

# Getting up to speed with CARTO

## Workshop objectives

1. Create your first interactive map with CARTO Builder
2. Transform this into a more interactive dashboard - complete with AI Agents - to enhance user experience
3. Learn how to run some simple spatial analysis and integrate this into your map

## Workshop outline

In this workshop, we'll be using road traffic accident data for London from 2024. We'll be using this to create a dashboard which helps our end-users understand areas of high risk.

Here's how we'll do it!

[Part 1... Making your first map](#)

[Part 2... Integrating further analysis](#)

Spatial Analysis roadshow [github repo](#)

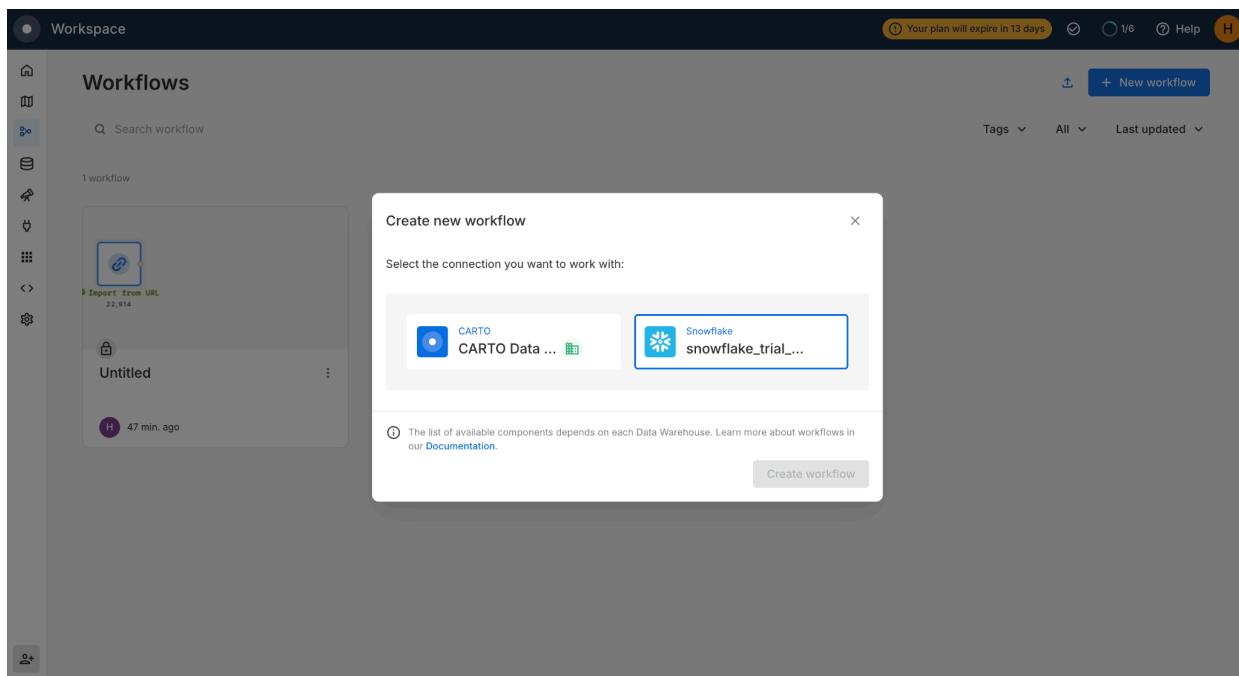
# Part 1: Making your first map

## 1.1 Importing data

We'll be using road traffic accidents sourced from the UK Department for Transport [here](#), which we've imported to this [Google Sheet](#).

There are lots of ways to import data into your Snowflake lakehouse with CARTO. We're going to be importing the Google Sheet directly via URL into CARTO Workflows - our low-code tool for automating spatial analysis pipelines - so we can integrate it easily with some wider analysis.

1. In the CARTO Workspace, head to the **Workflows** tab and select **Create a new workflow**. Select your Snowflake connection to use for this.



2. In the Workflow, switch to the **Components** tab on the right of the screen.
3. Search for **Import from URL** and drag this component onto the canvas.
4. Enter the below URL into the dialog and **Run** your workflow.

