APIs

APIs	Description	Import statement	Source Code
file2db.file2db(host, user, password, filename, db_name, tb_name, file_type=”csv”, db_type=”mysql”, key=None)	Imports raw structured/semi-structured data (csv, json) into database (MySQL, NoSQL	from datamidware.pydm import file2db	Source Code
db2file.db2file(host, user, password, file_path, db_name, tb_name, file_type=”csv”, db_type=”mysql”)	Exports table data as csv/json format from the database	from datamidware.pydm import db2file	Source Code
create_mysql_db.create_mysql_db(host, user, password, db_name)	Creates a new MySQL database	from datamidware.pydm import create_mysql_db	Source Code
csv2mysql.csv2mysql(host, user, password, filename, db_name, tb_name)	Imports csv file into MySQL database. If database already exists, import data in the existing database, if not exists, create new database and import data. If table already exists, add data in the existing table, if not exists, create new table and import data.	from datamidware.pydm import csv2mysql	Source Code
json2mysql.json2mysql(host, user, password, filename, db_name, tb_name, key=None)	Loads json file to MySQL database table	from datamidware.pydm import json2mysql	Source Code
mysql2csv.mysql2csv(host, user, password, file_path, db_name, tb_name)	Exports csv file from MySQL database table	from datamidware.pydm import mysql2csv	Source Code
csv2viz.csv2viz(filename=None, kind=None, x=None, y=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, file_type="png", show=True, …)	Plots csv data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)	from datamidware.pydm import csv2viz	Source Code, bar
json2viz.json2viz(filename=None, key=None, kind=None, x=None, y=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, file_type="png", show=True, …)	Plots json data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)	from datamidware.pydm import json2viz	Source Code, bar
db2viz.db2viz(host, user, password, db_name, tb_name, db_type=None, kind=None, x=None, y=None, xcol_pos=None, ycol_pos=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, save2db={}, file_type="png", show=True, …)	Plots Database table data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)	from datamidware.pydm import db2viz	Source Code, bar
mysql2viz.mysql2viz(host, user, password, db_name, tb_name, kind=None, x=None, y=None, xcol_pos=None, ycol_pos=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, save2db={}, file_type="png", show=True, …)	Plots MySQL Database table data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)	from datamidware.pydm import mysql2viz	Source Code, bar
mysql2image.mysql2image(host, user, password, db_name=None, tb_name=None, sql_query=None, file_path=None, image_col=None, ext="png")	Retrieve image stored as a BLOB from MySQL table	from datamidware.pydm import mysql2image	Source Code
image2mysql.image2mysql(host, user, password, db_name=None, tb_name=None,	Insert Image as a BLOB data into MySQL Table from disk or directory	from datamidware.pydm import image2mysql	Source Code

dir_path=None, ext="png")			

mysql_query.MySQLDatabase(host, user, password, db_name)
run_query()
select()
drop_column()
rename_column()
	Database query	from datamidware.pydm import mysql_query
		or
		from datamidware.pydm.mysql_query import MySQLDatabase
			Source Code

1. **file2db.file2db**(host, user, password, filename, db_name, tb_name, file_type="csv", db_type="mysql", key=None): (Source Code)

   Imports raw structured/semi-structured data (csv, json) into database (MySQL, NoSQL).
   Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the server
filename:	Filename to send to database
db_name:	Name of the database where data will be stored
tb_name:	Name of the table where data will be stored
file_type:	File type (csv, json), str; (default csv)
db_type:	Database type (MySQL, NoSQL), str; (default MySQL)
key:	Specific json key name to create MySQL table; (default None)

2. **db2file.db2file**(host, user, password, file_path, db_name, tb_name, file_type="csv", db_type="mysql"): (Source Code)

   Exports table data as csv/json format from the database
   Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the server
file_path:	File path to export data from database
db_name:	Database name from where data is exported
tb_name:	Name of the table from where data will be exported
file_type:	File type (csv, json), str; (default csv)
db_type:	Database type (MySQL, NoSQL), str; (default MySQL)

3. **create_mysql_db.create_mysql_db**(host, user, password, db_name): (Source Code)

   Creates a new MySQL database
   Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the server
db_name:	Database name to be created

4. **csv2mysql.csv2mysql**(host, user, password, filename, db_name, tb_name): (Source Code)

   Imports csv file into MySQL database
   Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the server
filename:	filename to send to database
db_name:.	name of the database -- if database already exists, import data in the existing database, if not exists, create new database and import data
tb_name	name of the table -- if table already exists, add data in the existing table, if not exists, create new table and import data.

