Visualizing Call Recordings with Amazon Quicksight

Call recordings have been processed with Amazon Transcribe and Amazon Comprehend and the information obtained from them has been dumped into a JSON document.

The current Document structure is as follows:

```
{
  "ConversationAnalytics": {
    "ConversationTime": "string",
    "ProcessTime": "string",
    "Duration": "float",
    "LanguageCode": "string",
    "EntityRecognizerName": "string",
    "SpeakerLabels": [
     {
       "Speaker": "string",
       "DisplayText": "string"
     }
    ],
    "SentimentTrends": [
     {
       "Speaker": "string",
       "AverageSentiment": "float",
       "SentimentChange": "float"
     }
    ],
    "CustomEntities": [
     {
        "Name": "string",
        "Count": "integer",
        "Values": [ "string" ]
     }
    ],
    "SourceInformation": [
     {
      "TranscribeJobInfo": {
        "TranscriptionJobName": "string",
        "CompletionTime": "string",
        "VocabularyName": "string",
        "MediaFormat": "string",
        "MediaSampleRateHertz": "integer",
        "MediaFileUri": "string",
        "MediaOriginalUri": "string",
        "ChannelIdentification": "boolean",
        "AverageAccuracy": "float"
      }
     }
```

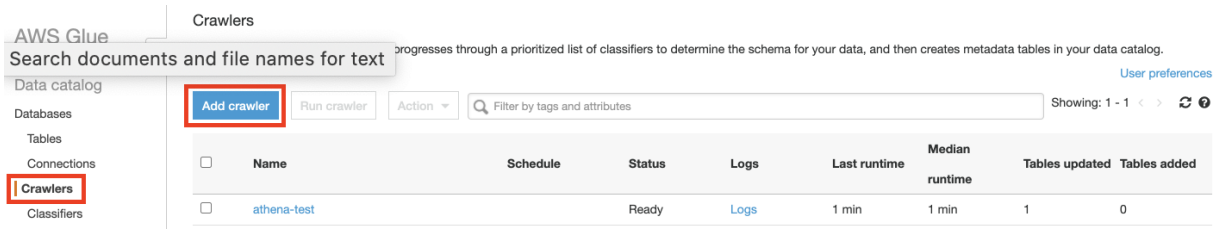
```
  ]
 },

 "SpeechSegments": [
  {
   "SegmentStartTime": "float",
   "SegmentEndTime": "float",
   "SegmentSpeaker": "string",
   "OriginalText": "string",
   "DisplayText": "string",
   "TextEdited": "boolean",
   "SentimentIsPositive": "boolean"__,
   "SentimentIsNegative": "boolean",
   "SentimentScore": "float",
   "BaseSentimentScores": {
    "Positive": "float",
    "Negative": "float",
    "Neutral": "float",
    "Mixed": "float"
   },
   "EntitiesDetected": [
    {
     "Type": "string",
     "Text": "string",
     "BeginOffset": "integer",
     "EndOffset": "integer",
     "Score": "float"
    }
   ],
   "WordConfidence": [
    {
     "Text": "string",
     "Confidence": "float",
     "StartTime": "float",
     "EndTime": "float",
     "Entity": "string"
    }
   ]
  }
 ]
}
```

In order to aggregate this data and consume it we will use Amazon Athena.

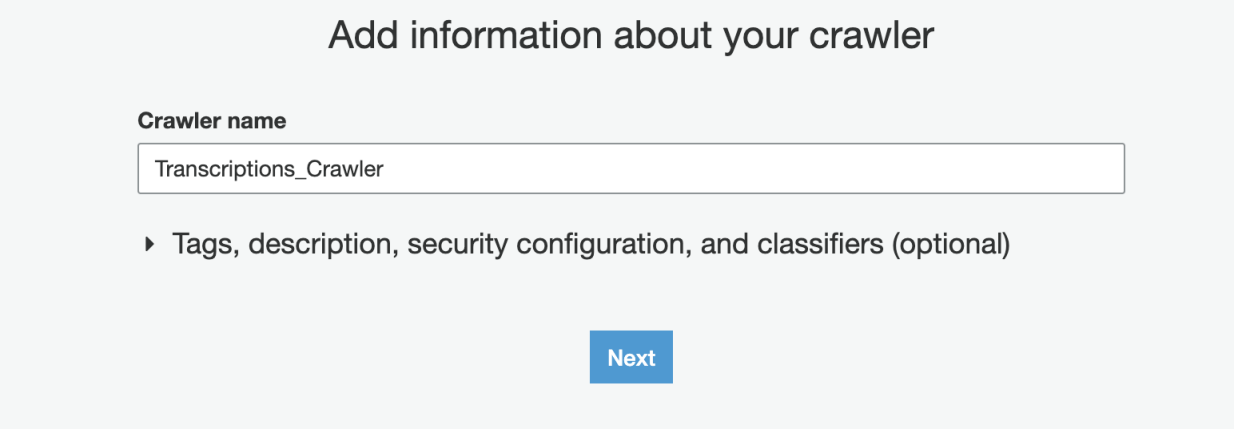
The first step is to create a table definition in AWS Glue Data Catalog, which will allow us to define where the Json files will be stored. At a high level we will define the location of the files and run a Crawler

Adding a Crawler to create a database

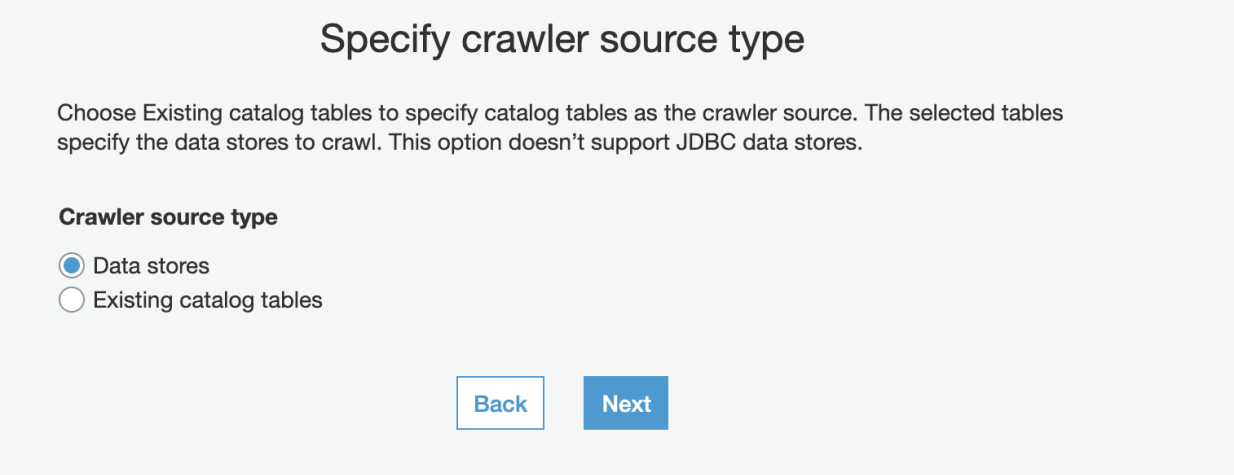
On the AWS Glue console, under Data Catalog, click on Crawlers, and then on the “Add Crawler” button.



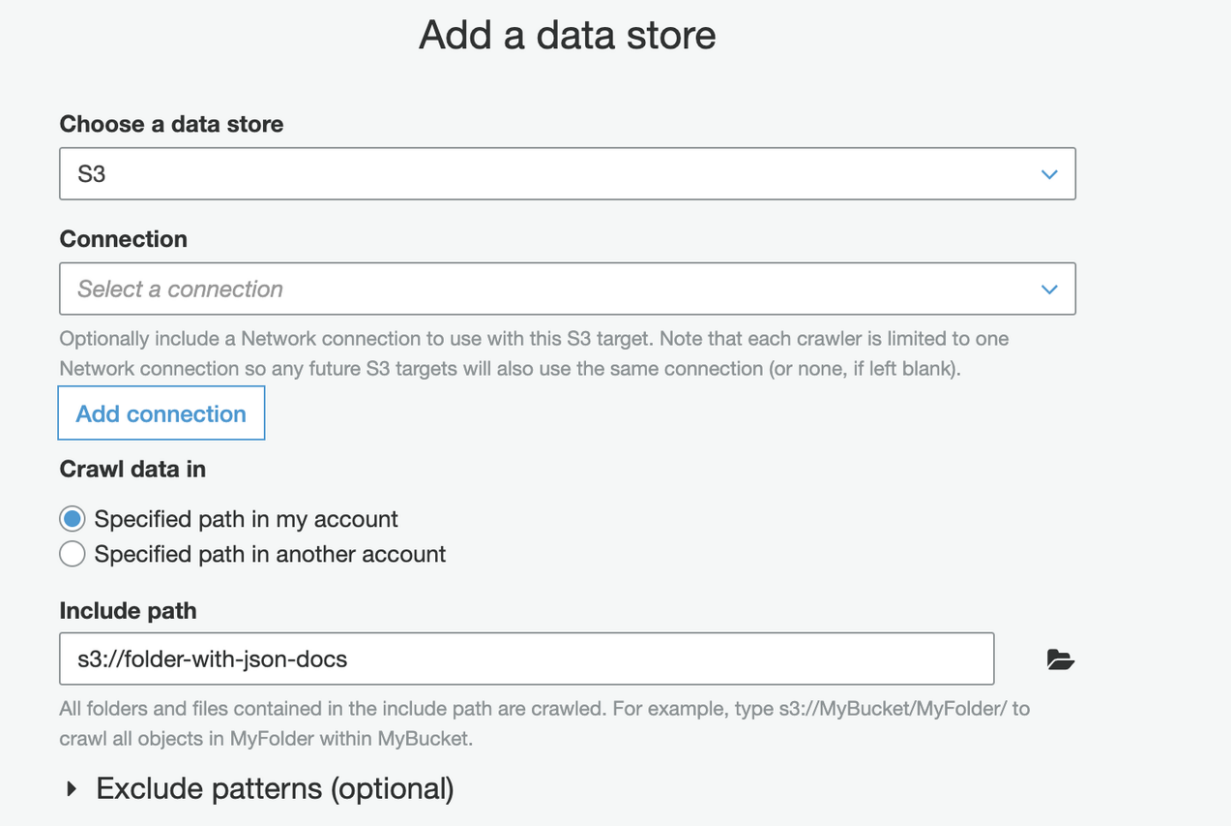
2. Enter a name and click Next



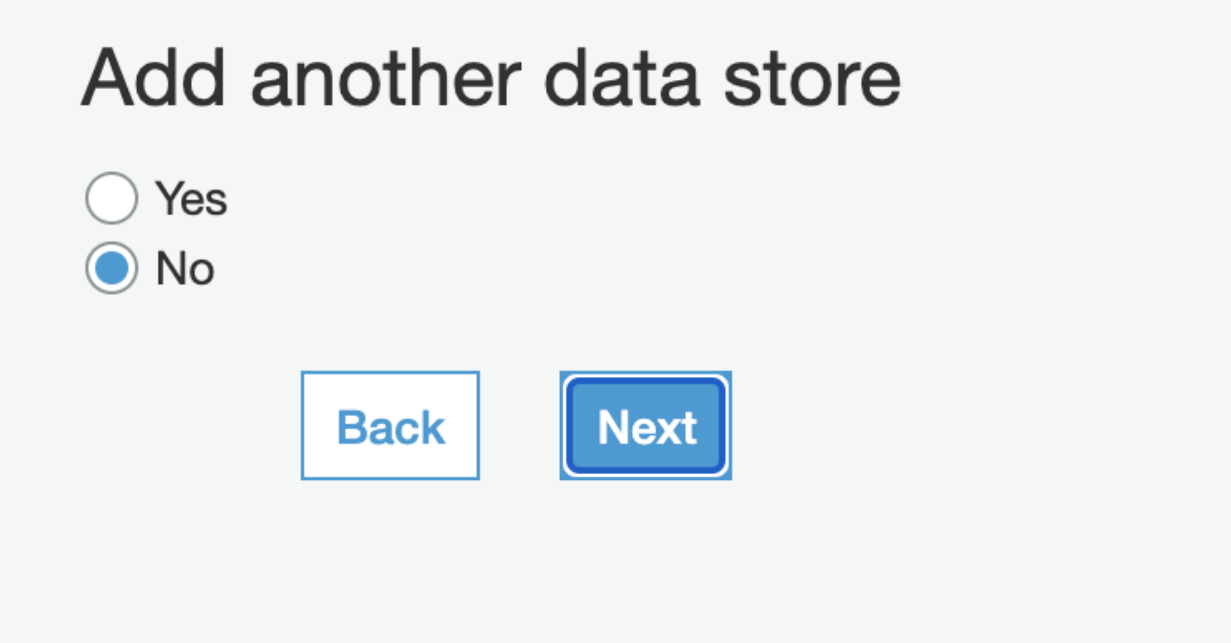
3. Select Data Stores and click Next



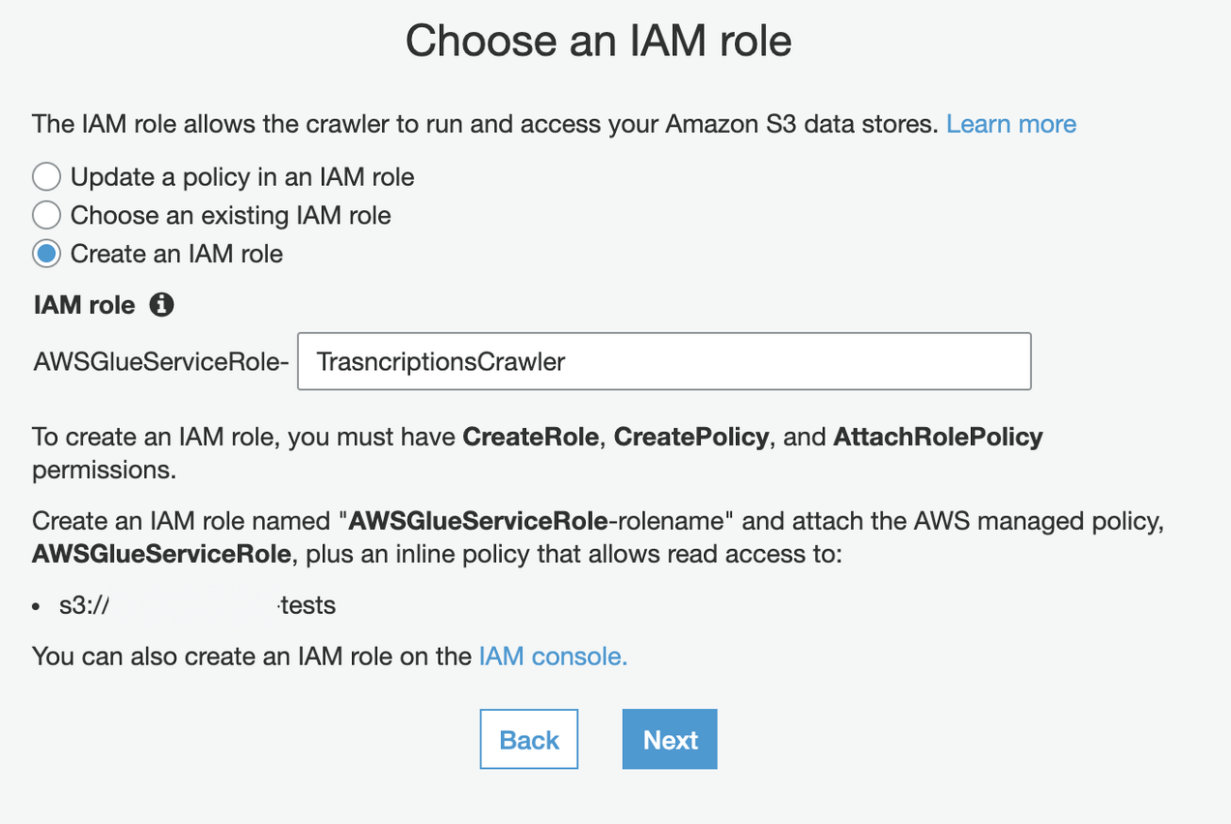
4. For the data store we will choose S3, and we will specify the path where the transcription json documents are stored on the "Include Path" field, then click next



