Data Visualisation with Google Charts

Present processed SQL data as pie, bar and line charts using Google Charts.

- Creating Charts with Google Charts
- Pie Charts from SQL Data
- Bar Charts from SQL Data
- Line Charts from SQL Data

Creating Charts with Google Charts

Google Charts is a free JavaScript library that can create professional graphs and charts using your data.

Google Charts can display:

- Pie Charts
- Bar Charts
- Column Charts
- Line Charts
- Area Charts

In this tutorial you will create your first chart using sample data.

Step 1 - Load the Google Charts Library

Add the following line inside the `<head>` section of your webpage.

```
<script src="https://www.gstatic.com/charts/loader.js"></script>
```

Step 2 - Create a Chart Container

Add a `div` where the chart will be displayed.

```
<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>
```

Step 3 - Create the Chart

Add the following JavaScript before the closing `</body>` tag.

```
<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =
        google.visualization.arrayToDataTable([

        ['Risk Level', 'Count'],

        ['Low',15],

        ['Medium',8],

        ['High',3]

    ]);


    var options = {

        title:'Security Events by Risk Level'

    };
```

```
    var chart =
        new google.visualization.PieChart(

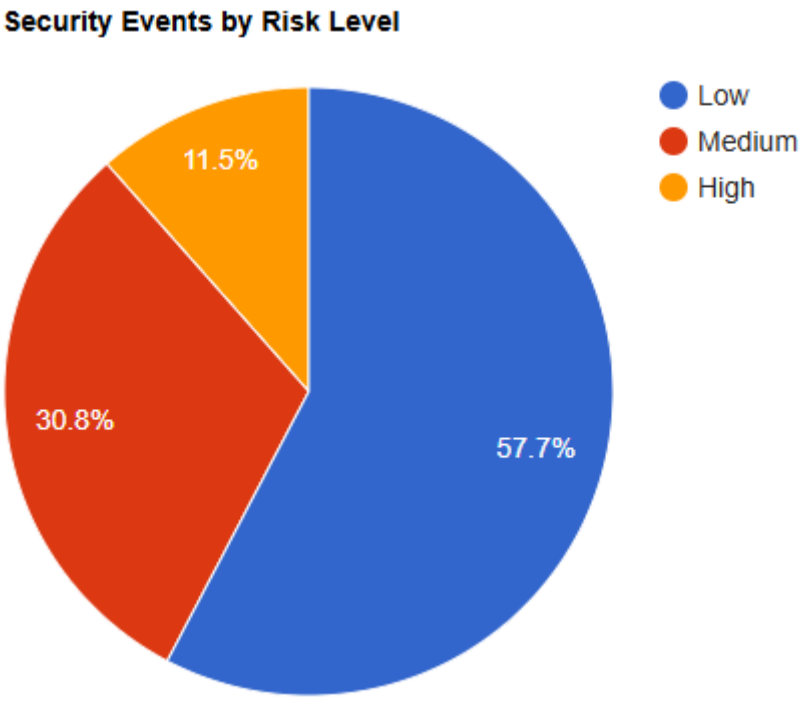
        document.getElementById('chart_div')

    );


    chart.draw(data, options);

}

</script>
```



Understanding the Data

Google Charts uses an array to store chart data.

```
[
 ['Risk Level', 'Count'],
```

```
  ['Low',15],

  ['Medium',8],

  ['High',3]
 ]
```

The first row contains the headings.

The remaining rows contain the data values.

Changing the Chart Type

The chart type is controlled by:

```
google.visualization.PieChart
```

You can change it to:

Chart	Class
Pie Chart	PieChart
Column Chart	ColumnChart
Bar Chart	BarChart
Line Chart	LineChart
Area Chart	AreaChart

Example:

```
google.visualization.BarChart
```

will display the same data as a bar chart.

Complete Example

```
<!DOCTYPE html>

<html>

<head>

    <title>Google Charts Example</title>

    <script src="https://www.gstatic.com/charts/loader.js"></script>

</head>

<body>

<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>



<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =
        google.visualization.arrayToDataTable([

        ['Risk Level', 'Count'],

        ['Low',15],

        ['Medium',8],
```

```
        ['High',3]

    ]);


    var options = {

        title:'Security Events by Risk Level'

    };


    var chart =
        new google.visualization.PieChart(

        document.getElementById('chart_div')

    );


    chart.draw(data, options);

}

</script>

</body>

</html>
```

Accessibility and data note

Provide the values in an HTML table or concise text summary as well as the chart. Do not rely on colour alone, and give the chart container an accessible name where practical. Google Charts loads JavaScript from Google; use fictional or non-personal summary data only.

