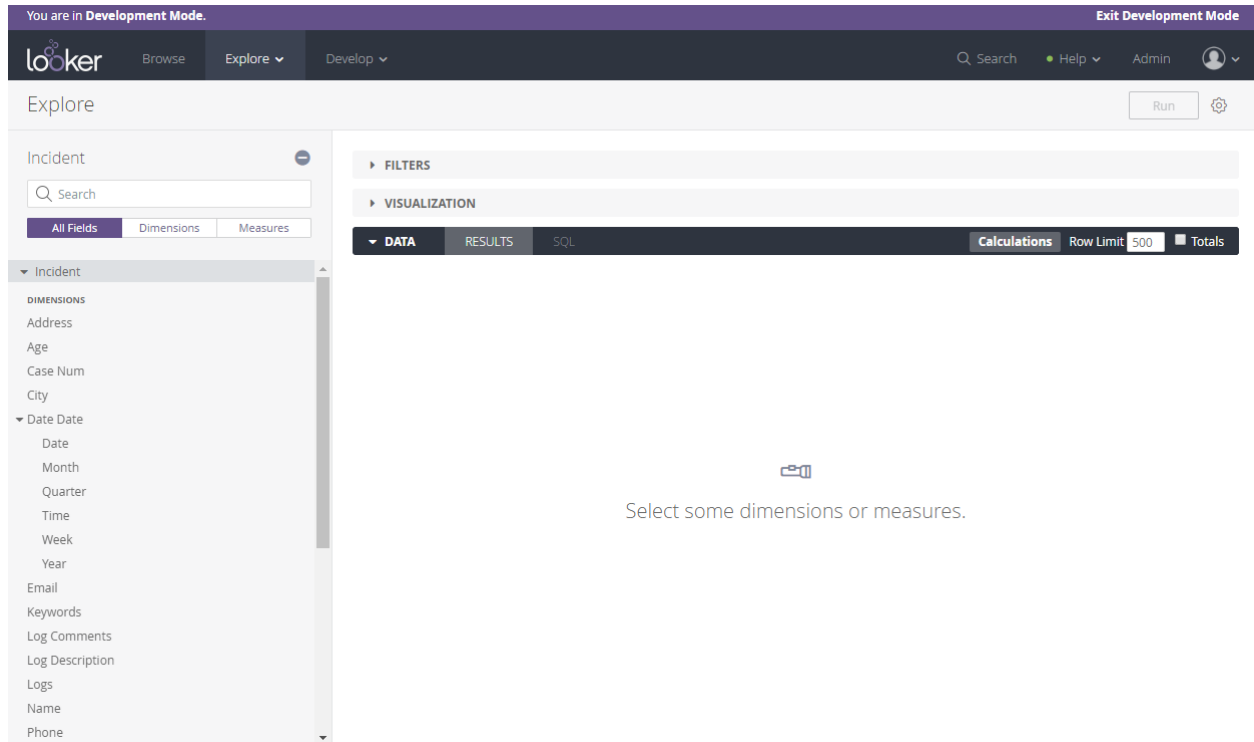


Once all the needed explorers are added, to access them click on the “Explore” menu and select the corresponding explorer in a project.



## Create a View/Look from a particular Explorer

To create a view click on an explorer created on the previous section.



After clicking on the explorer, the explorer’s workspace will be shown. The workspace can be used to build any view/look over the explorer. On the left side a panel all the dimensions/fields that belong to the explorer are shown together with different measures that could be applied over the underlying view.

There are two actions that can be performed on a dimension/measure: “Add” and “Filter”. In addition, just for dimensions the “Pivot” feature is available.

- Add dimension/measure to the **Data** section.