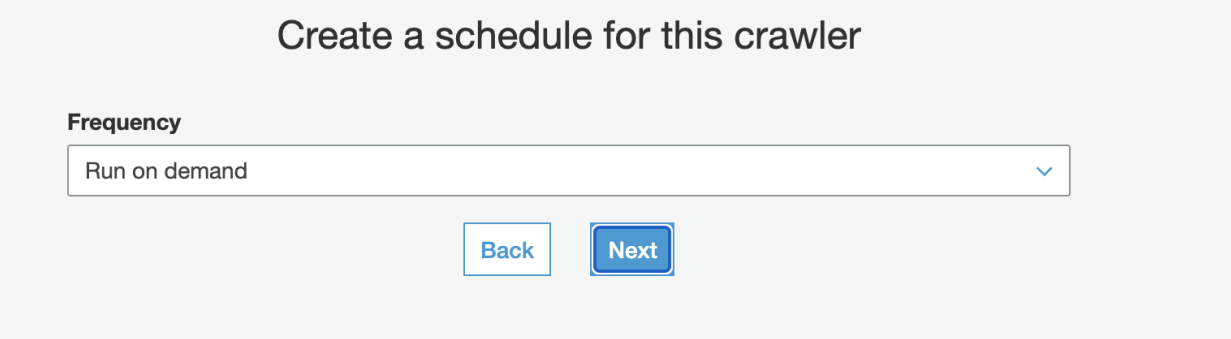
5. On add another data store, select No



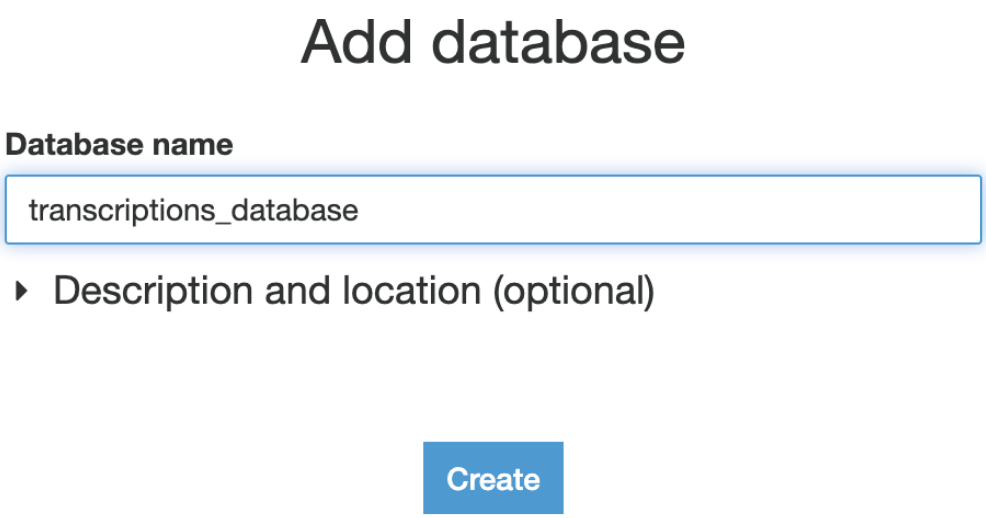
6. For the IAM role we will create a new IAM role, that will include a policy that allows read access to the S3 target we defined before:



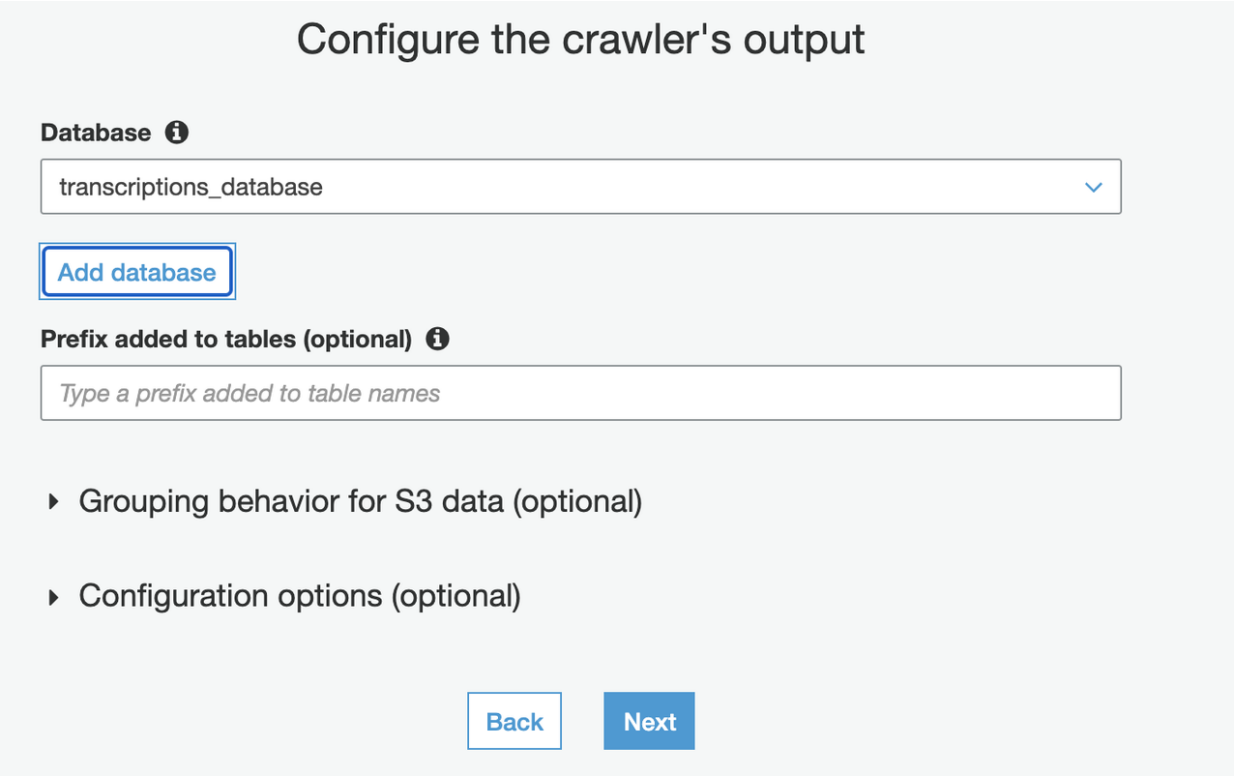
7. Define a schedule for your crawler, in this case we will select on demand:



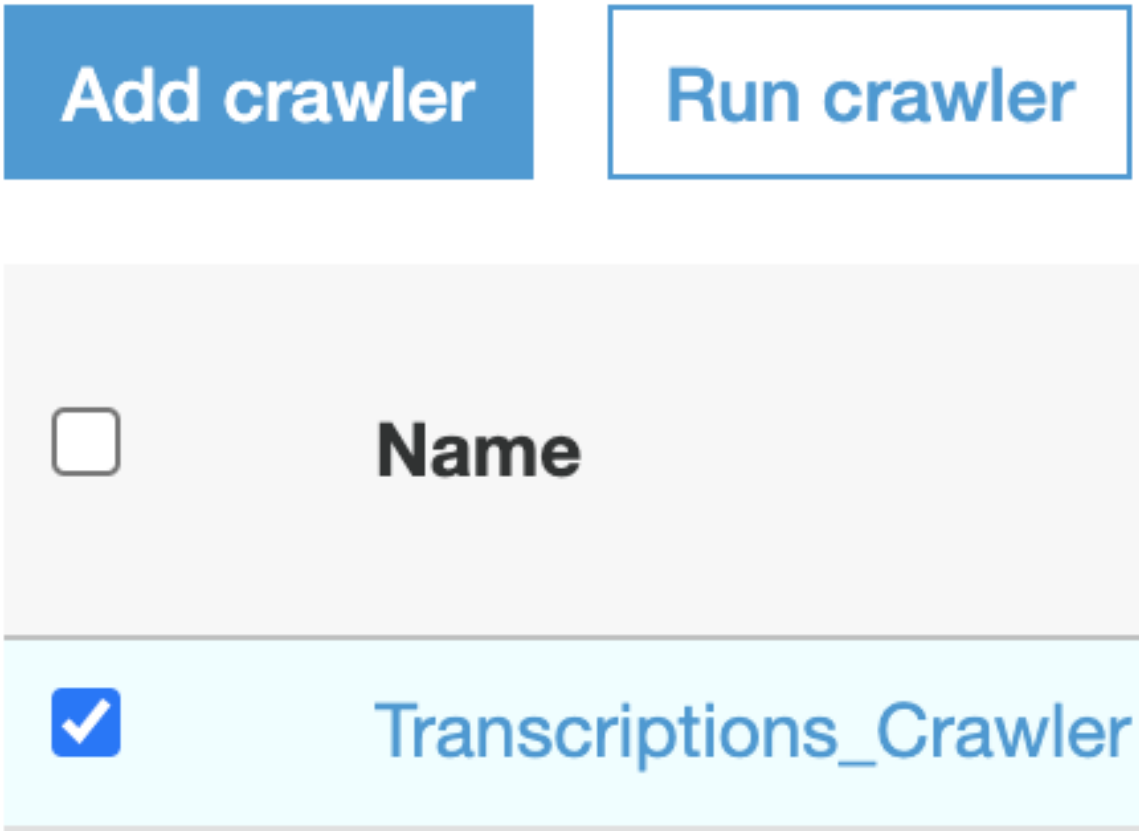
8. On the “Configure the Crawler’s output” screen, click on Add database, and enter a name.



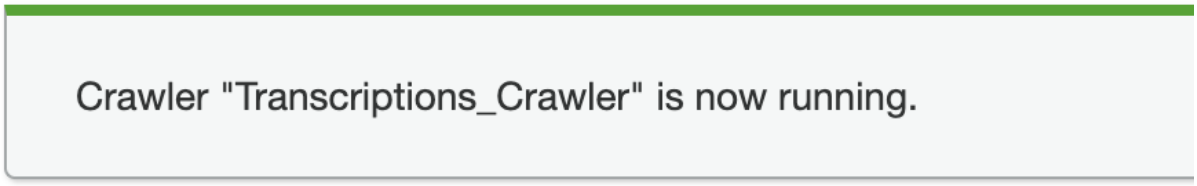
9. Select the new database you created and click next.



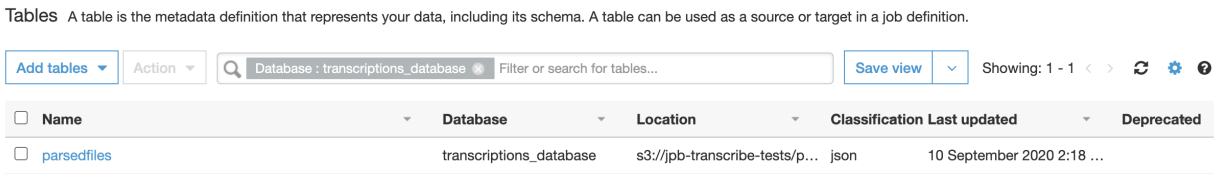
10. Click on Finish.
11. On the crawlers list, select your newly created crawler and click on “Run Crawler”



12. Wait for the crawler to finish



13. You can check if a new table has been created based on your transcriptions under Databases -> Tables

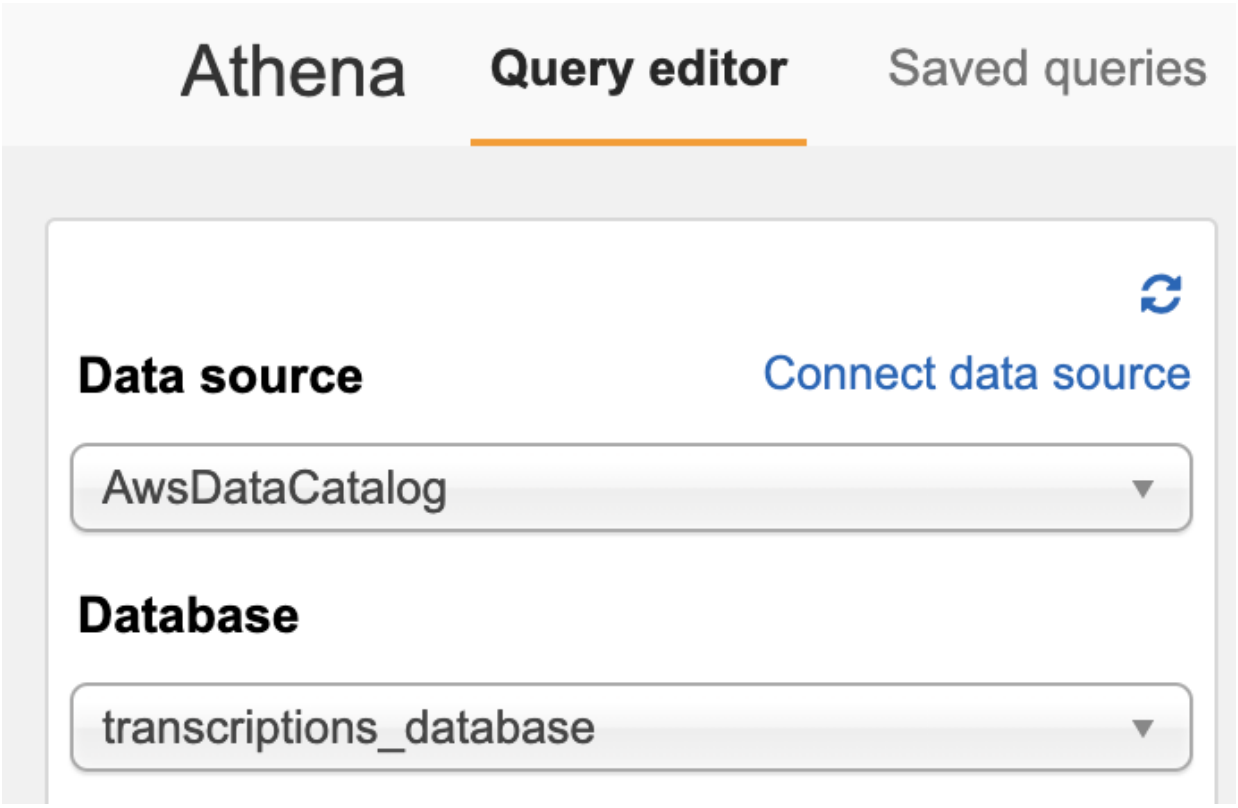


Creating queries and views with Athena

Transcription data will be made available to Amazon Quicksight through an Amazon Athena query.

