Data Visualization

Objectives

By the end of this session, you should be able to:

- ❑ Explore Menu and Tabs
- ❑ Dashboards and Data Sources
- ❑ Building Charts
- ❑ Type of Charts
- ❑ Bar and Column charts
- ❑ Circular Charts
- ❑ Complex Charts
- ❑ Channels and Integration

Introduction

- To start the MongoDB Charts GUI application, users need to first log into the Atlas cloud web application. MongoDB Charts (the PaaS version) is bound to one Atlas project (the "per project" option), so if there are multiple Atlas projects, the user needs to select the currently active Atlas project.
- The MongoDB Charts option needs to be activated before the first use. To do so, you need to click on the **Activate Now** button to activate the Charts application.
- The option buttons are displayed on the left side of the application
  - Dashboards:
  - Data Sources:
  - Charts Settings:

Dashboards

- The cloud application automatically creates an empty dashboard for us. The default dashboard has a name, **User's Dashboard**
- There are a few buttons and options available in the dashboard context:
  - **Edit Title / Description**: This option is used to change the current title or description of the dashboard.
  - **Duplicate Dashboard**: This option copies the dashboard to a new one, with a different name
  - **Delete Dashboard**: This option removes the dashboard from MongoDB Charts.
  - **Lock**: This option assigns dashboard permissions for Atlas project users. This option is not useful for free-tier Atlas Charts, as MongoDB does not allow you to manage project users and teams with the free tier.

Data Sources

- Data sources represent the interface between MongoDB database structures and the MongoDB Charts presentation engine.
- A data source is a pointer to a specific database collection (or collections) from which the data is processed to create a graph.
- **Data Source Permissions**
- Complex MongoDB projects can have many developers and business users working together with Charts. In such cases, the Atlas user who creates a new data source may need to share it with other Atlas project users.
- Once a user creates a new data source, they become the owner of that data source and can share it with other project members by clicking on the **ACCESS** button in the **Charts** tab in the **Data Sources** window

Building Charts

- New charts can be created in MongoDB Charts using the Chart Builder. To start the Chart Builder, open a dashboard. You can open your own user dashboard by clicking on the **User's Dashboard** link in the dashboard tab.
- **Fields**
  - Each field has a name and a data type
    - The following is the list of data types in the Chart Builder:
    - A – String
    - # – Numeric (integer or float)
    - Date

Types of Charts

Type / Name	Description	Data Usage
 Bar and Column	These display a sequence of rectangular bars (vertical) or columns (horizontal). The bar chart can display single or multiple values, either grouped with different colors or stacked one on top of the other.	These are preferable for discrete values or categories—for example, high, medium, and low. A bar or column chart is effective for low-cardinality (*) variables with only a few categories or distinct values.
 Line and Area	These display lines between discrete points on the graph. This chart type can also display multiple values, either as separate lines or stacked one on top of the other.	These are preferable for continuous variables or variables with many different values. For example, a line chart is especially useful to display how values change over time.
 Grid	These are graphs represented in the form of regular tiles that display data on a grid with two axes.	These are preferable for datasets without a set trend or a category. A grid graph is also called a "heatmap" because each tile can be represented with a different color intensity.
 Circular	A circular chart (also called a donut or pie chart) is a round, colored circle, which is often sub-divided into slices. There are no axes—horizontal or vertical. Slices can have values or percentages from the whole pie value.	Pie charts provide a full perspective for one domain and illustrate how the total value is distributed. For example, you can use a pie chart to display the number of sales per category: wholesales, direct sales, retail, and more.
 Text	Here, the data is listed in an alphanumeric format, either free text or in tables, with lines and columns.	Sometimes, a table is more useful than a graph. When you need to know precise values or to compare pairs of values, a table is the perfect solution.
 Geospatial	This type of chart uses a map to display data. The geospatial graph is therefore applicable for data that can be correlated with geographical locations.	Geospatial charts are useful to display how data is geographically distributed. It is more visually impactful to present data on a geographical map for users who need to see the geographical distribution for specific data metrics; for example, total sales per location is a great graph for sales managers.

Types of Charts

- Each chart type could have one or more sub-types that are visual variations of the main chart and are useful in different presentations. Since a chart sub-type is dependent on the main chart type
- There are four different sub-types for bar and column charts
  - **Grouped** sub-type is useful to compare values in different categories
  - **Stacked** is useful to see the cumulated values for all categories
  - **100%**
  - **Colored**
- The following list gives a brief description of each tab:
  - **Encode**: This is for defining the chart channels.
  - •**Filter**: This is for defining data filters
  - **Customize:** This is used to define functional and aesthetical customizations of charts,

Bar and Column Charts

- Bar and column charts are probably the most common type of charts
- Bar charts are especially useful to represent aggregated values for categorical data. The main designation for bar charts is therefore data categorization or classification.

Description of how categorical data can be defined

- **Data classification**: This pertains to data that can be identified based on a category or label, for example, quality (high, average, low) or color (white, red, blue). This could also include a few distinct numerical values or numbers used as categories (not values).
- **Data binning**: This means grouping data in a category based on an interval. Binning is the method used to represent graphs for statistical analysis, called histograms.