Pie Charts from SQL Data

In this tutorial, the chart data will come directly from a MySQL database.

The example below uses the `cyber_security_events` table.

Step 1 - Create the SQL Query

The following query counts how many events belong to each risk level.

```
SELECT

    risk_level,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY risk_level;
```

Example result:

risk_level	total
Low	15
Medium	8
High	3

Step 2 - Retrieve the Data with PHP

```php
<?php

$sql = "

SELECT

    risk_level,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY risk_level

";

$result = $pdo->query($sql);

$data = [];

while($row = $result->fetch())
{

    $data[] = [

        $row['risk_level'],

        (int)$row['total']

    ];

}

?>
```

Step 3 - Convert the PHP Array to JSON

Google Charts uses JavaScript arrays.

PHP can convert data into JavaScript using:

```
<?php

echo json_encode($data);

?>
```

Step 4 - Create the Chart

```
<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data = google.visualization.arrayToDataTable([

        ['Risk Level','Count'],

        ...<?php echo json_encode($data); ?>

    ]);
```

```
    var options = {

        title:
        'Security Events by Risk Level'

    };

    var chart =

        new google.visualization.PieChart(

            document.getElementById('chart_div')

        );

    chart.draw(data, options);

}

</script>
```

Step 5 - Add the Chart Container

```
<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>
```

Complete Example

```
<?php

$sql = "

SELECT
```

```
    risk_level,
    COUNT(*) AS total
FROM cyber_security_events
GROUP BY risk_level
";
$result = $pdo->query($sql);
$data = [];
while($row = $result->fetch())
{
    $data[] = [
        $row['risk_level'],
        (int)$row['total']
    ];
}
?>
<!DOCTYPE html>
<html>
<head>
<script src="https://www.gstatic.com/charts/loader.js"></script>
</head>
```

```
<body>

<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>

<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =

        google.visualization.arrayToDataTable([

        ['Risk Level','Count'],

        ...<?php echo json_encode($data); ?>

    ]);

    var options = {

        title:
        'Security Events by Risk Level'

    };

    var chart =

        new google.visualization.PieChart(

            document.getElementById('chart_div')
```

```
        );

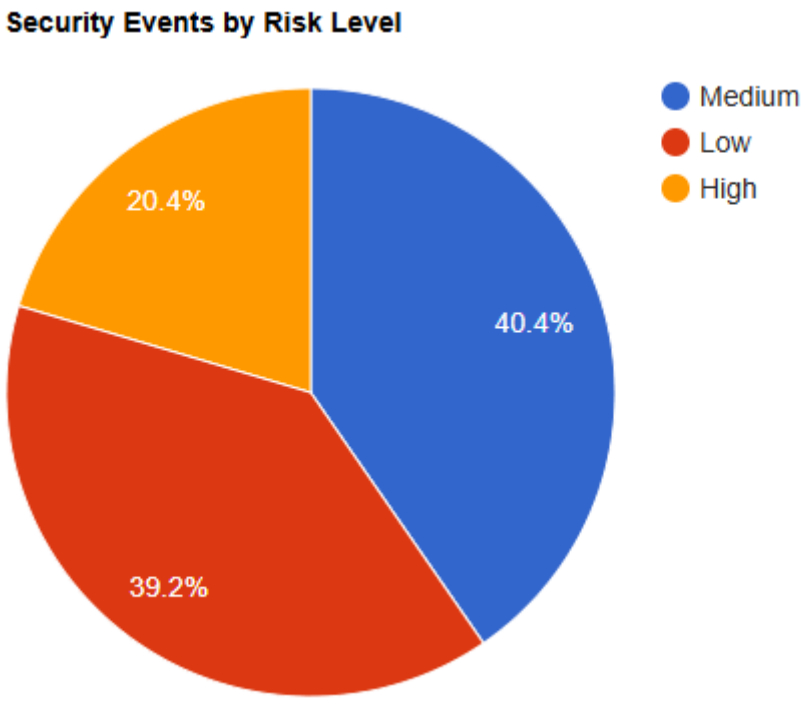
    chart.draw(data, options);

}

</script>

</body>

</html>
```



Accessibility and data note

Provide the values in an HTML table or concise text summary as well as the chart. Do not rely on colour alone, and give the chart container an accessible name where practical. Google Charts loads JavaScript from Google; use fictional or non-personal summary data only.