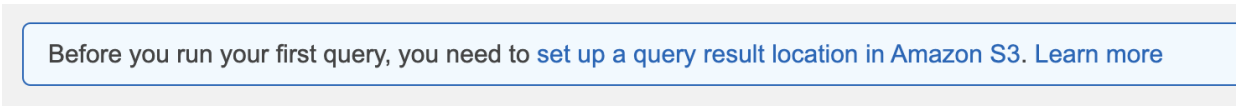
We will generate 3 views in order to have different data dimensions available.

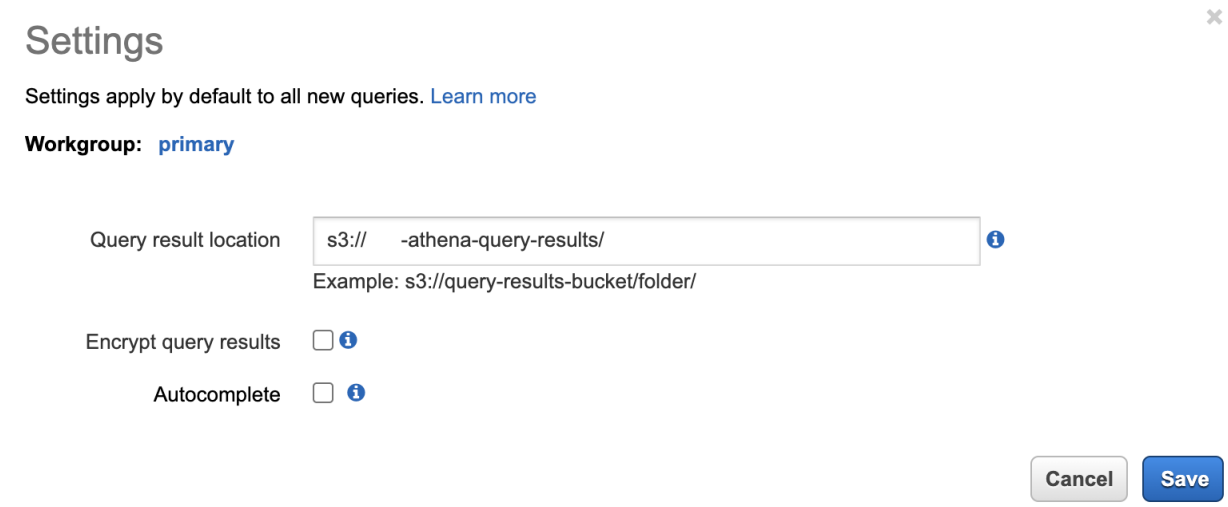
On the Amazon Athena query editor, we will select AwsDataCatalog as our Data source and we will choose the database we just created:



If you haven't used Amazon Athena before you need to set up a query result location on S3.

1. Create an S3 bucket for storing your results.
2. Click on the "set up a query result location in Amazon S3" link, and enter the name of the bucket you will use to store your results and click save.





We are going to generate 3 views that will be used by quick sight.

1st View – Call durations and Languages

```
CREATE OR REPLACE VIEW call_duration_and_language AS
SELECT
"regexp_extract"("$path", '[^/]+$') "filename"
, "conversationanalytics"."languagecode"
, "conversationanalytics"."speakerlabels"[1]."displaytext" "speaker_0"
, "conversationanalytics"."duration"
, "conversationanalytics"."conversationtime"
, "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtim
FROM
"Parsedfiles"
```

```
CREATE OR REPLACE VIEW call_duration_and_language AS
SELECT
  "regexp_extract"("$path", '[^/]+$') "filename"
, "conversationanalytics"."languagecode"
, "conversationanalytics"."speakerlabels"[1]."displaytext" "speaker_0"
, "conversationanalytics"."duration"
, "conversationanalytics"."conversationtime"
, "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
FROM
  "Parsedfiles"
```

2nd View – Conversation Segments

```
CREATE OR REPLACE VIEW conversation_segments AS

SELECT

"regexp_extract"("$path", '[^/]+$') "filename"

, "conversationanalytics"."languagecode"

, "conversationanalytics"."speakerlabels"[1]."displaytext" "speaker_0"

, "conversationanalytics"."conversationtime"
```

```
, "conversationanalytics"."duration"

, "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtim

, "segment"."segmentstarttime"

, "segment"."segmentendtime"

, "segment"."segmentspeaker"

, "segment"."sentimentispositive"

, "segment"."sentimentisnegative"

, "segment"."sentimentscore"

, "segment"."displaytext"

, "segment"."textedited"

, "segment"."entitiesdetected"

, (CASE WHEN (("conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND

FROM

(parsedfiles

CROSS JOIN UNNEST("speechsegments") t (segment))
```

```
CREATE OR REPLACE VIEW conversation_segments AS
SELECT
  "regexp_extract"("$path", '[^/]+$') "filename"
, "conversationanalytics"."languagecode"
, "conversationanalytics"."speakerlabels"[1]."displaytext" "speaker_0"
, "conversationanalytics"."conversationtime"
, "conversationanalytics"."duration"
, "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
, "segment"."segmentstarttime"
, "segment"."segmentendtime"
, "segment"."segmentspeaker"
, "segment"."sentimentispositive"
, "segment"."sentimentisnegative"
, "segment"."sentimentscore"
, "segment"."displaytext"
, "segment"."textedited"
, "segment"."entitiesdetected"
, (CASE WHEN (("conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND ("segment"."segmentspeaker" =
'spk_0')) THEN 'Agent' WHEN (("conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND
("segment"."segmentspeaker" = 'spk_1')) THEN 'Caller' END) "Speaker"
FROM
  (parsedfiles
CROSS JOIN UNNEST("speechsegments") t (segment))

```

3rd View – Extracted Entities

```
CREATE OR REPLACE VIEW extracted_entities AS

SELECT

"filename"

, "languagecode"
```

```
, "duration"
, "conversationtime"
, "formatedconversationtime"
, "segmentstarttime"
, "segmentendtime"
, "segmentspeaker"
, "speaker"
, "sentimentispositive"
, "sentimentisnegative"
, "sentimentscore"
, "displaytext"
, "textedited"
, "entity"."beginoffset"
, "entity"."endoffset"
, "entity"."score"
, "entity"."text"
, "entity"."type"
FROM
(conversation_segments
CROSS JOIN UNNEST("entitiesdetected") t (entity))
```

```
CREATE OR REPLACE VIEW extracted_entities AS
SELECT
  "filename"
, "languagecode"
, "duration"
, "conversationtime"
, "formatedconversationtime"
, "segmentstarttime"
, "segmentendtime"
, "segmentspeaker"
, "speaker"
, "sentimentispositive"
, "sentimentisnegative"
, "sentimentscore"
, "displaytext"
, "textedited"
, "entity"."beginoffset"
, "entity"."endoffset"
, "entity"."score"
, "entity"."text"
, "entity"."type"
FROM
  (conversation_segments
CROSS JOIN UNNEST("entitiesdetected") t (entity))
```

Creating the Amazon QuickSight Dataset

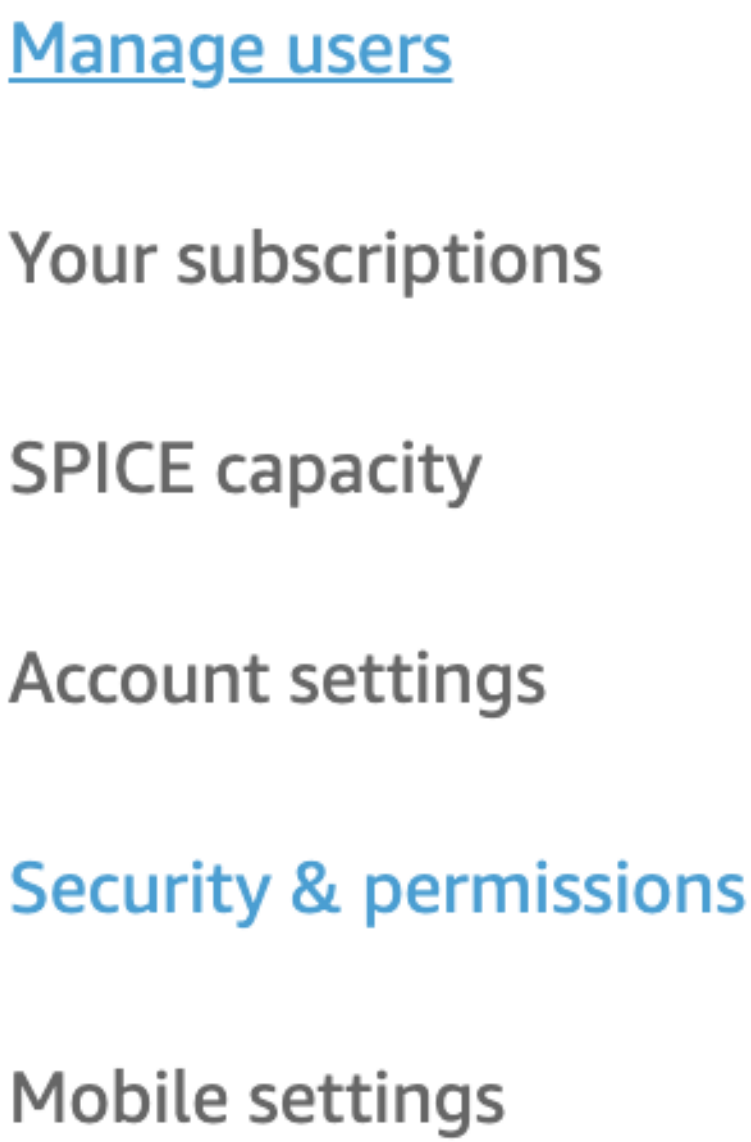
Now that we have our views created, we will create 3 datasets on Quicksight that will allow us to explore the 3 data dimensions we have available.

If you haven't signed up for Amazon Quicksight, follow the process presented when you visit the console. Also make sure you enable Amazon Athena Permissions (they are available by default).

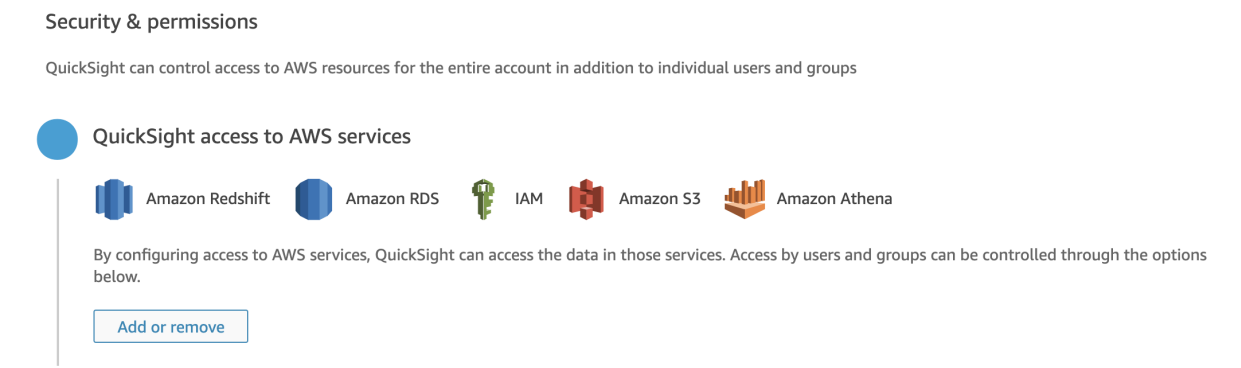
Also we need to make sure that Amazon Quicksight has access to the S3 bucket where your transcriptions are.

1. Click on Manage QuickSight on the top right corner of your screen:

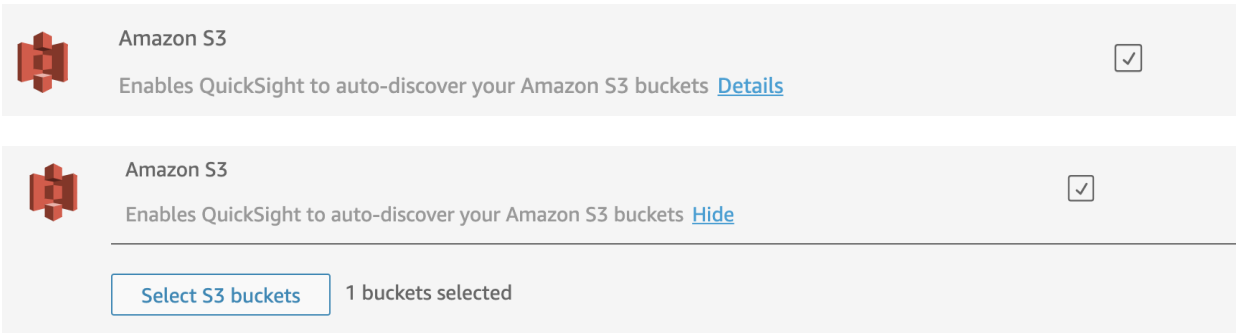
2. Click on "Security & Permissions":