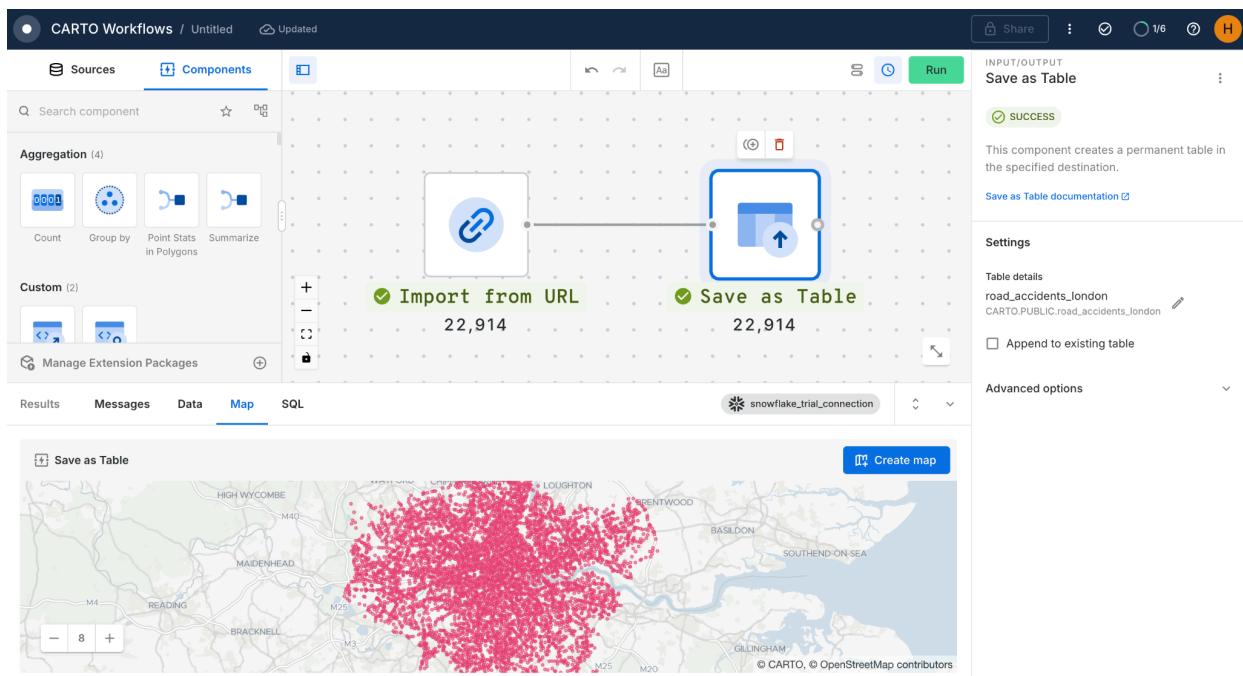
```
https://docs.google.com/spreadsheets/d/11I2WwnbpByYks21-vMa0p31bChEED2f6VS7EWKD5w1I/gviz/tq?tqx=out:csv&sheet=Collisions
```

 You can replicate this for any Google Sheet using the following syntax:

<https://docs.google.com/spreadsheets/d/{sheetid}/gviz/tq?tqx=out:csv&sheet={tab}>

💡 If you have any issues with this process, you can also download the data from the github [here](#), and just drag and drop it onto the Workflow canvas.

5. You can preview the imported data in the Map and Data preview tabs at the bottom of the screen.
6. Drag a **Save as Table** component onto the canvas and connect it to the Import as URL component (see below). Set a location and table name in your Snowflake lakehouse to save the table to. Run!



7. Open the Map preview of the **Save as Table** component, and select Create Map. You will now be taken to a fresh version of CARTO Builder with your data pre-loaded - now let's get mapping!