5. **json2mysql.json2mysql**(host, user, password, filename, db_name, tb_name, key=None): (Source Code)

Loads json file to MySQL database table
Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the server
filename:	filename to send to database
db_name:.	name of the database -- if database already exists, import data in the existing database, if not exists, create new database and import data
tb_name	name of the table -- if table already exists, add data in the existing table, if not exists, create new table and import data.
key:	Specific json key name to create MySQL table; (default None)

6. **mysql2csv.mysql2csv**(host, user, password, file_path, db_name, tb_name): (Source Code)

Exports csv file from MySQL database table
Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the mysql server
file_path:	File path to save csv file
db_name:	Name of the database from where data will be exported
tb_name:	Name of the table from where data will be exported

7. **csv2viz.csv2viz**(filename=None, kind=None, x=None, y=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, file_type="png", show=True, update_title={}, update_xaxes={}, update_yaxes={}, xtickangle=None, ytickangle=None, xtickformat=None, ytickformat=None, update_legend={}, update_font={}, hover_name=None, hover_data=None, barmode="relative", bargap=0.15, bargroupgap=0.1, color_discrete_sequence=None, fig_width=1200, fig_height=800, color_continuous_scale=None, uniformtext_minsize=8, uniformtext_mode="hide", marker={}, selector={}, trace_name=None, trace_text=None, texttemplate='%{text:.2s}', textangle=None, textposition="outside",textfont={}, bar_width=None, hoverinfo=None, hoverlabel=None, hovertemplate=None, hovertext=None): (Source Code)

Plots csv format data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)
Parameter:

filename:	Input filename (csv file)
kind:	Plot kind (bar, horizontal bar, hist …)
x:	x data; list or array-like data
y:	y data; list or array-like data
color:	Color of the fig (e.g., color of the bars)

title:	Title of the figure
labels:	x-label and y-label
set_col_color:	Show specific column data using specific color
update_trace_text:	Text trace; if True, shows trace; (default False)
sort_asc:	If True, Sort in ascending order; (default False)
sort_desc:	If True, Sort in descending order; (default False)
N_largest:	Shows top N values, N=1, 2,… (default None)
N_smallest:	Shows bottom N values, N=1, 2,.. (default None)
file_path:	File path to save figure
file_type:	File type to save figure (png, jpeg, pdf); (default .png)
show:	It True, show current figure; if False, does not show current figure; (default True)
Others: update_title, update_xaxes, update_yaxes, xtickangle, ytickangle, update_legend,…	Source code: csv2viz, bar
	See plotly **fig.update_layout()** on https://plotly.com/python/creating-and-updating-figures/#updating-figure-layouts https://plotly.com/python/figure-labels/ See **fig.update_traces()** on https://plotly.com/python/text-and-annotations/

	For e.g.,
	update_title={'y': 0.94, 'x': 0.5, 'xanchor': 'center', 'yanchor': 'top'},
	texttemplate='%{text:.2s}', textposition="inside",
	textfont=dict(family='Courier'),
	marker=dict(color=colors, opacity=0.8), plot_bgcolor="rgba(0, 0, 0, 0)",
	paper_bgcolor="rgba(0, 0, 0, 0)", update_xaxes=dict(ticksuffix="%")

8. **json2viz.json2viz**(filename=None, key=None, kind=None, x=None, y=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, file_type="png", show=True, update_title={}, update_xaxes={}, update_yaxes={}, xtickangle=None, ytickangle=None, xtickformat=None, ytickformat=None, update_legend={}, update_font={}, hover_name=None, hover_data=None, barmode="relative", bargap=0.15, bargroupgap=0.1, color_discrete_sequence=None, fig_width=1200, fig_height=800, color_continuous_scale=None, uniformtext_minsize=8, uniformtext_mode="hide", marker={}, selector={}, trace_name=None, trace_text=None, texttemplate='%{text:.2s}', textangle=None, textposition="outside",textfont={}, bar_width=None, hoverinfo=None, hoverlabel=None, hovertemplate=None, hovertext=None): (Source Code)