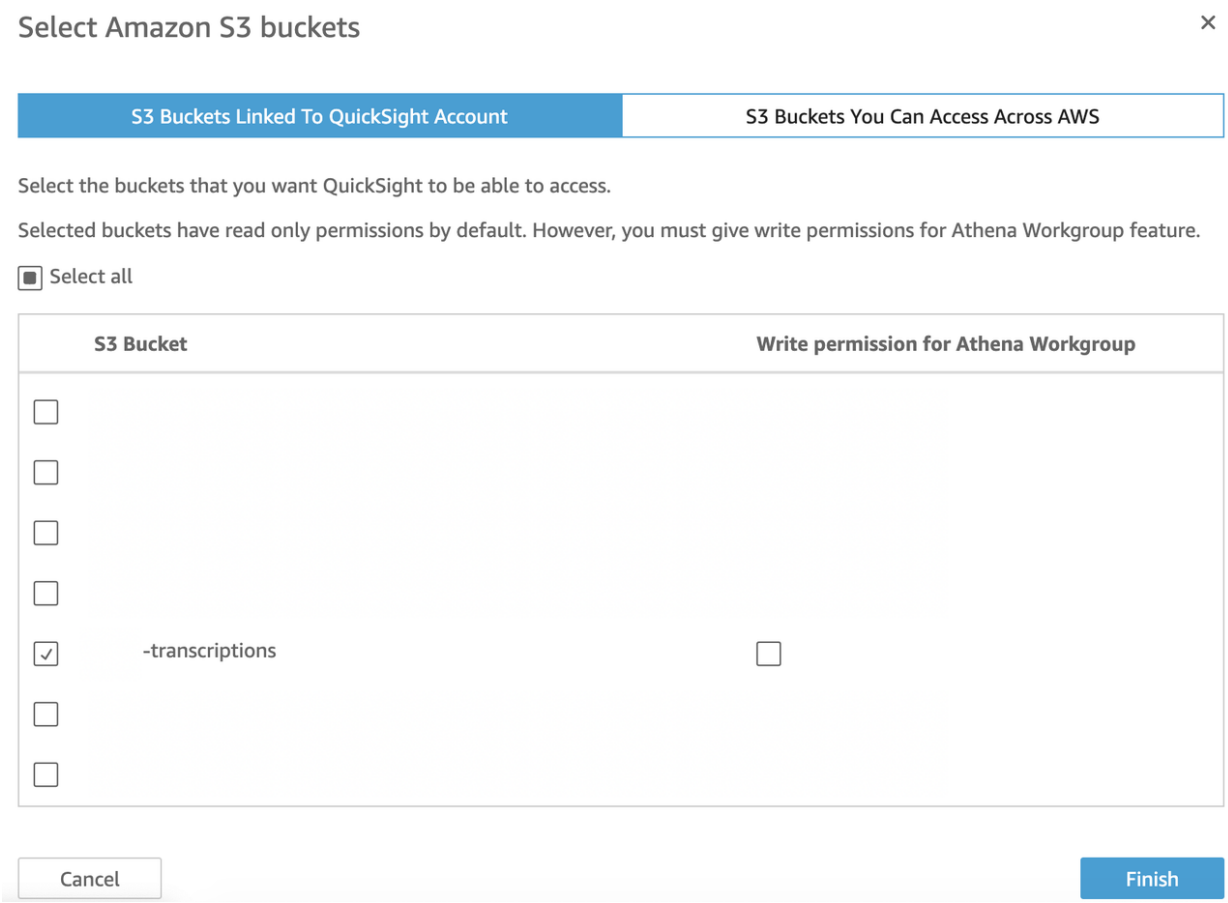
3. Click on “Add or remove”:



4. On “Amazon S3” click on “details” and then click on “Select S3 Buckets”:



5. Select the S3 bucket where your transcriptions are stored and click on “Finish”:



After the permissions are granted we can move forward to create the Datasets.

1. On Amazon Quicksight's, on the left menu, click on Datasets


Recent


My folders


Shared folders

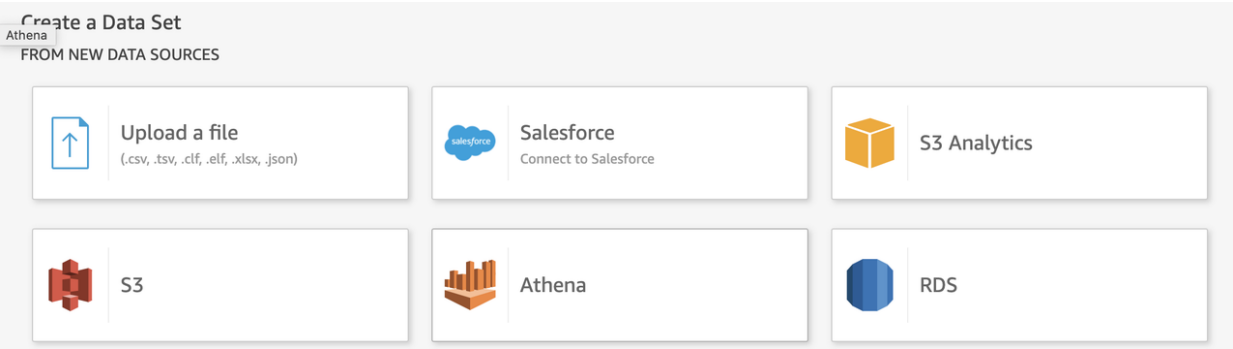

Dashboards


Analyses

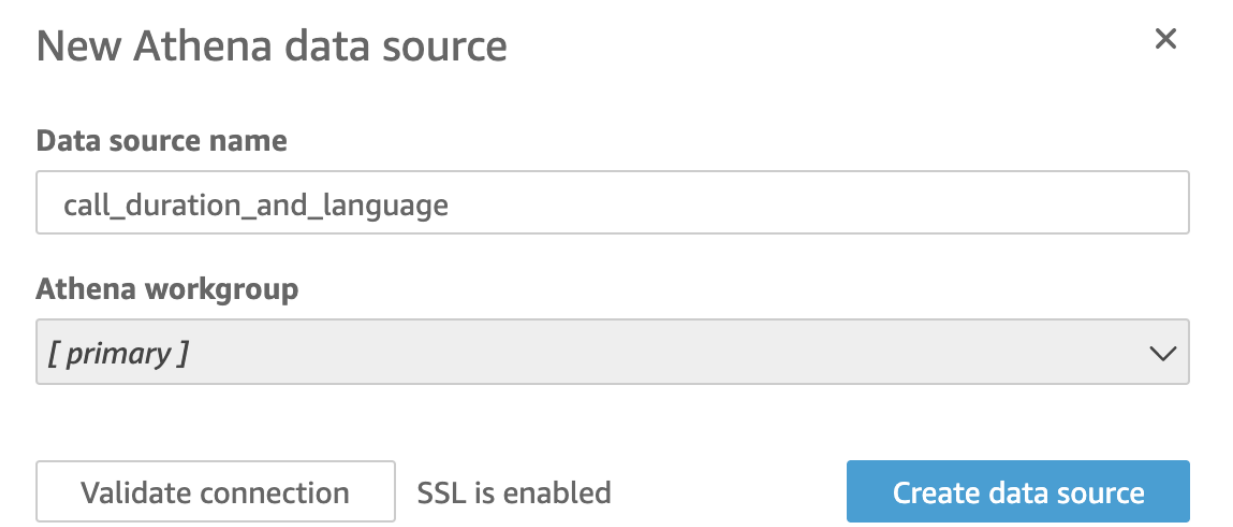

Datasets

2. Click on the “New Dataset” button on the top right

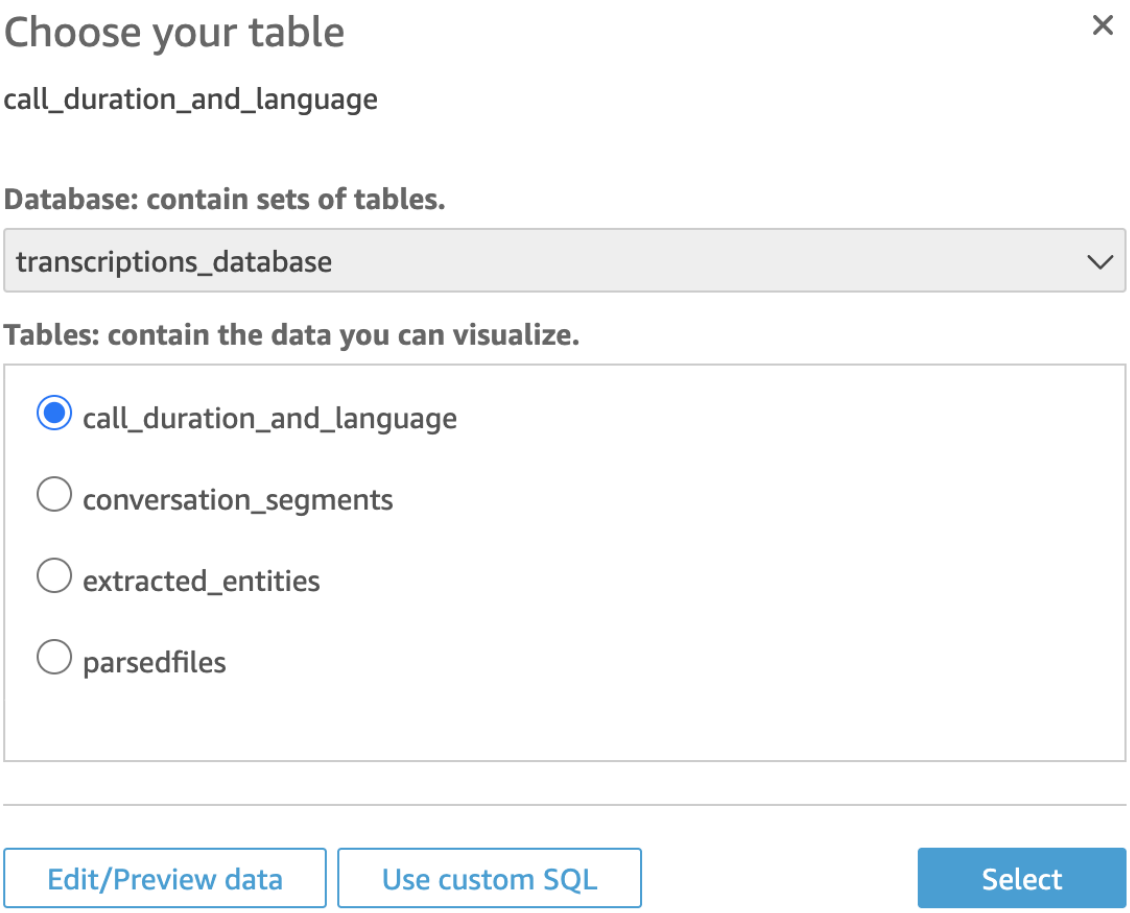
3. Click on “Athena”



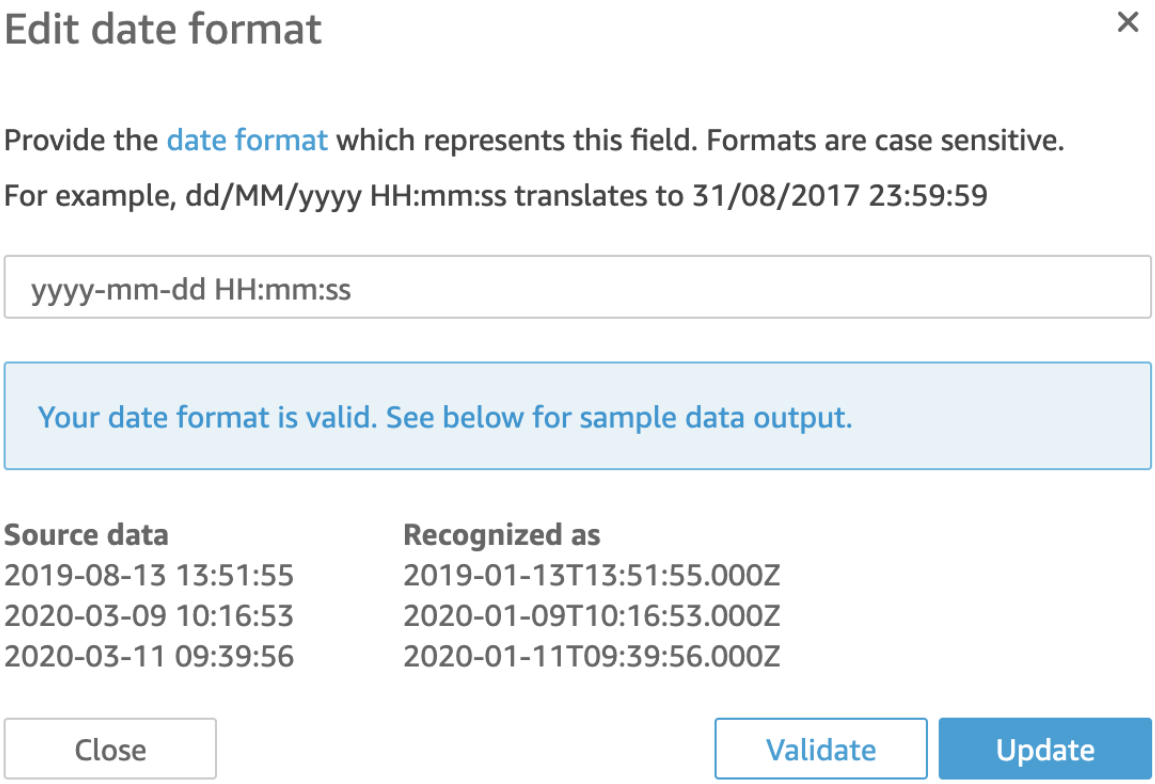
4. We will create one datasource for each view have, we will start with call_duration_and_language:



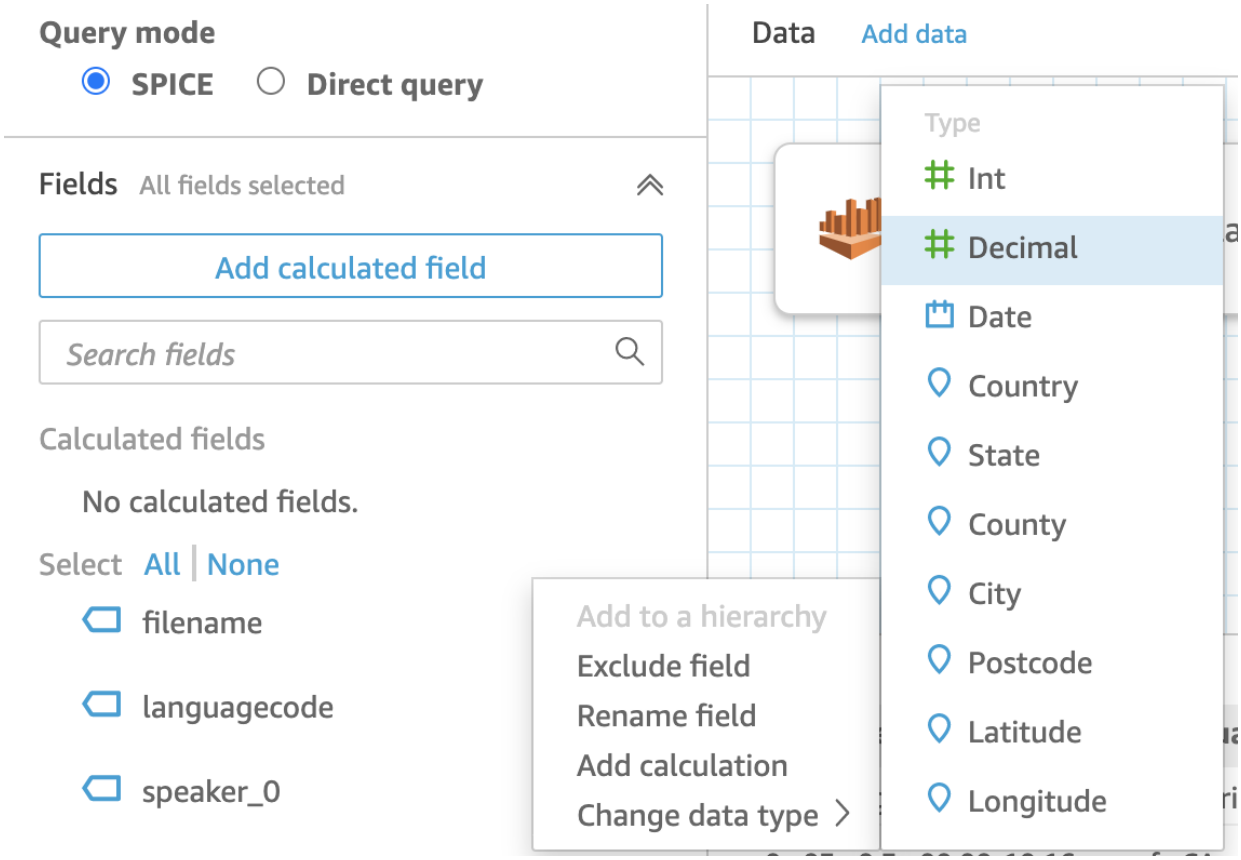
5. We will select the “transcriptions_database” and select “call_duration_and_language”, then we will click on “Edit/Preview data”



6. On the “Edit/Preview data” screen, select SPICE for query mode.
7. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update.