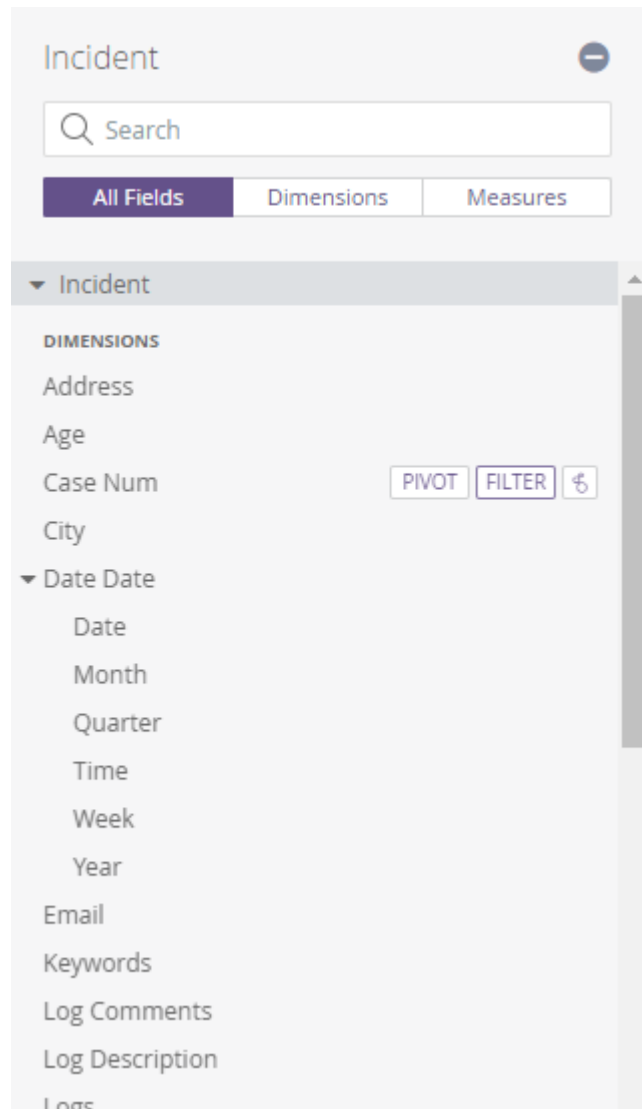
Adding a dimension/measure to the Data section means that it will be added to the select clause when querying the underlying view. This can be done by clicking on a dimension/measure.

- Add dimension/measure to the **Filters** section.

Adding a dimension/measure to the Filters section means that it will be added to the where clause as a condition when querying the underlying view. Before querying, the condition over this dimension/measure has to be set up in the Filters section.

This can be done by clicking on the “Filter” feature of the corresponding dimension/measure.



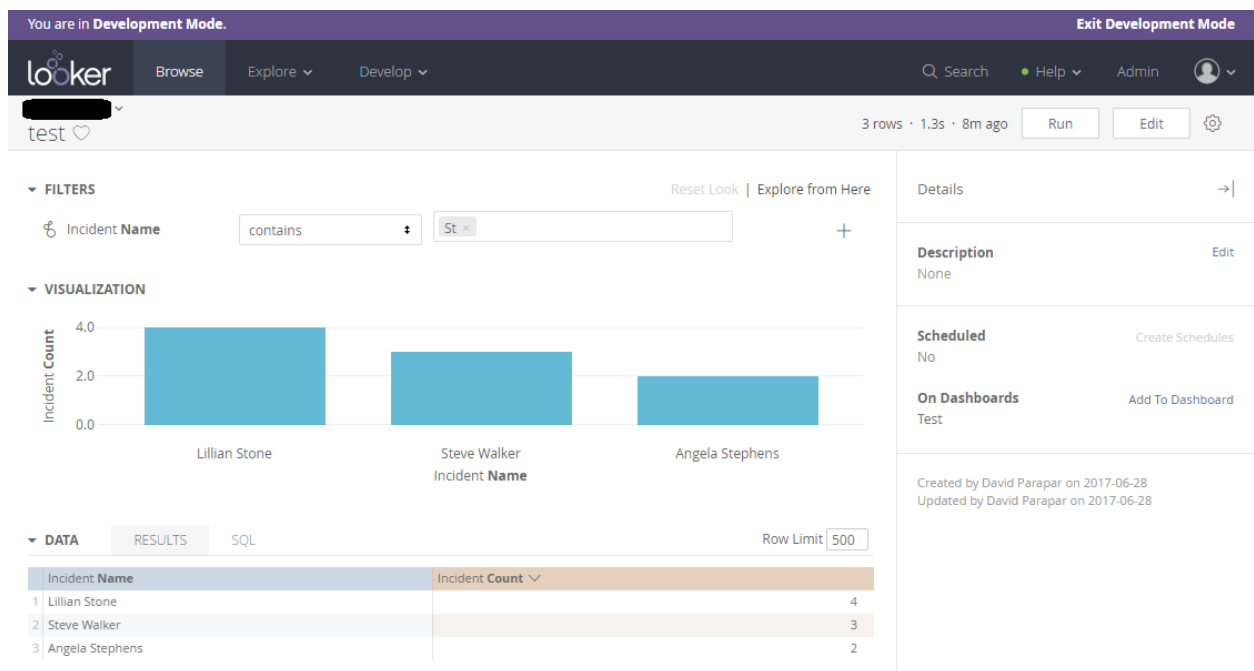


There are 3 sections on the right side of the explorer’s workspace:

- **Data**  
In this section the different dimensions and measures will be added as columns. The results gathered after executing the corresponding query on the underlying view will be displayed here after clicking on the “Run” button.
- **Visualization**  
In this section different options are available (table, columns, bar, scatterplot, line...) to select how the results obtained will be displayed. These visualization settings as a View/Look will be the same used when the View/Look is added to a Dashboard.
- **Filters**  
In this section different dimensions and measures can be added as where conditions. After clicking on the “Run” button a query will be executed over the underlying view including the filters configured here so the results gathered will meet these conditions.

For instance, let's create a View/Look from the Incident explorer in order to get how many incidents have been created by customers that have "St" as part of their names::

- Click on the "Name" dimension and the "Count" measure to add them to the "Data" section.
- Click on the "Filter" feature of the "Name" dimension to add it to the "Filter" section.
- Go to the "Filter" section, select the "contains" operation and enter "St" as the condition value.
- Go to the "Visualization" section and select the "Column" option.
- Click on the "Run" button in order to execute the corresponding query and see the results.
- Click on the "Gear" icon on the right upper corner and select the "Save as a look" option.
- Enter the name and a description.
- Finally click on "Save".

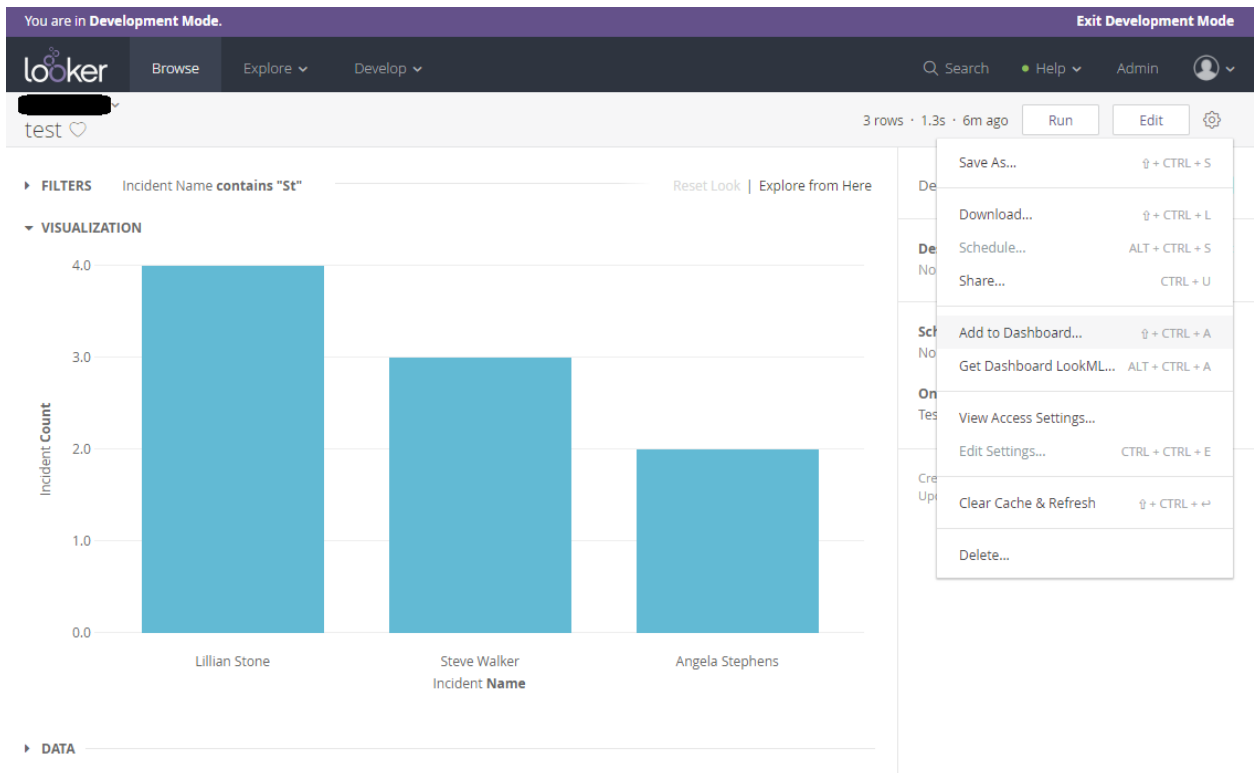


The screenshot shows the Looker interface in Development Mode. The view is named 'test' and has 3 rows. The filters section shows 'Incident Name' with the operation 'contains' and the value 'St'. The visualization is a bar chart showing the incident count for three names: Lillian Stone (4), Steve Walker (3), and Angela Stephens (2). The data table below the chart shows the following results:

| Incident Name     | Incident Count |
|-------------------|----------------|
| 1 Lillian Stone   | 4              |
| 2 Steve Walker    | 3              |
| 3 Angela Stephens | 2              |

## Add a View/Look to a Dashboard

To add a View to a new Dashboard click on the "Gear" icon on the top right corner and select the "Add to dashboard" option.



A view can be added to an existing dashboard or to a new one. For this example we are going to add this view to a new Dashboard so we click on “New Dashboard” and set a name.

After creating the dashboard, select it and click on the “Add” button so the view is added to this dashboard.

You are in **Development Mode**. Exit Development Mode

looker | Browse | Explore

test

Incident Name contains "S"

VISUALIZATION

Incident Count

Lillian Stone | Steve Walker | Angela Stephens

Incident Name

DATA

**Add Look to a Dashboard in this Space**

**Development Mode Enabled**  
Looks added to a Dashboard in Development Mode may not display correctly to other users until you sync your LookML with production.

The selected visualization is **Column**

Space: [Redacted]

Dashboard:

- Customer
- Denodo\_Customer360\_Incidents
- Test

Created by David Parapar on 2017-06-28  
Updated by David Parapar on 2017-06-28

You are in **Development Mode**. Exit Development Mode

looker | Browse | Explore | Develop

test

3 rows · 2.1s · just now

The "test" Look has been updated on "Test" Dashboard.

Filters: Incident Name contains St

Visualization: Incident Count

Incident Count

Lillian Stone | Steve Walker | Angela Stephens

Incident Name

DATA | RESULTS | SQL

| Incident Name     | Incident Count |
|-------------------|----------------|
| 1 Lillian Stone   | 4              |
| 2 Steve Walker    | 3              |
| 3 Angela Stephens | 2              |