Bar Charts from SQL Data

Bar charts are useful for comparing values between categories.

In this tutorial, the number of security events at each risk level will be displayed as a bar chart.

Step 1 - Create the SQL Query

```
SELECT

    risk_level,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY risk_level;
```

Example result:

risk_level	total
Low	15
Medium	8
High	3

Step 2 - Retrieve the Data with PHP

```
<?php
```

```
$sql = "

SELECT

    risk_level,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY risk_level

";

$result = $pdo->query($sql);

$data = [];

while($row = $result->fetch())
{

    $data[] = [

        $row['risk_level'],

        (int)$row['total']

    ];

}

?>
```

Step 3 - Create the Chart Container

```
<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>
```

Step 4 - Create the Bar Chart

```
<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =

        google.visualization.arrayToDataTable([

        ['Risk Level','Events'],

        ...<?php echo json_encode($data); ?>

    ]);


    var options = {

        title:'Security Events by Risk Level',

        legend:{position:'none'}

    };
```

```
    var chart =

        new google.visualization.BarChart(

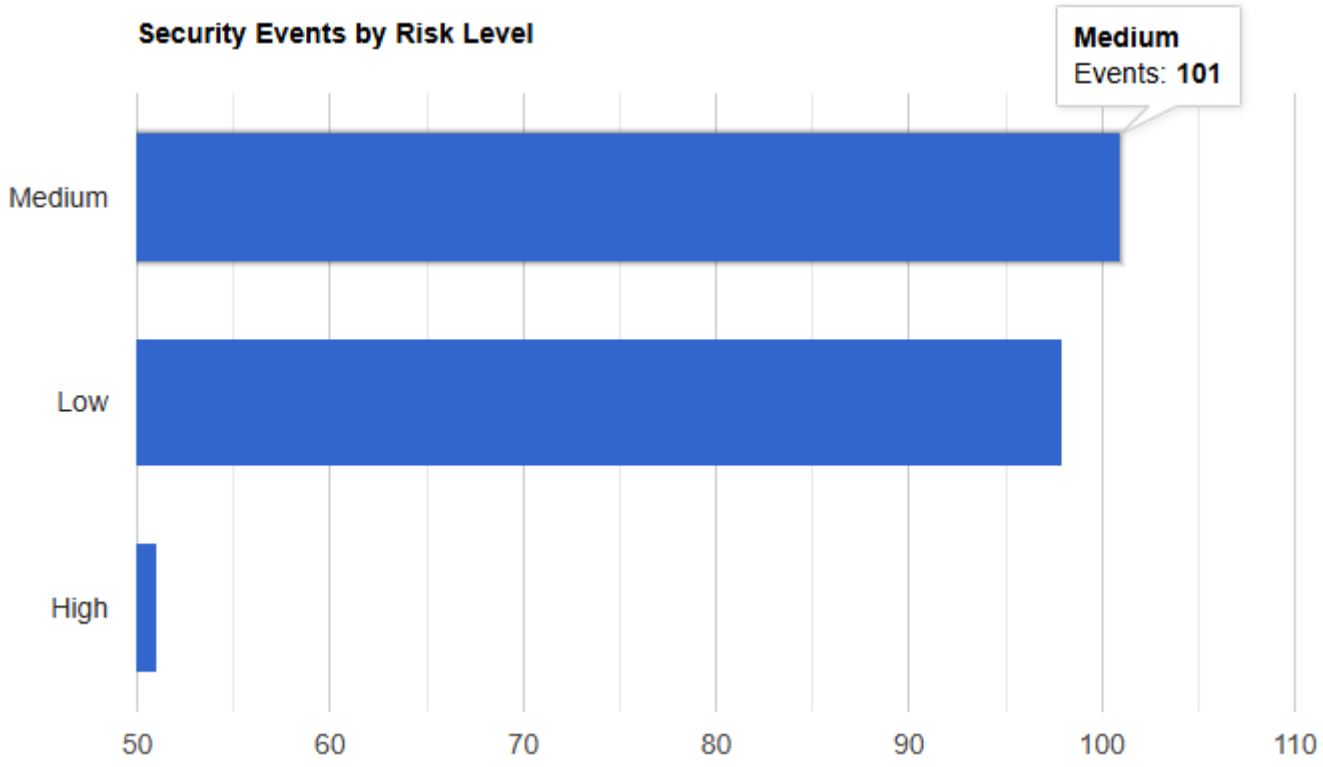
            document.getElementById('chart_div')

        );


    chart.draw(data, options);

}

</script>
```



Changing the Chart Type

Only one line of code needs to change.