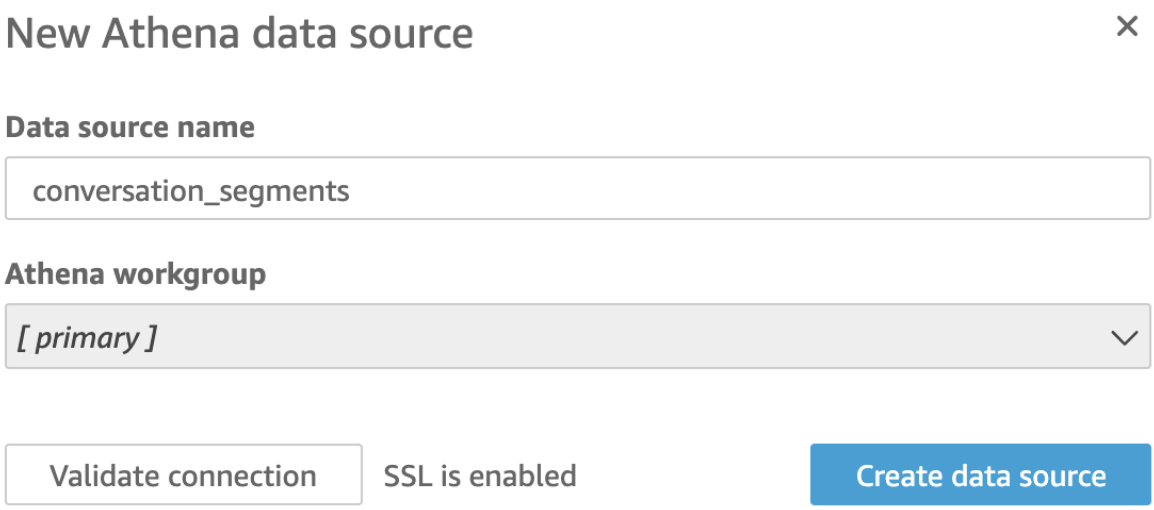
8. For the field duration, select the data type decimal



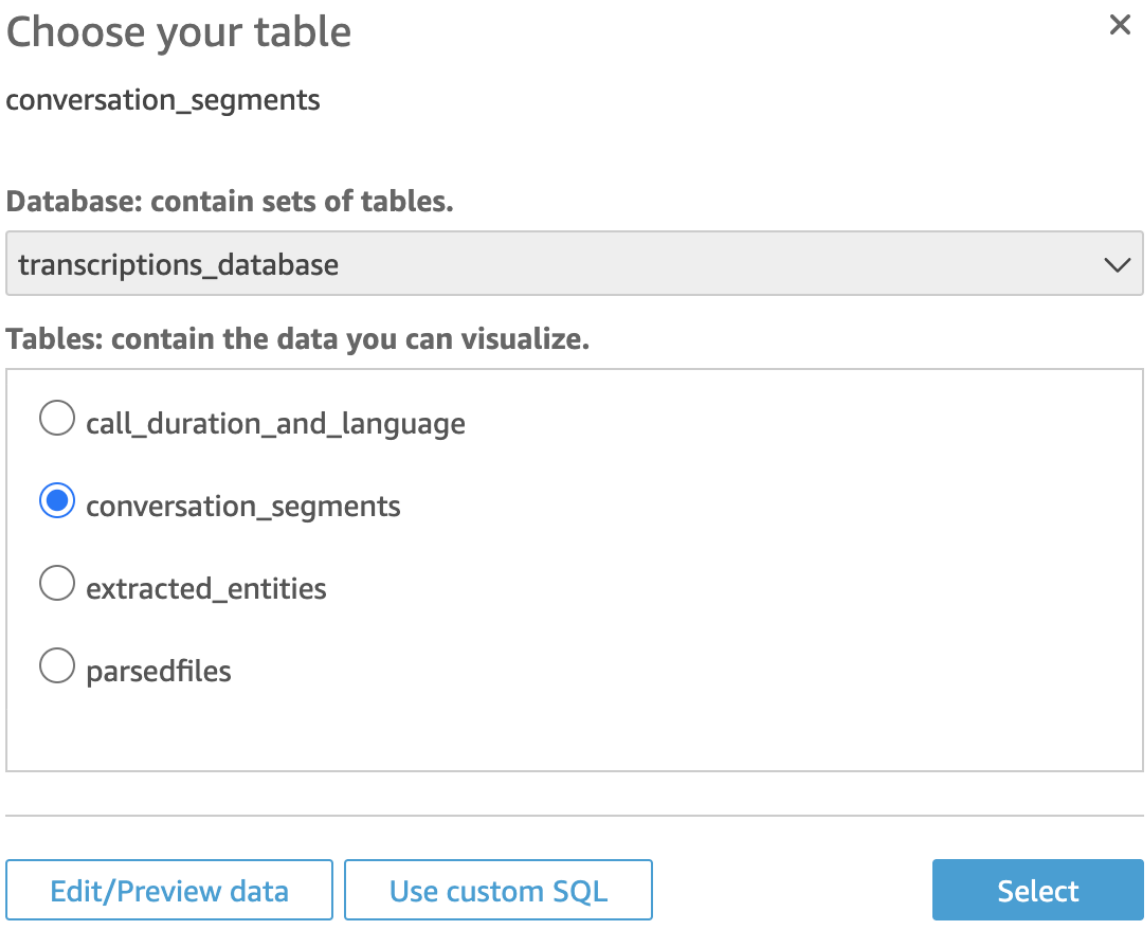
9. Click on “Save”

Next we are going to create the data source for the conversation_segments view:

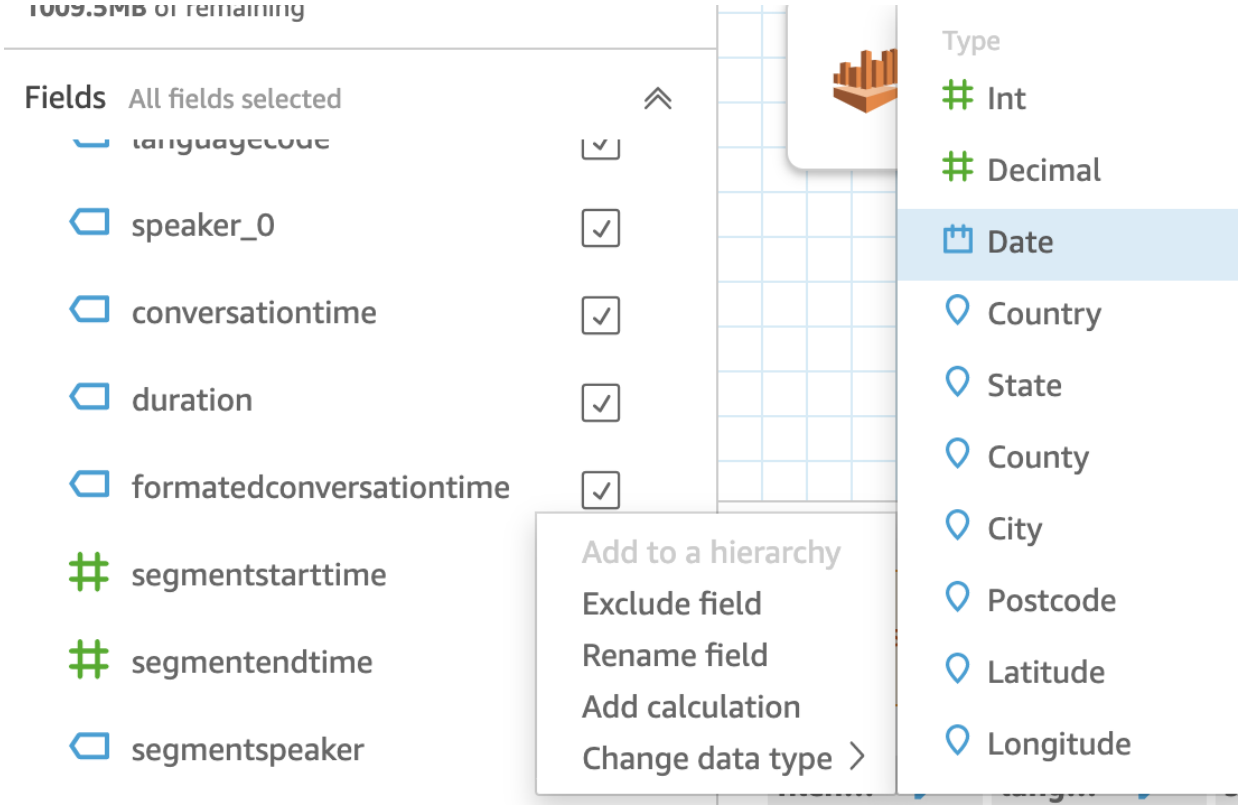
1. We will create a new data source, on Data Sets click on create Data set, select Athena and name it conversation_segments:



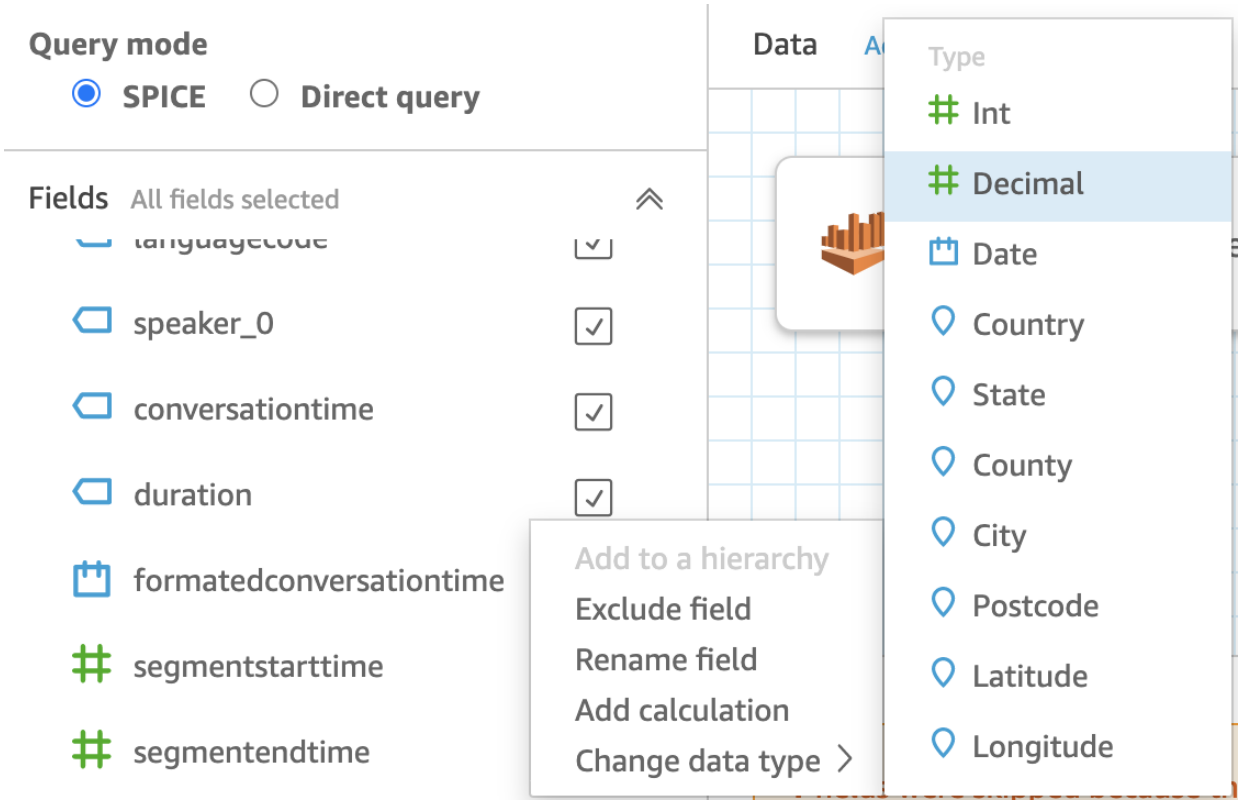
2. Select under transcriptions_database “conversation_segments” and click on Edit/Preview data:



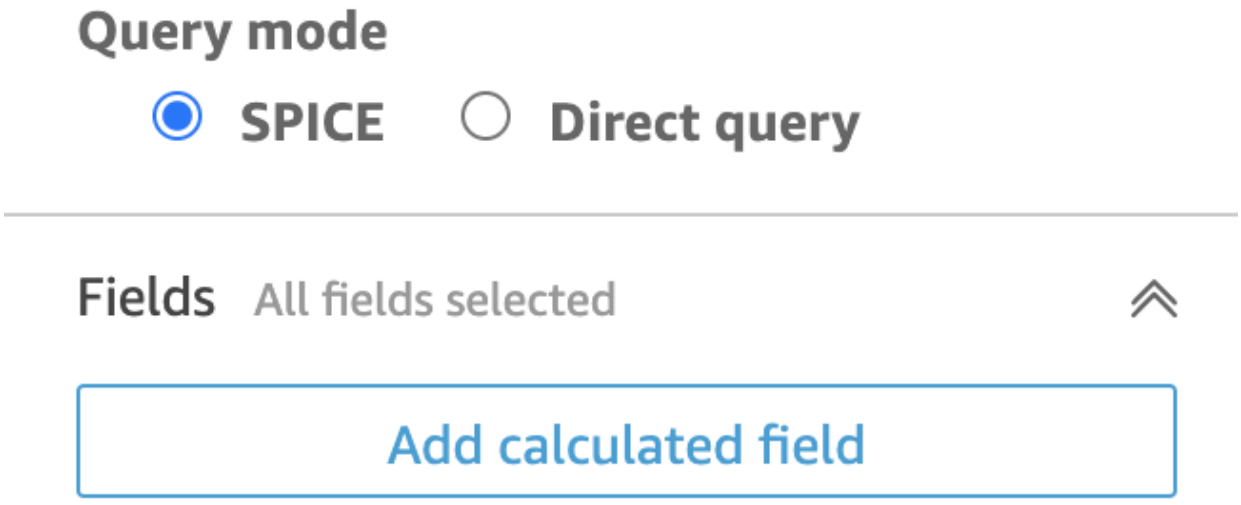
3. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update