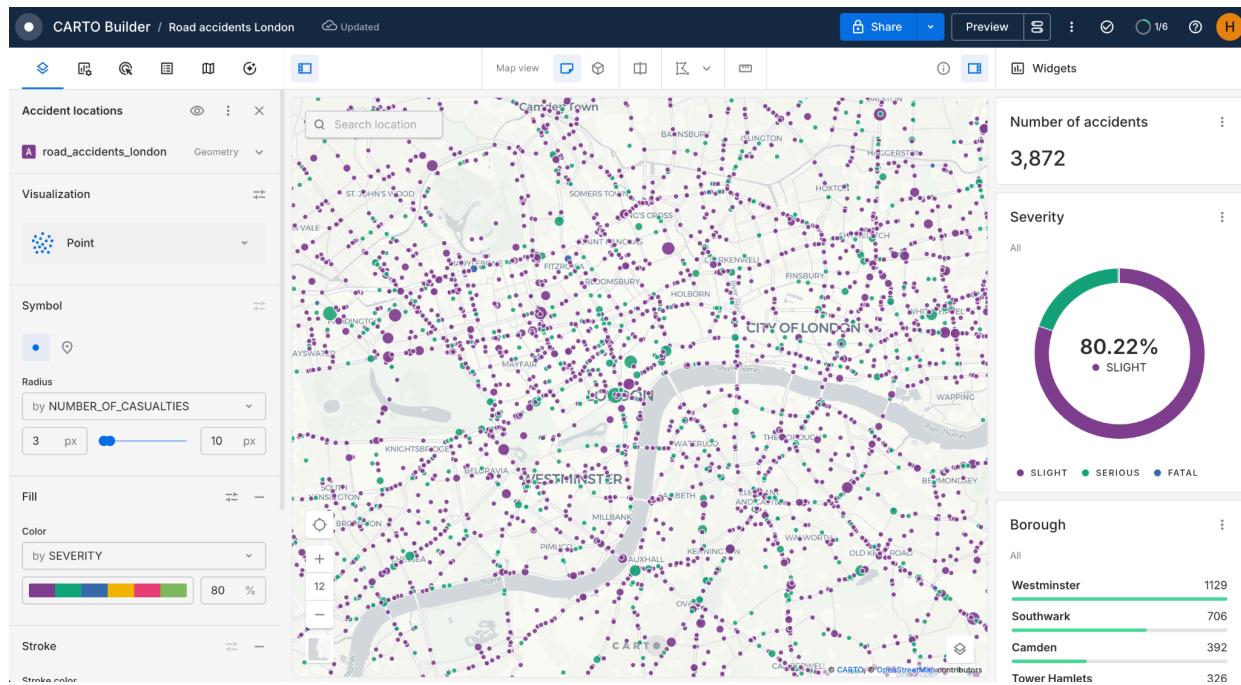
## 1.2 Building a simple map

1. First thing's first! At the top left of your screen, click the pre-generated map name to rename your map.
2. In the **Layers** panel on the left-side of the screen, you will see that the data is loaded into Layer 1. Click on the three dots next to Layer 1 to rename your layer.

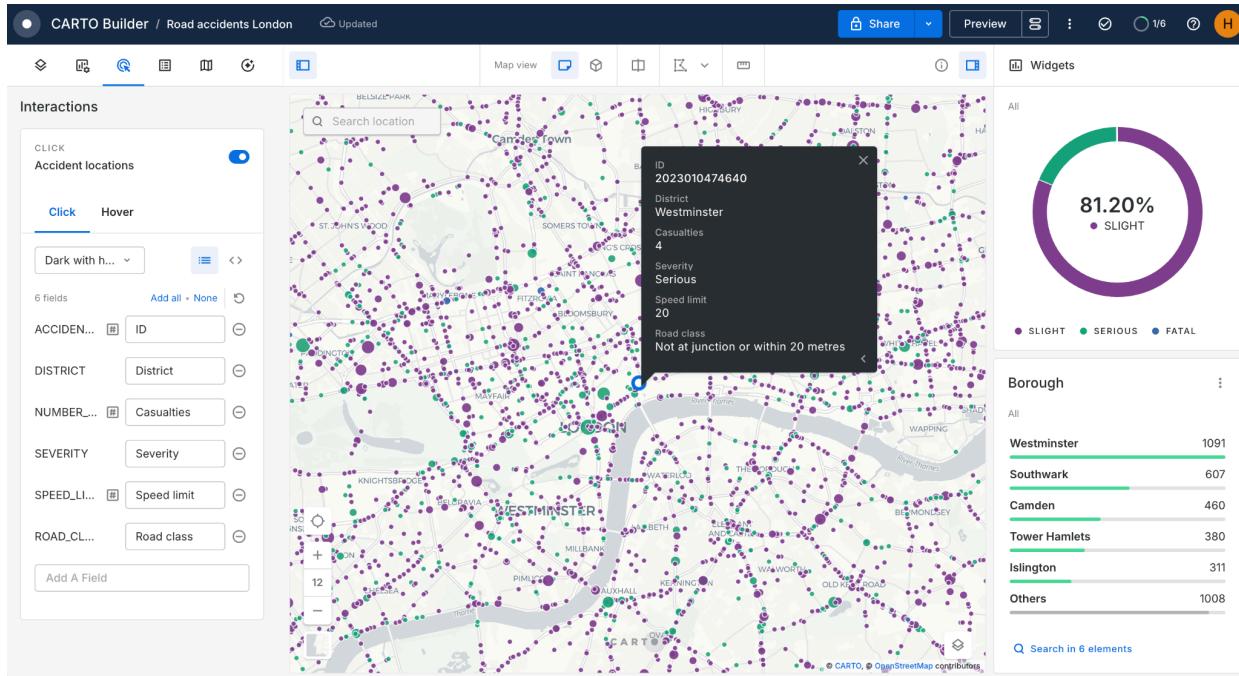
3. Click on the layer name to open the styling options for the layer... and over to you! Here are some styling options you might like to explore:

- a. In the **Visualization** section, try the different layer visualization options like heatmap or grid.
  - b. Set the **Symbol** radius to be determined by the number of casualties, with a radius ranging from 3 to 12.
  - c. Change the **Fill** and **Stroke** colors, or set them to be defined by one of the variables - we've used severity.
4. Let's add a couple of quick interactions to make our map easier to navigate. On the top-left of the screen, switch from the Layers panel to the following tabs:

- a. **Widgets:** these are dynamic graphical elements which help your user to understand your data. We've added a formula, pie and category widget. Change the widget behaviour to "viewport" and watch the values change as you move around the map!



- b. **Pop-ups:** from the **Interactions** menu, add a pop-up and select the variables you want to show in each pop-up. You can change the style, formatting and field names - and even edit the HTML to fully customize these.



 Getting ahead? There's much more to explore with Builder - try switching to 3D view, experimenting with blending modes, changing the basemap or adding a map description.

5. **Share** your map! You can do this from the menu at the top-right of the screen.

We'll come back to this map in a bit...

## 1.3 Adding AI Agents to your map

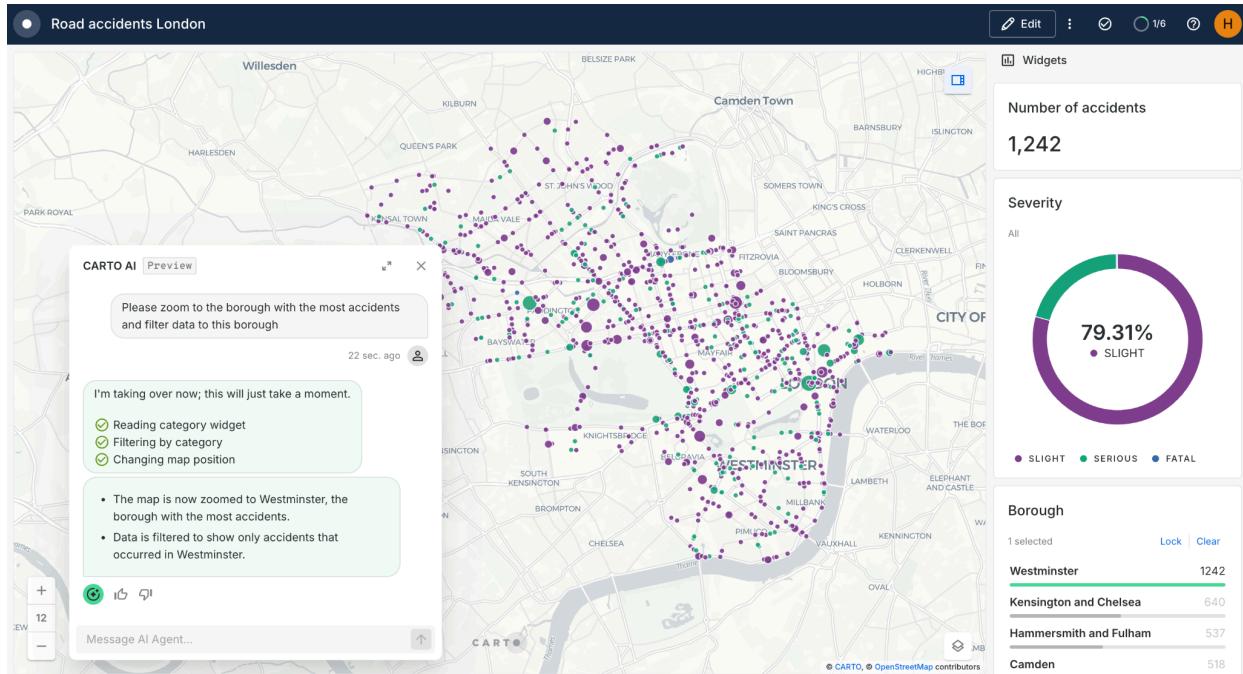
Now, let's get our map AI enabled!