Pie Chart:

```
new google.visualization.PieChart()
```

Bar Chart:

```
new google.visualization.BarChart()
```

Column Chart:

```
new google.visualization.ColumnChart()
```

Line Chart:

```
new google.visualization.LineChart()
```

Complete Example

```
<?php

$sql = "

SELECT

    risk_level,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY risk_level

";

$result = $pdo->query($sql);

$data = [];

while($row = $result->fetch())
```

```
{

    $data[] = [

        $row['risk_level'],

        (int)$row['total']

    ];

}

?>

<!DOCTYPE html>

<html>

<head>

<script src="https://www.gstatic.com/charts/loader.js"></script>

</head>

<body>

<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>


<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);
```

```
function drawChart()
{
    var data =
        google.visualization.arrayToDataTable([
        ['Risk Level','Events'],
        ...<?php echo json_encode($data); ?>
    ]);

    var options = {
        title:'Security Events by Risk Level',
        legend:{position:'none'}
    };

    var chart =
        new google.visualization.BarChart(
            document.getElementById('chart_div')
        );

    chart.draw(data, options);
}
</script>
```

```
</body>


</html>
```

Accessibility and data note

Provide the values in an HTML table or concise text summary as well as the chart. Do not rely on colour alone, and give the chart container an accessible name where practical. Google Charts loads JavaScript from Google; use fictional or non-personal summary data only.

Line Charts from SQL Data

Line charts are used to display trends and changes over time.

In this tutorial, the number of security events per day will be displayed using a line chart.

Step 1 - Create the SQL Query

The following query counts the number of events recorded each day.

```
SELECT

    DATE(event_timestamp) AS event_date,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY DATE(event_timestamp)

ORDER BY event_date;
```

Example result:

event_date	total
2026-06-01	12
2026-06-02	18
2026-06-03	9
2026-06-04	15

Step 2 - Retrieve the Data with PHP

```
<?php

$sql = "

SELECT

    DATE(event_timestamp) AS event_date,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY DATE(event_timestamp)

ORDER BY event_date

";

$result = $pdo->query($sql);

$data = [];

while($row = $result->fetch())
{

    $data[] = [

        $row['event_date'],

        (int)$row['total']

    ];

}
```

```
?>
```

Step 3 - Create the Chart Container

```
<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>
```