4. Change the data type for the field “duration” to Decimal



5. We will create a calculated field called calc_CallSentiment, for this click on Add calculated field.



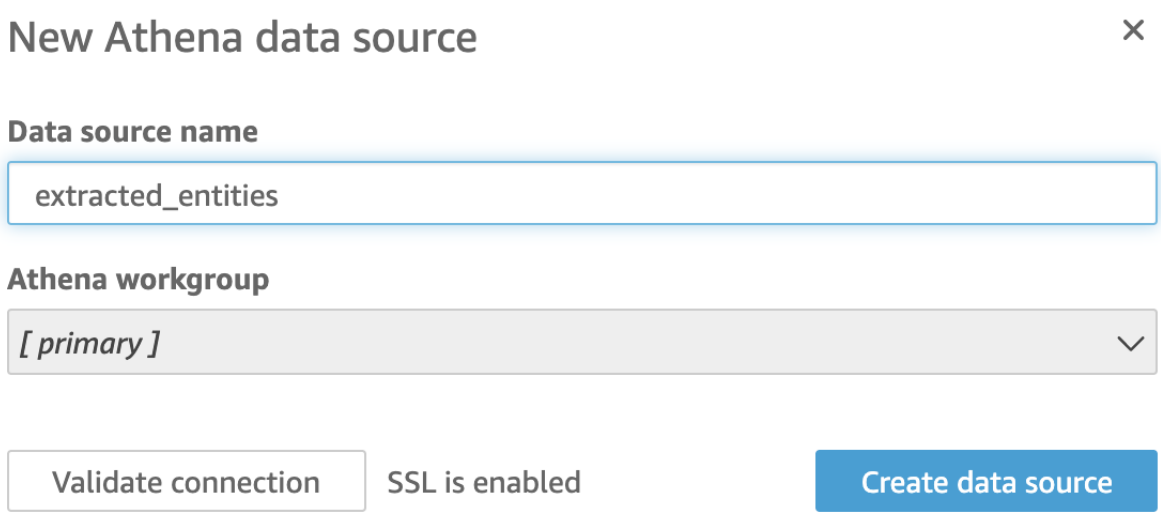
6. Enter the name and the content of the formula:

```
ifelse(({sentimentispositive} =1),"Positive",({sentimentisnegative}=1),"Negative","Neu
```

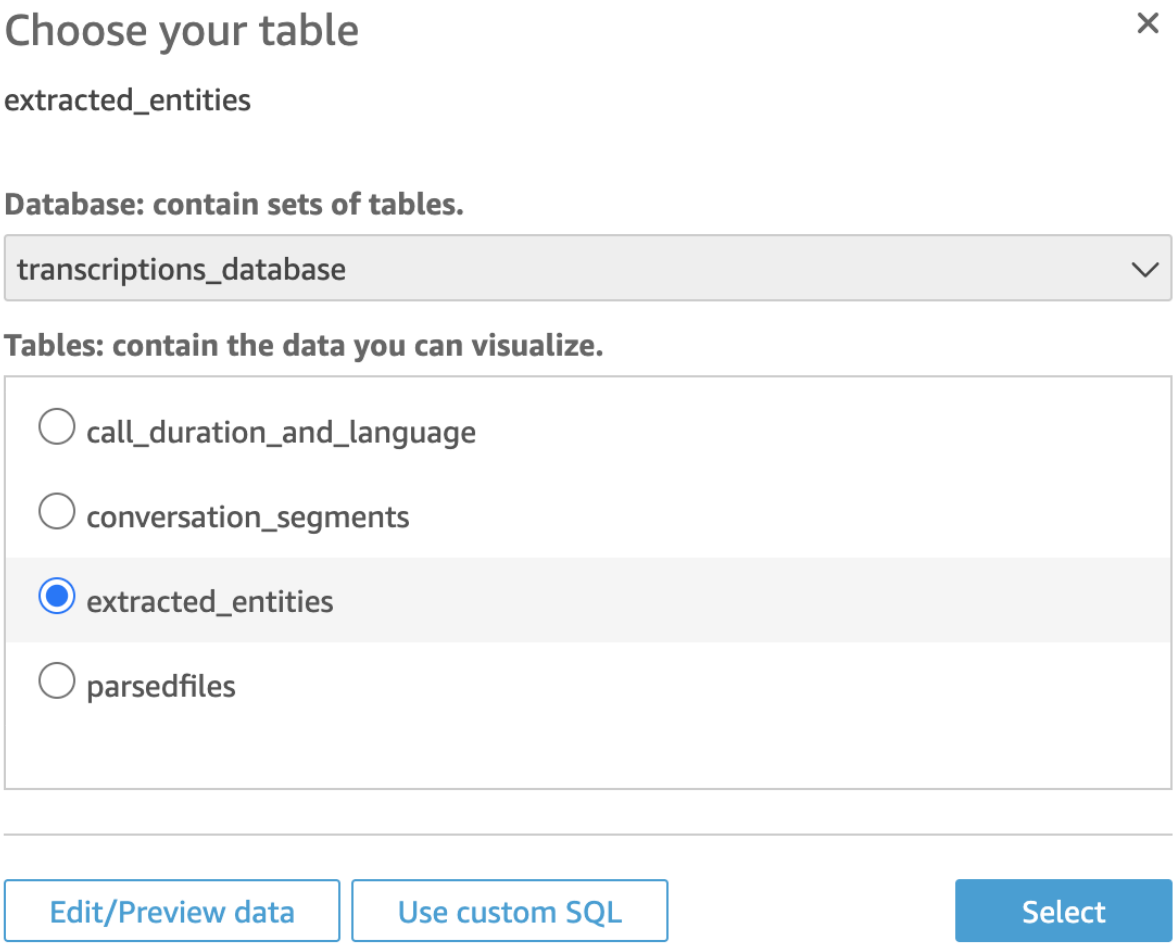
7. Click on Save

Finally we are going to create the view for the entities.

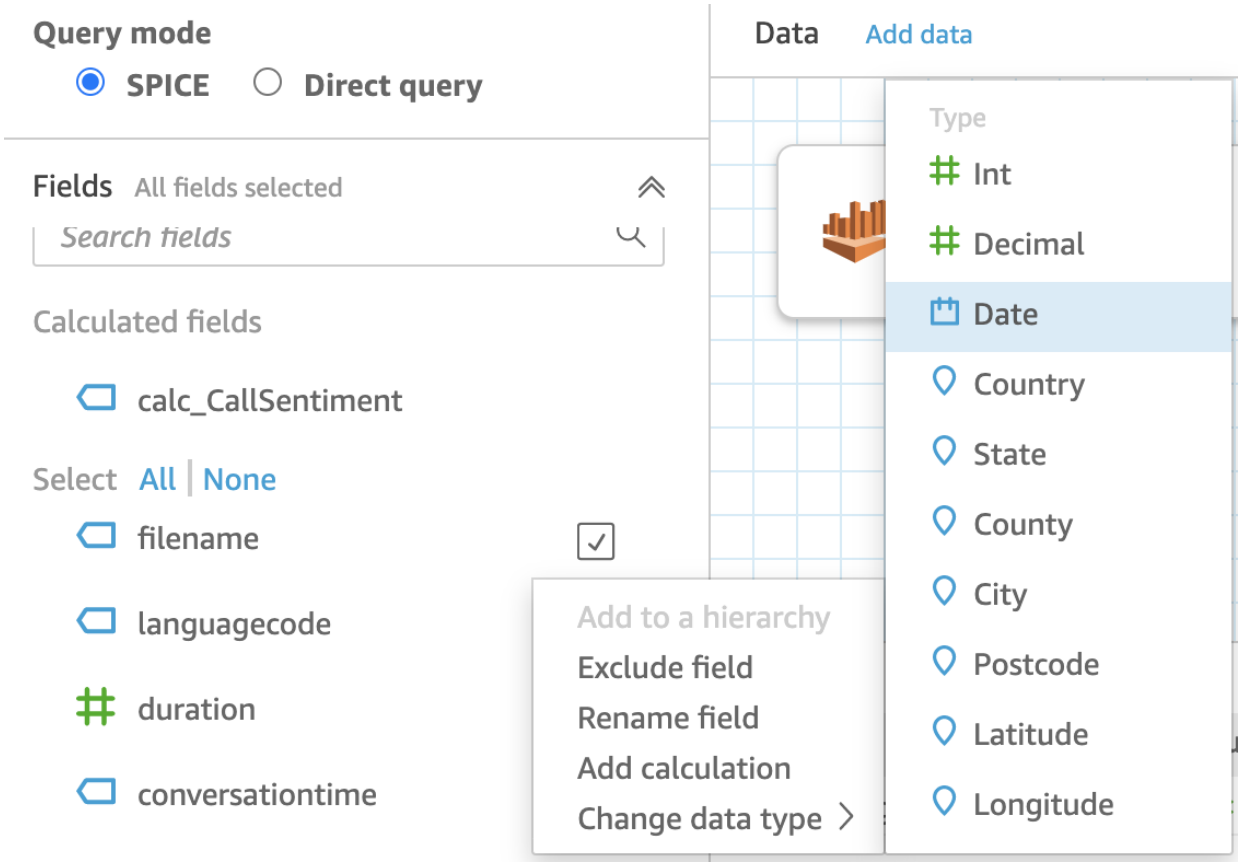
1. We will create a new data source, on Data Sets click on create Data set, select Athena and name it extracted_entities:



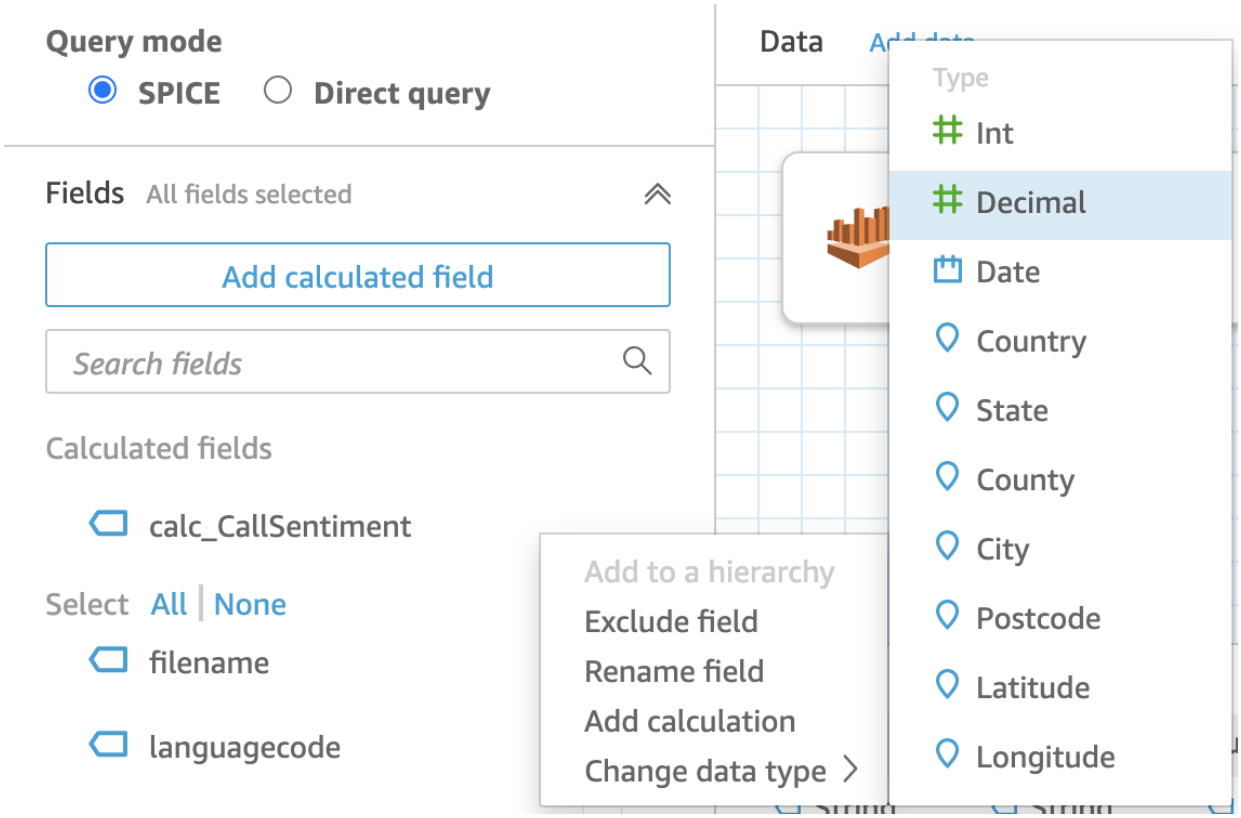
2. Select under transcriptions_database “extracted_entities” and click on Edit/Preview data:



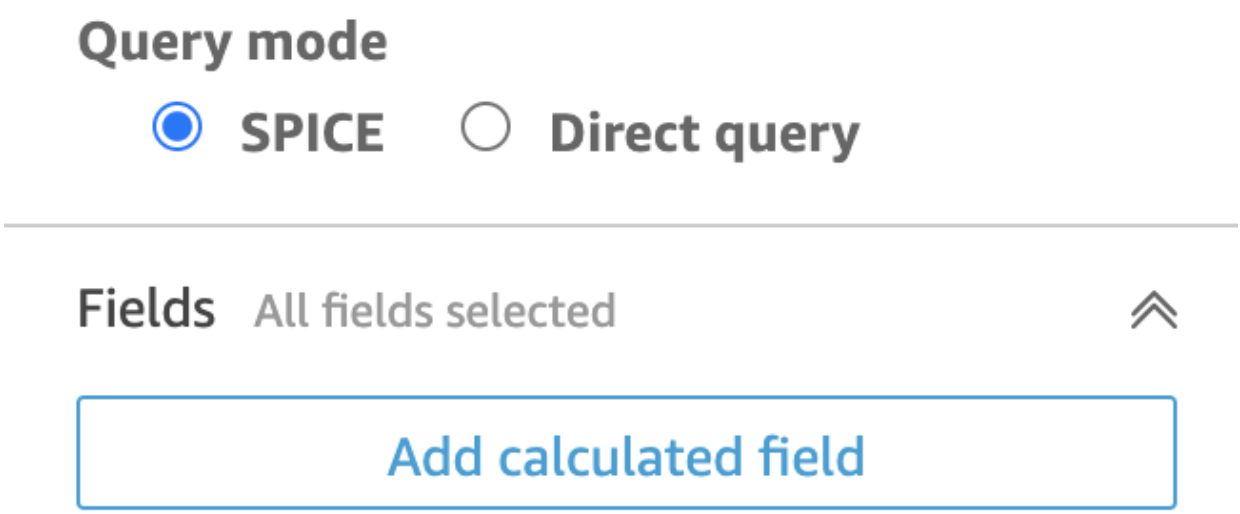
3. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update