Row Limit: 500

**Details**

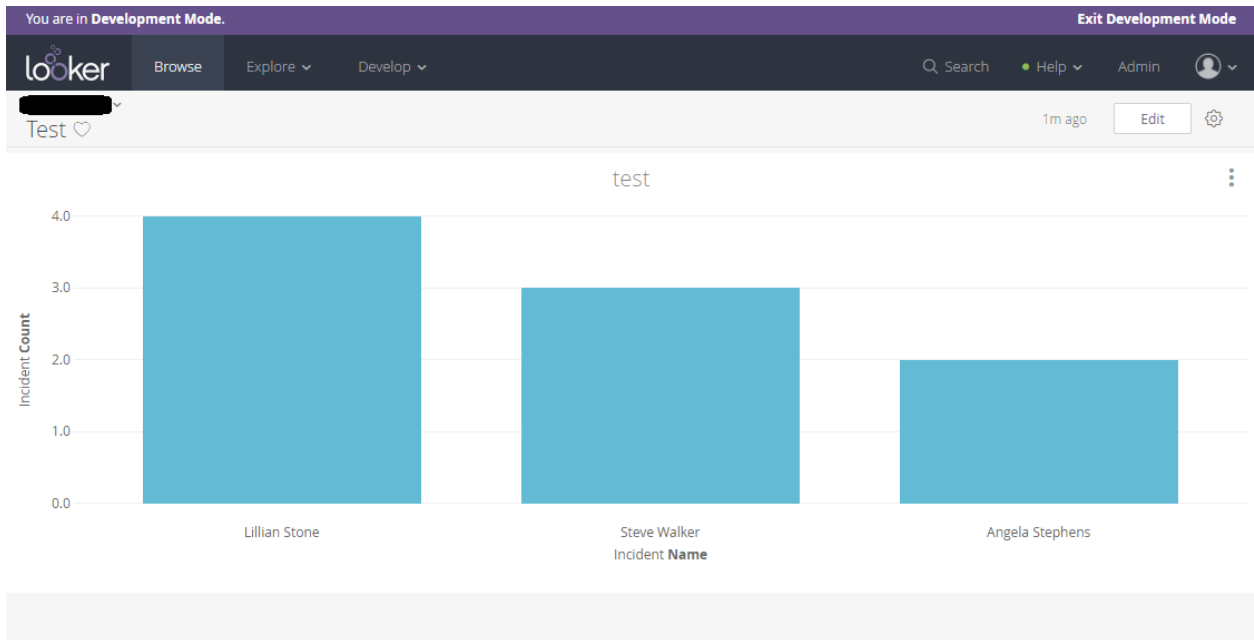
**Description**: None

**Scheduled**: No

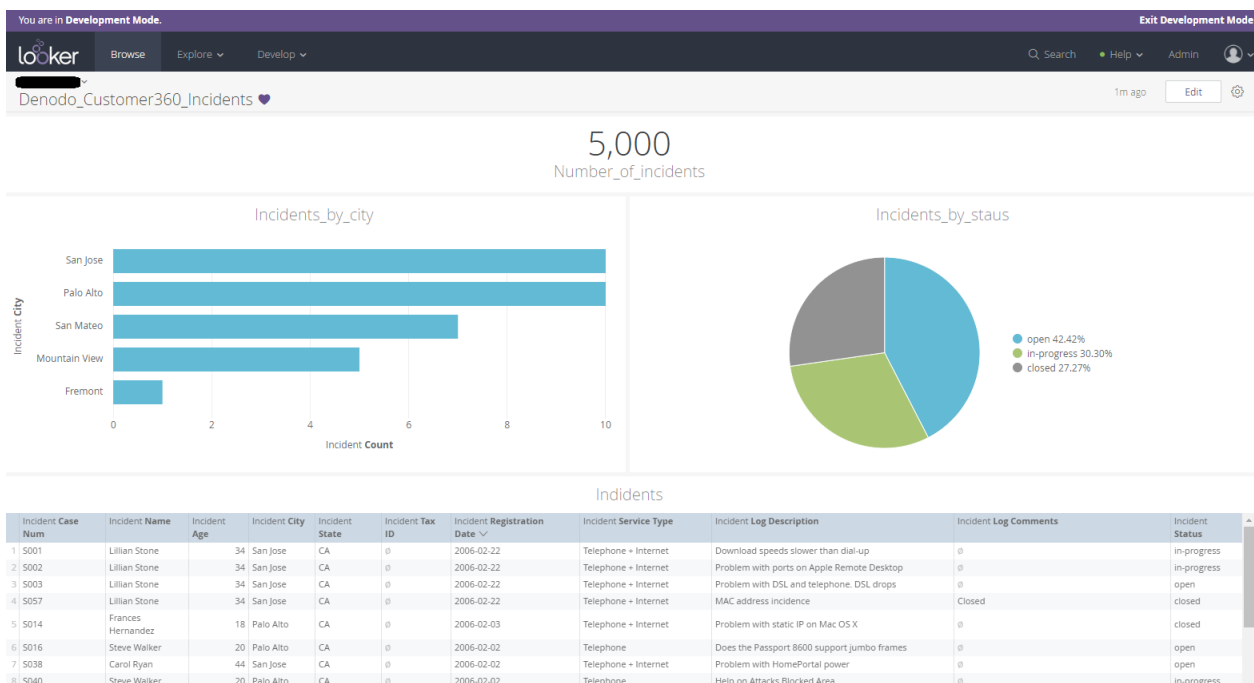
**On Dashboards**: Test

Created by David Parapar on 2017-06-28  
Updated by David Parapar on 2017-06-28

Once this is done, to go to the newly created dashboard click on the dashboard link with the chosen name or click on the “Browse” menu, by clicking on the link with the account name and selecting the corresponding dashboard in the Dashboards section.



It is possible to continue adding different looks to this dashboard in order to build a more complex and useful one as the following:



## References



[LookML Reference](#)