Circular Charts

- Circular charts are colored round circles or semi-circles, often sub-divided into slices to represent values or percentages.
- The circular chart is also "unidimensional," which means that the graph can only represent a single set of scalar values and not values that can be represented in a Cartesian coordinate system.
- There are two sub-types of circular charts: **Donut** and **Gauge**:
  - **Donut**: This represents a full, colored circle (pie), which is divided into slices that represent values or percentages.
  - **Gauge**: This represents a semi-circle, with a ratio from the total.

Geospatial Charts

- Geospatial charts are a special category of charts wherein geographical data is the main ingredient for building the graph.
- Geospatial information can be specific or more general.
  - Precise longitude and latitude coordinates
  - An address that can be mapped using a map engine
  - Broader locations such as cities, regions, or countries.
- There are a few chart sub-types for geospatial charts as follows:
  - Choropleth charts: This chart shows colored geographical areas.
  - Scatter charts: This chart requires a precise address or location.
  - Heatmap charts: A heatmap displays colors with different intensities on a map

Complex Charts

- **Preprocessing and Filtering Data**

  Data preprocessing includes the following:

  - **Data filtering**: Filtering the data such that only certain documents are selected
  - **Data type change**: Modifying the data type so that it fits the Chart Builder better
  - **Adding new fields**: Adding custom fields that do not exist in the MongoDB database.

- **Filtering Data**

  - Data filtering allows users to select only a subset of documents from a MongoDB collection.

- **Adding Custom Fields**

  - Charts allows users to add custom fields that can be used to build charts.
  - Most of these custom fields are either derived or calculated using the source database values.

Complex Charts

- There are two types of fields that can be added:
  - **MISSED**: This option is used to add a field that is missing from the list of fields.
  - **CALCULATED**: This is used to add a new field that does not exist in the collection
- **Changing Fields**
  - MongoDB Charts allows users to change fields to a data type appropriate for chart plotting when the data from database is not the right data type.

Channels

- The encoding channels are one of the most important aspects of data visualization and it decides how the data is visualized in the chart.
- Each encoding channel has a name and a type. The channel name defines the target in the graph—that is, the end to which the channel will be used.

Channel	Accepted Data Type	Description
`+ category`	Numeric, Date, or String	This is a channel that accepts a categorical type of data, as described before in the *Bar Charts* section. A chart element will be plotted for each unique occurrence of a value. The dataset must have unique values (no duplicates). Values in this category channel can be sorted if required.
`+ value`	Numeric or Date	This channel accepts data values (not categories). The dataset can have duplicate values. One example where this channel type is used is in continuous line charts.
`+ aggregation`	Numeric, Date, or String	An aggregation expression is applied to group elements based on a specific category. We will learn more details about these in the next section, *Aggregation and Binning*.
`+ geopoint`	Geodata (X, Y)	This is used for longitude and latitude coordinates that can be mapped on a geographical map. This type of channel is valid only for geospatial scatter and heatmap charts.

Channels

Aggregation and Binning

- Data in one channel is often combined with a category data type channel so that it can calculate aggregate values for each category.
- There are a few types of aggregations

Data Type	Aggregation Function
Numerical fields	`SUM`: This function calculates the sum of all elements. `MIN`: This function finds the minimum value. `MAX`: This function finds the maximum value. Other statistical functions (example: `Standard Deviation`).
String and Date fields	`COUNT`: This function counts the total number of occurrences. `DISTINCT`: This function counts the distinct occurrences.

Integration

- How the dashboards and charts can be used for presentations and applications?
- One option is to save the charts as images and integrate them into MS PowerPoint presentations or to publish them as web page content, it has one main disadvantage in that the chart image is static
- Another option is to use MongoDB Charts as a presentation tool. This option guarantees that charts are refreshed and rendered each time the database is updated.
- MongoDB Charts has an option to publish charts as dynamic content for web pages and web applications. It can also be easily integrated into an MS PowerPoint presentation called as **Embedded Charts** and allows charts to be automatically refreshed after a pre-established time interval.

Integration

- Embedding charts is an option you can use to share charts outside of the MongoDB Charts.
- There are three methods to share charts:
  - **Unauthenticated:** Users are not required to authenticate themselves to access the chart.
  - **Authenticated**: Users are required to authenticate themselves to access the chart.
  - **Verified Signature**: Users are required to provide a signature key to access the chart
- Choosing the method depends on data security requirements and policies.
- The **Unauthenticated** method is acceptable for learning or testing with non-sensitive data.
- The **Verified Signature** method should always be used for integration with other applications.

Summary

- In this session, you learned about:
- Earlier chapters focused on the Charts user interface rather than MongoDB programming. The results that can be achieved using the Atlas cloud Charts module are impressive, allowing users to focus on data rather than programming and presentation.
- There are various chart types and sub-types to choose from, which makes Charts both more effective and easier to work with. MongoDB Charts can also be easily integrated with other web applications using the **EMBED CODE** option.
- In the next chapter, we will look at a business use case in which MongoDB will be used for managing the backend.