4. Change the data type for the field “duration” to Decimal



5. We will create a calculated field called calc_CallSentiment, for this click on Add calculated field.



6. Enter the name and the content of the formula:

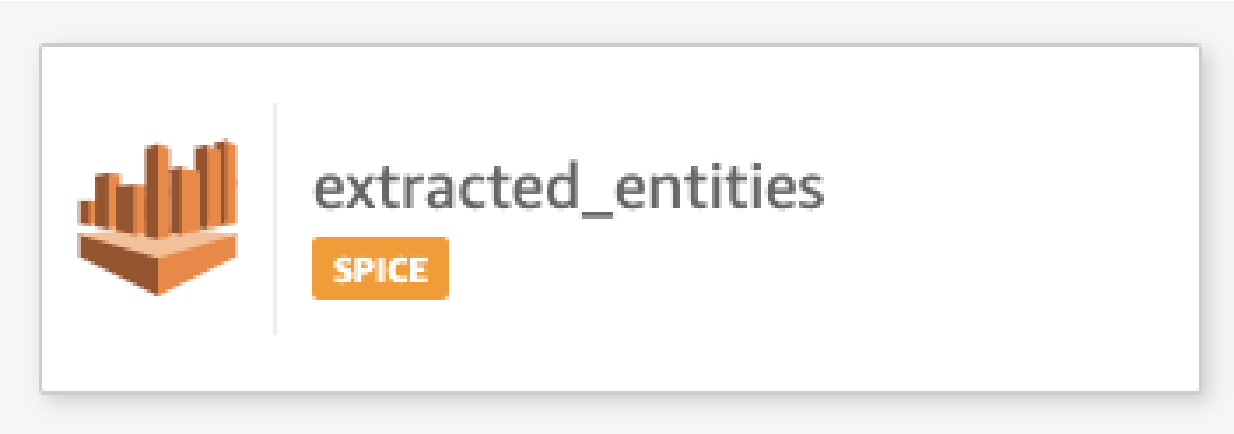
```
ifelse(({sentimentispositive} =1),"Positive",({sentimentisnegative}=1),"Negative","Neu
```

7. Click on Save

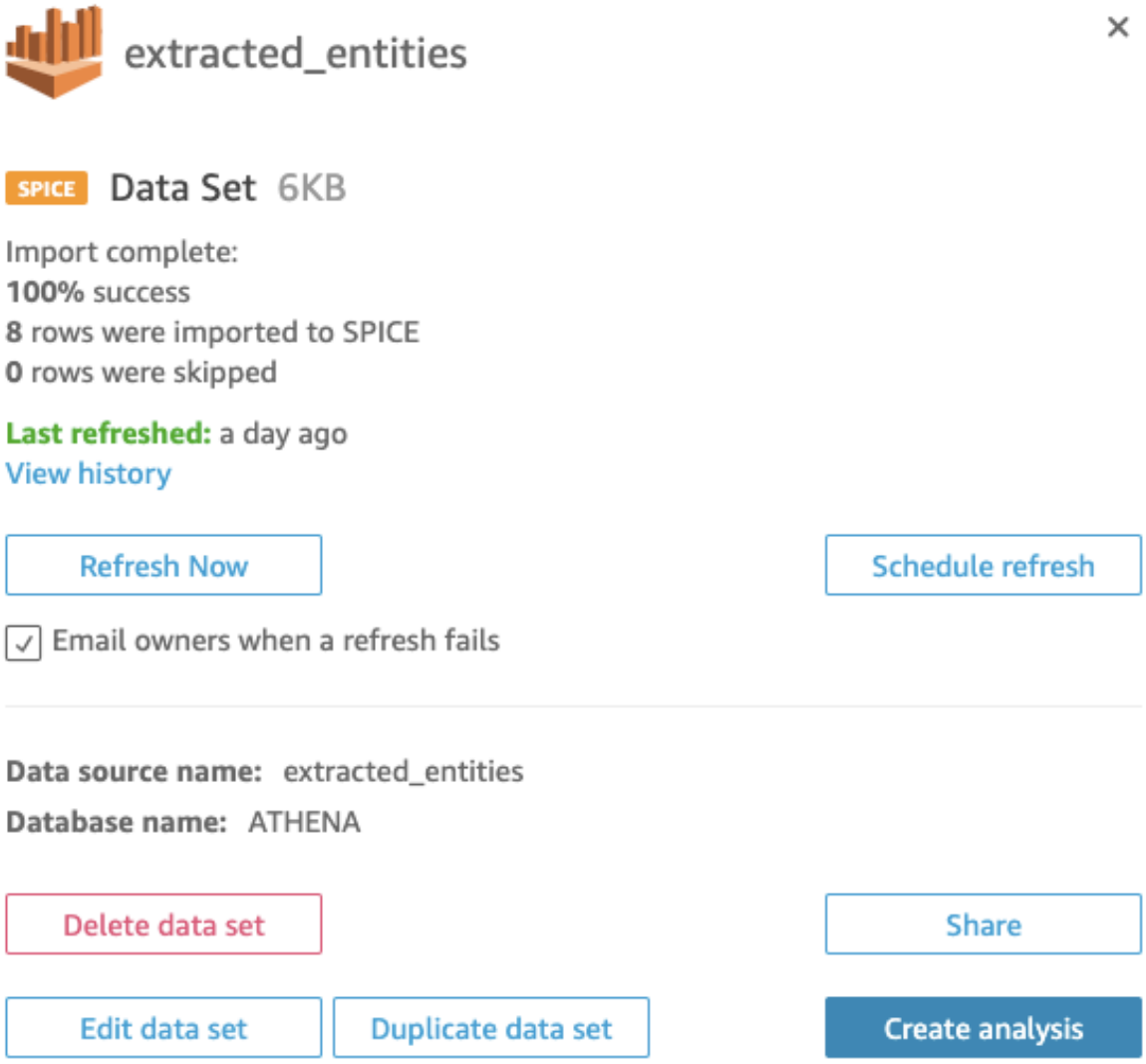
Creating sample Dashboards

Now that we have our datasets created in quicksight we can start creating our dashboards.
Click on "New Analysis"

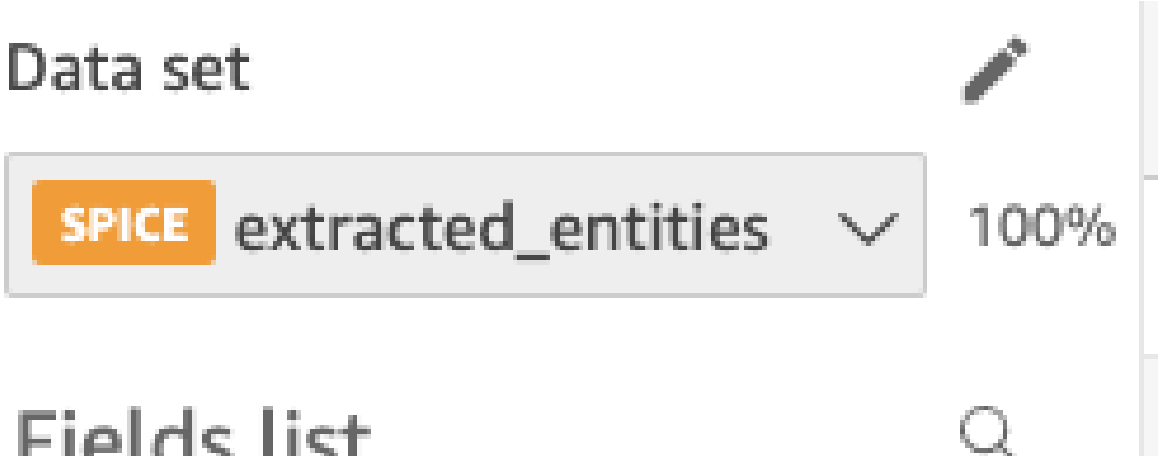
1. Select the "extracted entities" dataset

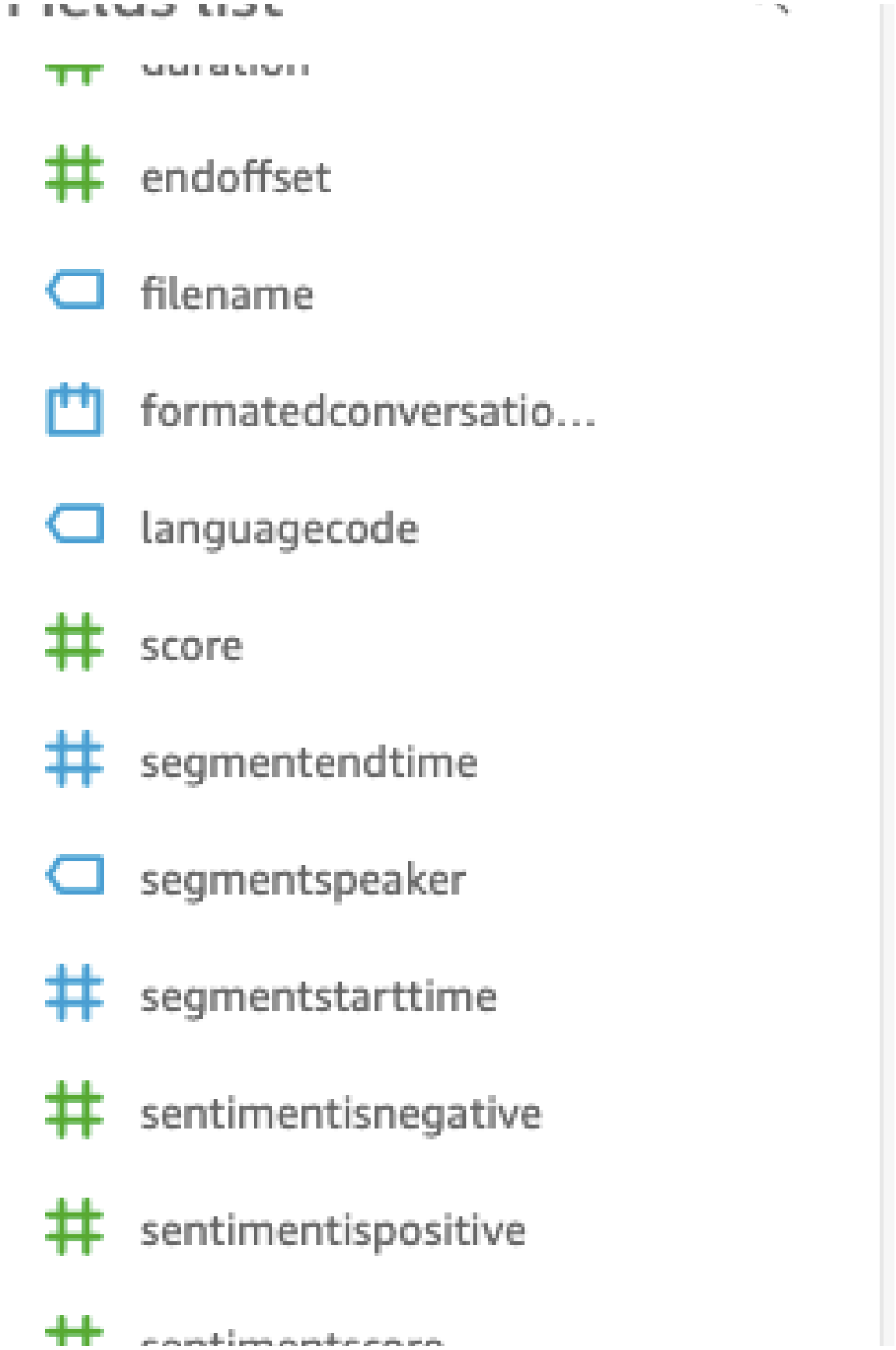


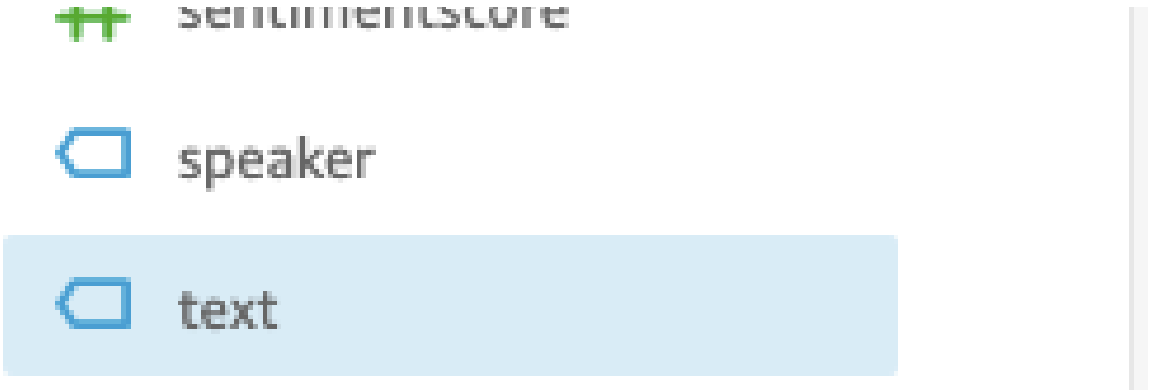
2. Click on "Create Analysis"