1. Back in the [CARTO Workspace](#), switch to the **Settings** tab.
2. Go to Customizations > AI Agents and toggle on **Enable AI in Builder**.
3. Back in your Builder map, refresh the page and you will now see a new **AI Agents** panel on the left of the screen. From here, toggle this on and **republish your map**.
4. Now, switch to map **Viewer** mode. You can do this by simply replacing "Builder" with "Viewer" in your URL.
5. You should see the AI Agent at the bottom of the screen - meaning your map is AI-ready! Let's try quizzing it... Here are some of the things you can ask:
  - a. Search and zoom to locations, or extract map coordinates.
  - b. Extract widget values, or filter data by widgets.

### c. Switch layers on and off

With this in mind, you may want to go back to editor mode (easily do this by changing "viewer" to "builder" in your url) and further configure widgets to help you ask the questions you want. So, for example with our map, we've asked "Please zoom to the borough with the most accidents and filter data to this borough."

Note in the AI Agent tab you can provide additional context, a user guide and example questions.



Remember to republish your map any time you make changes!

# Part 2 [extension]: Integrating further analytics with your map

## 2.1 Aggregating the points to a grid

In our map in its current state, it's quite hard to derive any meaningful patterns when zoomed out to the full extent - let's change that by creating a frequency grid that appears as we zoom out.

1. First, we need to add a second instance of this layer. In the **Sources** panel, select **Add source from...** and navigate to where your layer is saved in your Snowflake lakehouse to add it to the map.
2. Like we did before, rename the layer Accident frequency grid.
3. In the Sources panel, click the three dots next to the newly created source B and select **Query this table** to bring up the in-app SQL console.
4. Add the following code as a field to the SQL query. This will generate a new field called H3 which contains a H3 Spatial Index reference.

```
, CARTO.CARTO.H3_FROMGEOPOINT(geom, 10) AS h3
```