Step 4 - Create the Line Chart

```
<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =

        google.visualization.arrayToDataTable([

        ['Date','Events'],

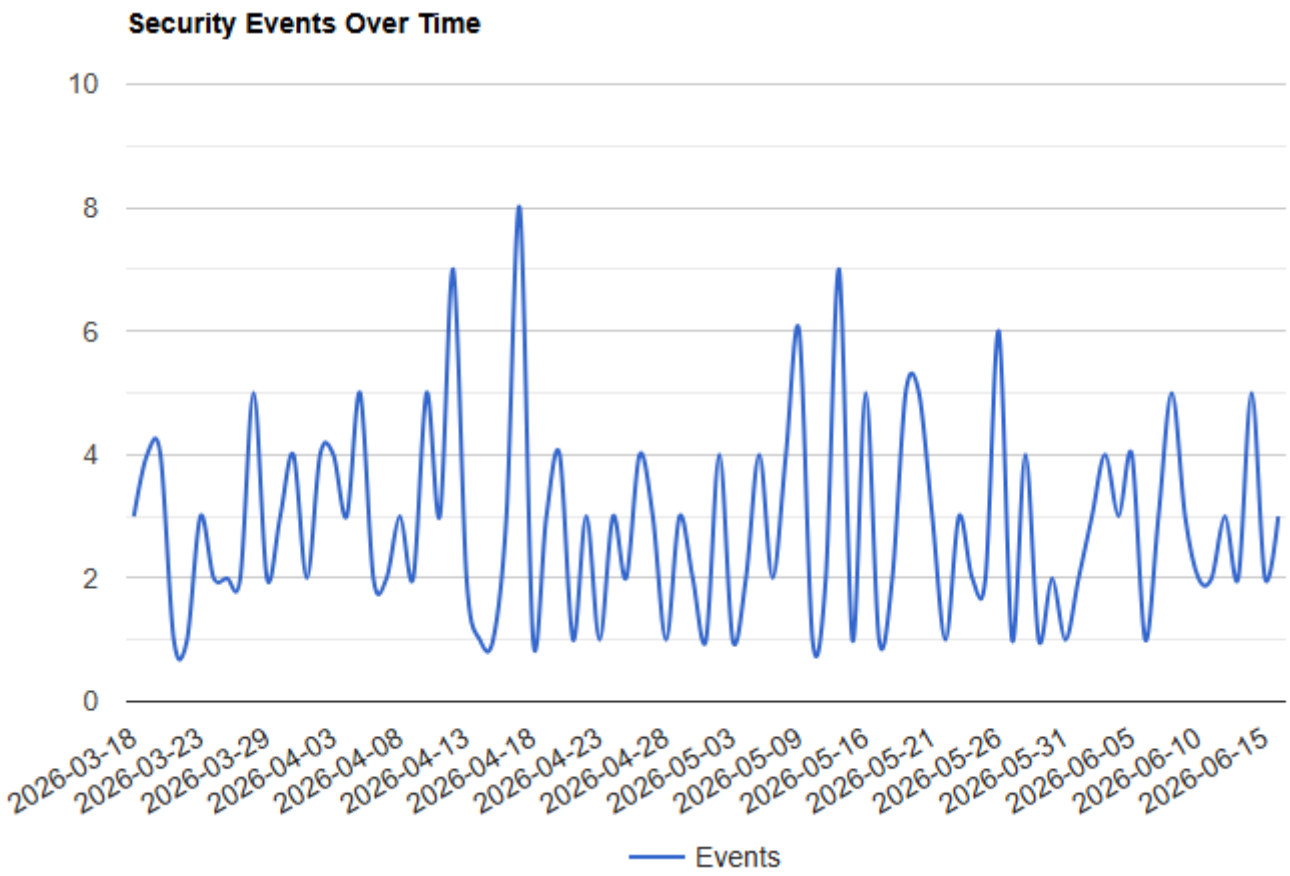
        ...<?php echo json_encode($data); ?>

    ]);
```

```
    var options = {
        title:'Security Events Over Time',
        curveType:'function',
        legend:{position:'bottom'}
    };

    var chart =
        new google.visualization.LineChart(
            document.getElementById('chart_div')
        );

    chart.draw(data, options);
}
</script>
```



Understanding the Chart

The horizontal axis displays:

```
Date
```

The vertical axis displays:

```
Number of Security Events
```

This makes it easier to identify:

- Busy periods
- Trends over time
- Sudden increases in activity
- Patterns in the data

Complete Example

```
<?php

$sql = "

SELECT

    DATE(event_timestamp) AS event_date,

    COUNT(*) AS total

FROM cyber_security_events

GROUP BY DATE(event_timestamp)

ORDER BY event_date

";

$result = $pdo->query($sql);

$data = [];

while($row = $result->fetch())
{

    $data[] = [

        $row['event_date'],

        (int)$row['total']

    ];

}

?>
```

```
<!DOCTYPE html>

<html>

<head>

<script src="https://www.gstatic.com/charts/loader.js"></script>

</head>

<body>

<div id="chart_div"
     style="width:100%; max-width:900px; min-height:320px;">
</div>


<script>

google.charts.load(
    'current',
    {'packages':['corechart']}
);

google.charts.setOnLoadCallback(drawChart);

function drawChart()
{

    var data =

        google.visualization.arrayToDataTable([

        ['Date','Events'],

        ...<?php echo json_encode($data); ?>

    ]);
```

```
    var options = {
        title:'Security Events Over Time',
        curveType:'function',
        legend:{position:'bottom'}
    };

    var chart =
        new google.visualization.LineChart(
            document.getElementById('chart_div')
        );

    chart.draw(data, options);
}
</script>
</body>
</html>
```

Accessibility and data note

Provide the values in an HTML table or concise text summary as well as the chart. Do not rely on colour alone, and give the chart container an accessible name where practical. Google Charts loads JavaScript from Google; use fictional or non-personal summary data only.