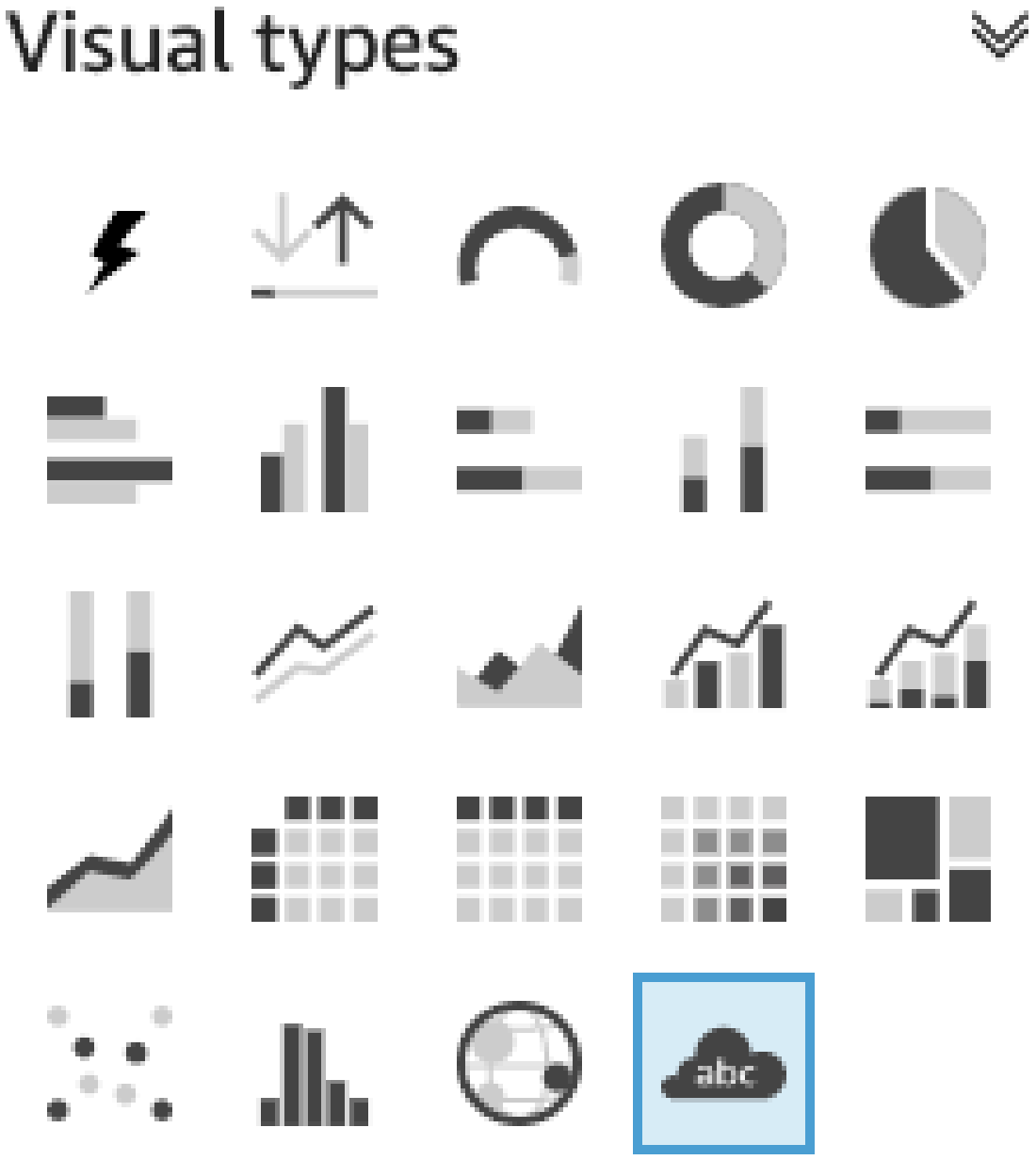
3. On the Field list, select the field "Text"




duration
endoffset
filename
formatedconversatio...
languagecode
score
segmentendtime
segmentspeaker
segmentstarttime
sentimentisnegative
sentimentispositive



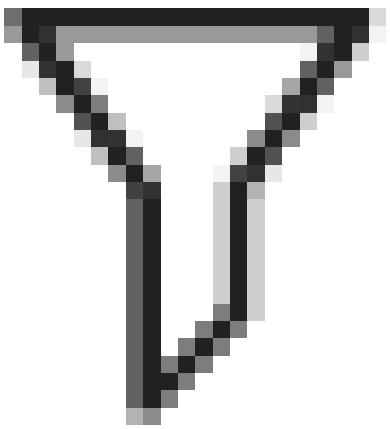
4. On the Visual types, select the word cloud



5. On the left toolbar, click on filter

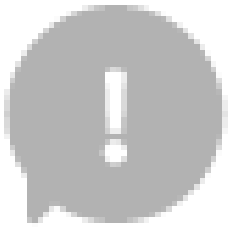
UUU

Visualize



Filter

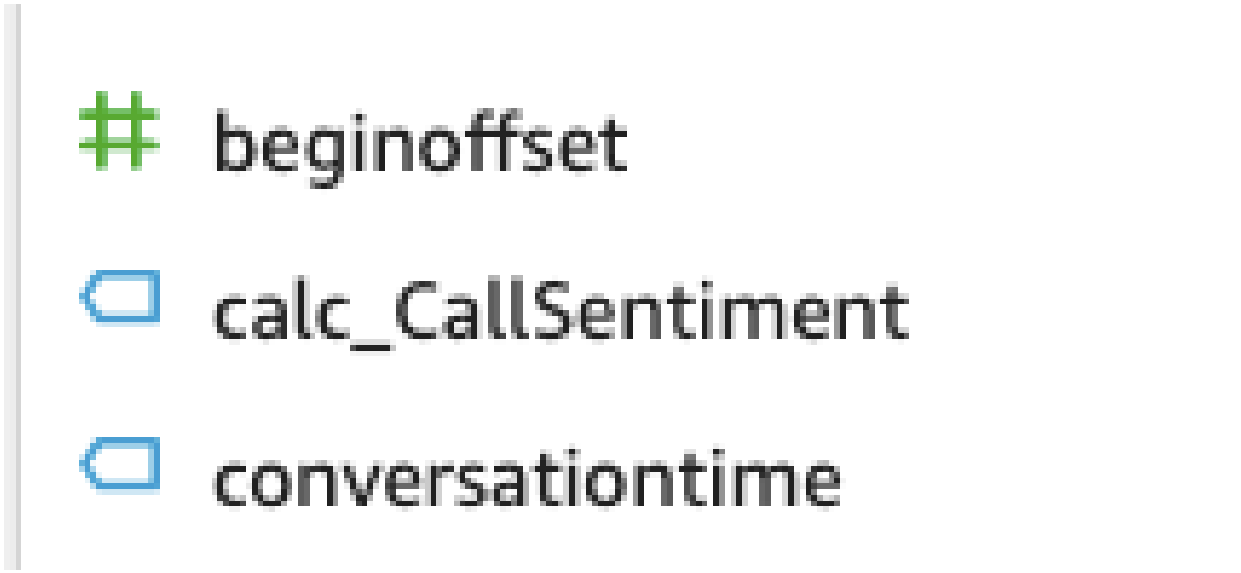
6. Click on "Create one..."



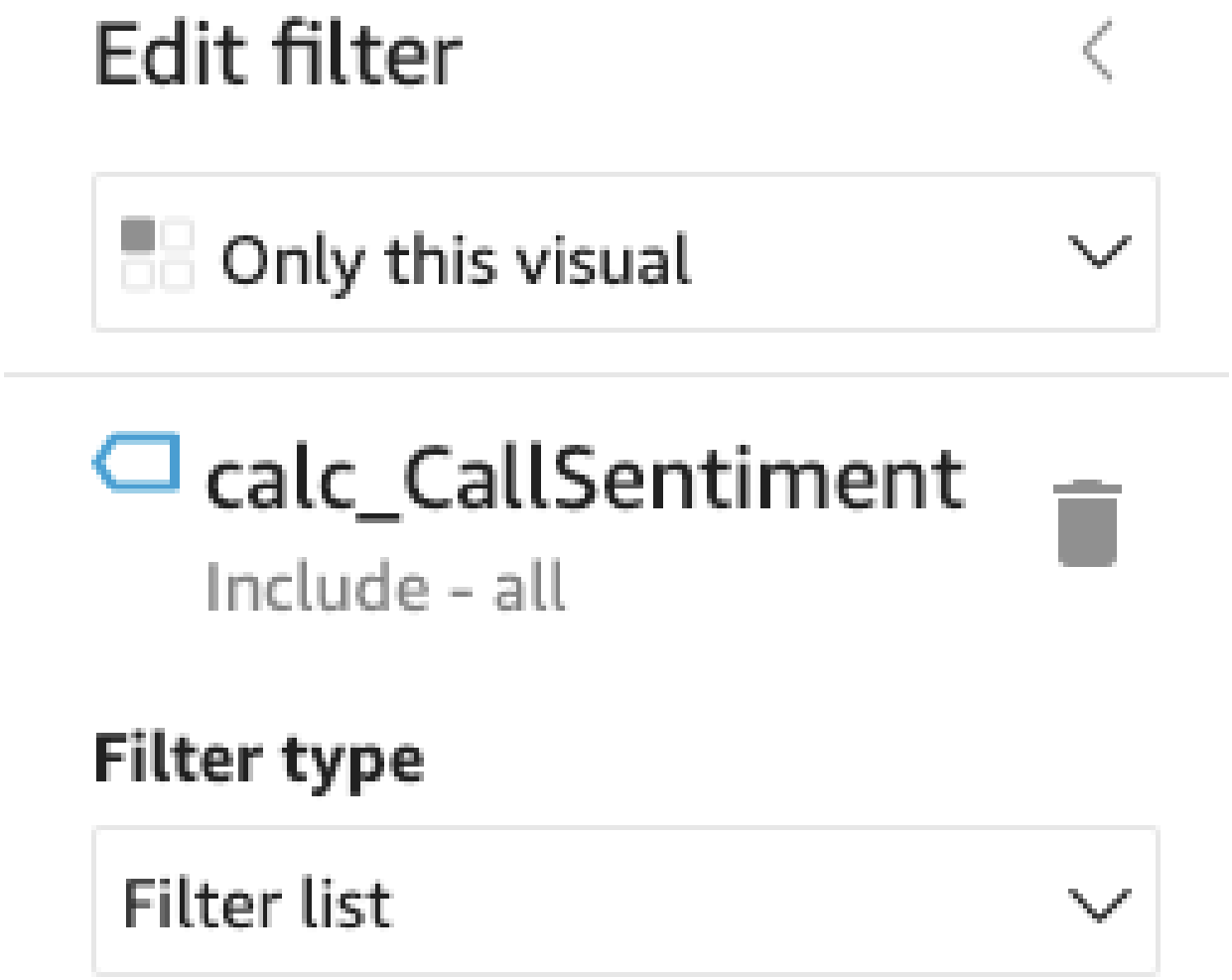
No filters for the selected visual

Create one...

7. Click on the field "calc_CallSentiment"



8. On the filter list select the Sentiment related to that entity and click



Looking for a value? Search or
Refresh this list.

Show selected values

Search values

Select all

Neutral

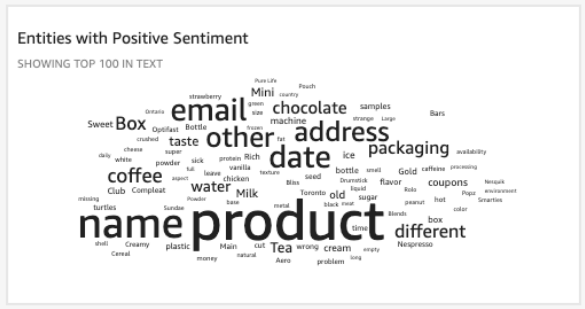
Positive

OR

Add filter condition

Note: There are limitations on how you can group filters.
Learn more

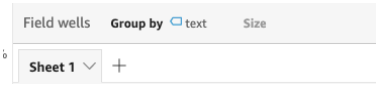
9. You can change the visual title by double clicking on it.



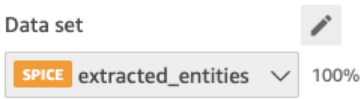
Other Examples

We can create visuals from the other datasets, for this example we will create an additional page to display visualizations.

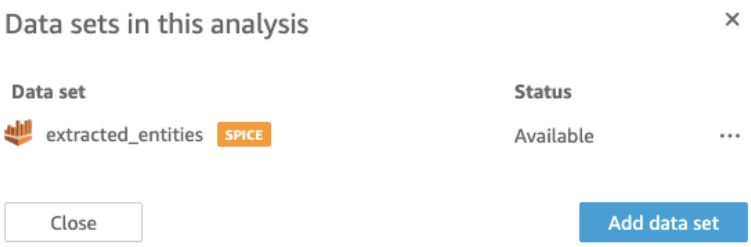
1. Click on the + sign next to the current Sheet name



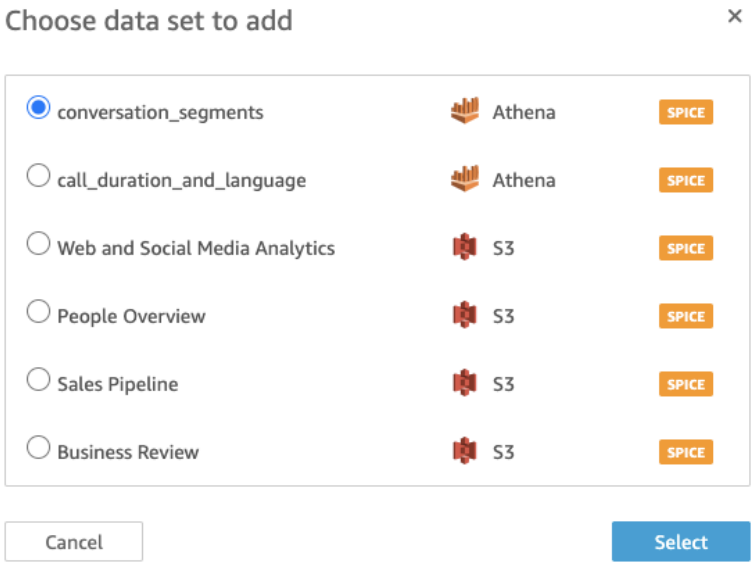
2. Click on the pencil icon on top of your current data set:



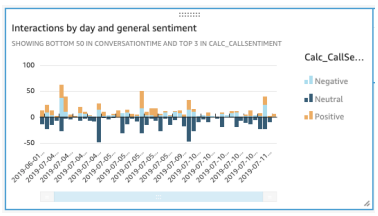
3. Click on Add Dataset

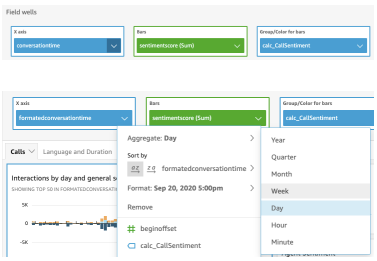


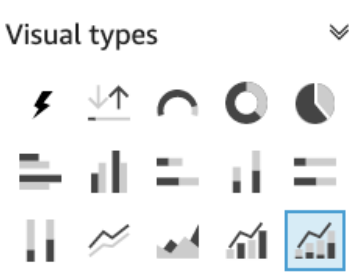
4. Select a dataset, in my case it is "conversation_segments", click on the Select button:

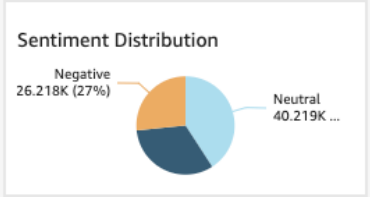


Conversation Segments









Language and Duration