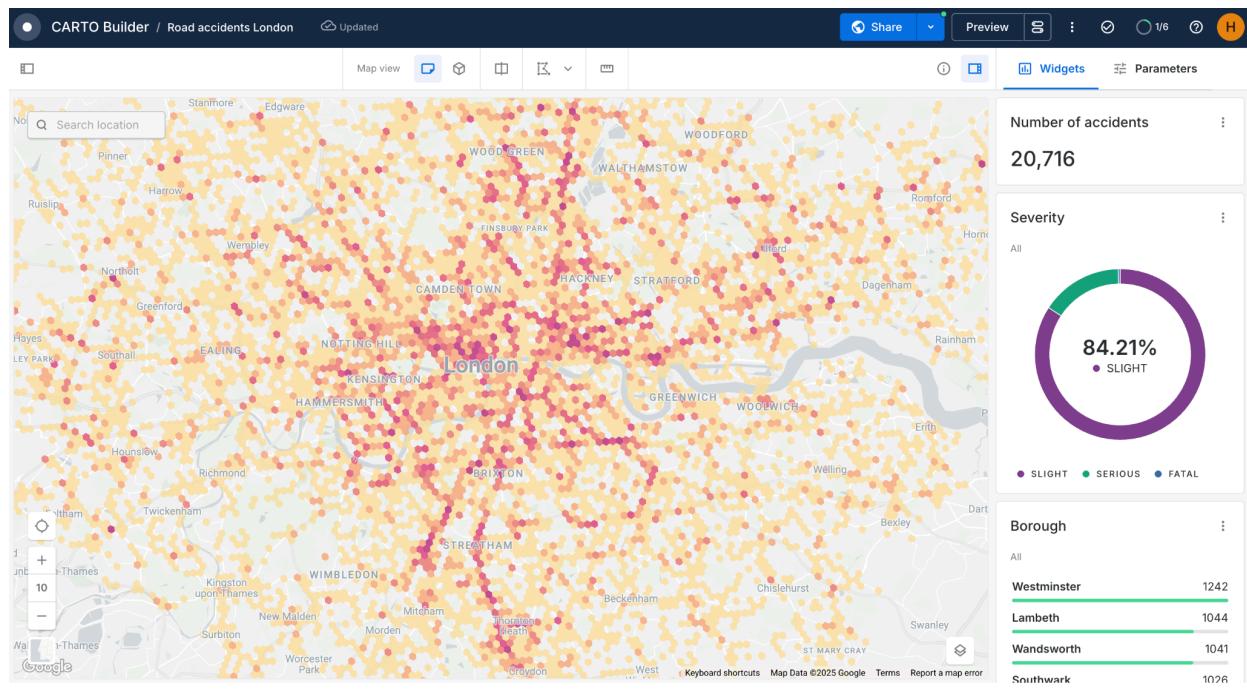
So, altogether your new query should look like:

```
SELECT *, CARTO.CARTO.H3_FROMGEOPOINT(geom, 10) AS H3 FROM  
"PROJECT"."SCHEMA"."ROAD_ACCIDENTS_LONDON"
```

 Note that the above analysis can also be run from your Snowflake console, or completed code-free in Workflows!



5. Open the **Layer** style for this layer. Expand the drop-down next to geometry, and set the **spatial column** to H3 and the **spatial type** to H3 and click Apply... and watch your layer transform into a grid layer!
6. Remove the **Stroke** and set the **Fill** to NUMBER\_OF\_CASUALTIES(SUM) with a quantize scale - and you have an aggregated grid!



Check out how the resolution of the layer changes as you zoom in and out! You can also set this manually with the **Aggregation size** option in the layer style. Learn more about this and other unique properties of Spatial Index grids like H3 in our [ebook](#).

**💡 Extension:** looking integrate more statistical analysis into your spatial problem-solving? This would be a good moment to leverage the Getis Ord\* statistic to calculate accident hotspots:

The screenshot shows a workflow titled "Hotspots" in the "CARTO Workflows / Hotspots" section. The workflow consists of five steps connected by arrows: "H3 from GeoPoint" (with 22,914 points), "Group by" (with 13,449 groups), "Getis Ord" (with 13,449 results), "Where" (with 1,195 filtered results), and "Save as Table" (with 1,195 saved rows). The "Getis Ord" step is currently selected. Below the workflow, the "Data" tab is active, displaying a table with three columns: "H3" (string), "GI" (number), and "P\_VALUE" (number). The table contains four rows of data. To the right of the table, there are settings for the analysis: "Index column" set to "H3", "Value column" set to "NUMBER\_OF\_CASUALTIES\_SUM", and "Kernel function for spatial weights" set to "triangular". The "Size" setting is set to 3. The overall status is "SUCCESS".

Check out our full guide to this here: [How to calculate spatial hotspots and which tools do you need?](#)