Plots json data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)
Parameter:

filename:	Input filename (csv file)
key:	Specific json key name to visualize (default None)
kind:	Plot kind (bar, horizontal bar, hist …)
x:	x data; list or array-like data
y:	y data; list or array-like data
color:	Color of the fig (e.g., color of the bars)
title:	Title of the figure
labels:	x-label and y-label
set_col_color:	Show specific column data using specific color
update_trace_text:	Text trace; if True, shows trace; (default False)
sort_asc:	If True, Sort in ascending order; (default False)
sort_desc:	If True, Sort in descending order; (default False)
N_largest:	Shows top N values, N=1, 2,… (default None)
N_smallest:	Shows bottom N values, N=1, 2,.. (default None)
file_path:	File path to save figure
file_type:	File type to save figure (png, jpeg, pdf); (default .png)
show:	It True, show current figure; if False, does not show current figure; (default True)

Others: update_title, update_xaxes, update_yaxes, xtickangle, ytickangle, update_legend,…	Source code: json2viz, bar
	See plotly **fig.update_layout()** on https://plotly.com/python/creating-and-updating-figures/#updating-figure-layouts https://plotly.com/python/figure-labels/ See **fig.update_traces()** on https://plotly.com/python/text-and-annotations/
	For e.g., update_title={'y': 0.94, 'x': 0.5, 'xanchor': 'center', 'yanchor': 'top'}, texttemplate='%{text:.2s}', textposition="inside", textfont=dict(family='Courier'), marker=dict(color=colors, opacity=0.8), plot_bgcolor="rgba(0, 0, 0, 0)", paper_bgcolor="rgba(0, 0, 0, 0)", update_xaxes=dict(ticksuffix="%")

9. **db2viz.db2viz**(host, user, password, db_name, tb_name, db_type=None, kind=None, x=None, y=None, xcol_pos=None, ycol_pos=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, save2db={}, file_type="png", show=True, update_title={}, update_xaxes={}, update_yaxes={}, xtickangle=None, ytickangle=None, xtickformat=None, ytickformat=None, update_legend={}, update_font={}, hover_name=None, hover_data=None, barmode="relative", bargap=0.15, bargroupgap=0.1, color_discrete_sequence=None, fig_width=1200, fig_height=800, color_continuous_scale=None, uniformtext_minsize=8, uniformtext_mode="hide", marker={}, selector={}, trace_name=None, trace_text=None, texttemplate='%{text:.2s}', textangle=None, textposition="outside",textfont={}, bar_width=None, hoverinfo=None, hoverlabel=None, hovertemplate=None, hovertext=None): (Source Code)

Plots Database table data – (Bar chart; kind="bar", Horizontal bar; kind="barh")
Parameter:

host:	Host name
user:	User name
password:	Root password to connect to the database
db_name:	Name of the database from where data will be visualized
tb_name:	Name of the table from where data will be visualized
db_type:	Database type (MySQL, NoSQL), str; (default MySQL)
kind:	Plot kind (bar, horizontal bar, hist …)
x:	x data; list or array-like data
y:	y data; list or array-like data
xcol_pos:	Str; Table column name that is used as x label (default None)
ycol_pos:	Str; Table column name that is used as y label (default None)
color:	Color of the fig (e.g., color of the bars)
title:	Title of the figure
labels:	x-label and y-label
set_col_color:	Show specific column data using specific color
update_trace_text:	Text trace; if True, shows trace; (default False)
sort_asc:	If True, Sort in ascending order; (default False)
sort_desc:	If True, Sort in descending order; (default False)
N_largest:	Shows top N values, N=1, 2,… (default None)
N_smallest:	Shows bottom N values, N=1, 2,.. (default None)
file_path:	File path to save figure
save2db:	Save figure in database; Dict; dict(host="host", user="user", password="password", db_name= "database_name", tb_name="table_name")
file_type:	File type to save figure (png, jpeg, pdf); (default .png)
show:	It True, show current figure; if False, does not show current figure; (default True)