## 2.2 Setting zoom visibility

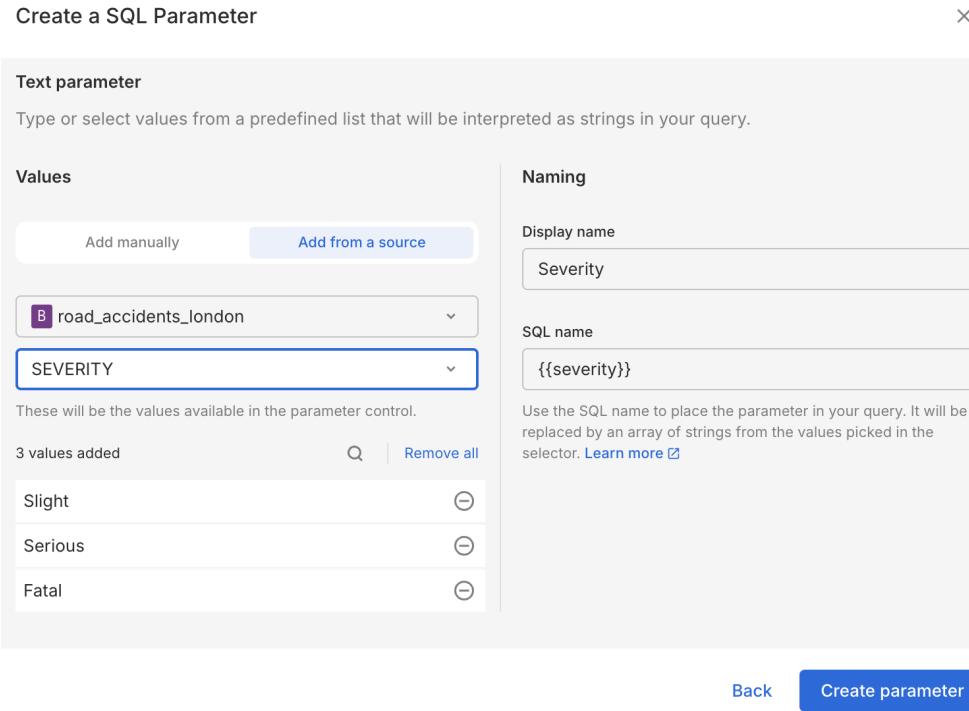
Now, let's set up our map so that we see the frequency grid when we zoom out, and individual points when we zoom in.

1. In the original Accident locations (point) layer menu, expand the **Visualization** section. Set the **Visibility by zoom level** to be from 13-24. Now, we'll only see the individual points when we zoom in closer, making our map much easier to read.

## 2.3 Adding user-defined SQL parameters

Our map is looking great! Now let's add some SQL parameters to our map. These are "placeholders" which can be added to the queries driving your map. Your end users can specify the values to be substituted into these placeholders, allowing them to run some analysis (controlled by you!) - saving you both huge amounts of time!

1. Open the **Parameters {}** window from the **Sources** panel.
2. Add a text parameter, giving it the display name **Severity**. Note the SQL name - likely `{{severity}}`.
3. Under **Values**, select **Add from source**. Choose the individual point location layer (likely source A) and choose the field **SEVERITY**.



4. You should now see a **Parameters** tab has been created on the right-hand side of your screen, in the same location as the widget tab.
5. Let's add a second parameter - this time choosing a numeric parameter.
  - a. Call this parameter Road speed.
  - b. Change the type from Simple slider to **Range slider** - you will notice this time that two SQL parameters have been generated; {{road\_speed\_from}} and {{road\_speed\_to}}.
  - c. Set the minimum and maximum values to 20 and 70 respectively.
  - d. Change the scale from Continuous to **Discrete**, and set the **Step increment** as 10.

Now, we just need to add these to our sources!

6. In the **Sources** panel, click on the three buttons next to your point layer, and select **Query this table**. Add the following SQL snippet to the end of your query:

```
WHERE SEVERITY IN {{severity}}
AND
(SPEED_LIMIT >= {{road_speed_from}} AND SPEED_LIMIT <= {{road_speed_to}})
```

7. If you switch to the **Preview** of the map, you will see that your map users can now select values from the list - and these in turn update the SQL query, which update the wider map!
8. Finally, let's repeat this for the aggregated grid layer - it should now look something like the below (make sure to update the section highlighted in yellow).

```
SELECT *, CARTO.CARTO.H3_FROMGEOPOINT(geom,10) AS H3
FROM "PROJECT"."SCHEMA"."ROAD_ACCIDENTS_LONDON"
WHERE SEVERITY IN {{severity}}
AND
(SPEED_LIMIT >= {{road_speed_from}} AND SPEED_LIMIT <=
{{road_speed_to}})
```

Now, both layers should update when you modify the parameters!

---

Thank you for completing this workshop! If you have any feedback or questions, please contact [marketing@cartodb.com](mailto:marketing@cartodb.com), or reach out to the workshop host Helen on [LinkedIn](#).

Looking for more learning resources? Check out the [CARTO Academy](#)!