Others: update_title, update_xaxes, update_yaxes, xtickangle, ytickangle, update_legend,…	Source code: json2viz, bar
	See plotly **fig.update_layout()** on https://plotly.com/python/creating-and-updating-figures/#updating-figure-layouts https://plotly.com/python/figure-labels/ See **fig.update_traces()** on https://plotly.com/python/text-and-annotations/

	For e.g.,
	update_title={'y': 0.94, 'x': 0.5, 'xanchor': 'center', 'yanchor': 'top'},
	texttemplate='%{text:.2s}', textposition="inside",
	textfont=dict(family='Courier'),
	marker=dict(color=colors, opacity=0.8), plot_bgcolor="rgba(0, 0, 0, 0)",
	paper_bgcolor="rgba(0, 0, 0, 0)", update_xaxes=dict(ticksuffix="%")

10. **mysql2viz.mysql2viz**(host, user, password, db_name, tb_name, kind=None, x=None, y=None, xcol_pos=None, ycol_pos=None, color=None, title=None, labels={}, set_col_color=None, update_trace_text=False, sort_asc=False, sort_desc=False, N_largest=None, N_smallest=None, file_path=None, save2db={}, file_type="png", show=True, update_title={}, update_xaxes={}, update_yaxes={}, xtickangle=None, ytickangle=None, xtickformat=None, ytickformat=None, update_legend={}, update_font={}, hover_name=None, hover_data=None, barmode="relative", bargap=0.15, bargroupgap=0.1, color_discrete_sequence=None, fig_width=1200, fig_height=800, color_continuous_scale=None, uniformtext_minsize=8, uniformtext_mode="hide", marker={}, selector={}, trace_name=None, trace_text=None, texttemplate='%{text:.2s}', textangle=None, textposition="outside",textfont={}, bar_width=None, hoverinfo=None, hoverlabel=None, hovertemplate=None, hovertext=None): (Source Code)

Plots MySQL Database table data – (Bar chart; kind=”bar”, Horizontal bar; kind=“barh”)
Parameter:

host:	Host name
user:	User name
password:	Root password to connect to the database
db_name:	Name of the database from where data will be visualized
tb_name:	Name of the table from where data will be visualized
db_type:	Database type (MySQL, NoSQL), str; (default MySQL)
kind:	Plot kind (bar, horizontal bar, hist …)
x:	x data; list or array-like data
y:	y data; list or array-like data
xcol_pos:	Str; Table column name that is used as x label (default None)
ycol_pos:	Str; Table column name that is used as y label (default None)
color:	Color of the fig (e.g., color of the bars)
title:	Title of the figure
labels:	x-label and y-label
set_col_color:	Show specific column data using specific color
update_trace_text:	Text trace; if True, shows trace; (default False)
sort_asc:	If True, Sort in ascending order; (default False)
sort_desc:	If True, Sort in descending order; (default False)
N_largest:	Shows top N values, N=1, 2,… (default None)
N_smallest:	Shows bottom N values, N=1, 2,.. (default None)
file_path:	File path to save figure
save2db:	Save figure in database; Dict; dict(host=”host”, user=“user”, password=“password”, db_name= “database_name”, tb_name=“table_name”)
file_type:	File type to save figure (png, jpeg, pdf); (default .png)
show:	It True, show current figure; if False, does not show current figure; (default True)

Others: update_title, update_xaxes, update_yaxes, xtickangle, ytickangle, update_legend,…	Source code: json2viz, bar
	See plotly **fig.update_layout()** on https://plotly.com/python/creating-and-updating-figures/#updating-figure-layouts https://plotly.com/python/figure-labels/ See **fig.update_traces()** on https://plotly.com/python/text-and-annotations/

	For e.g.,
	update_title={'y': 0.94, 'x': 0.5, 'xanchor': 'center', 'yanchor': 'top'},
	texttemplate='%{text:.2s}', textposition="inside",
	textfont=dict(family='Courier'),
	marker=dict(color=colors, opacity=0.8), plot_bgcolor="rgba(0, 0, 0, 0)",
	paper_bgcolor="rgba(0, 0, 0, 0)", update_xaxes=dict(ticksuffix="%")

11. **mysql2image.mysql2image**(host, user, password, db_name=None, tb_name=None,
sql_query=None, file_path=None, image_col=None, ext="png"):

Retrieve image stored as a BLOB from MySQL table
Parameter:

host:	Host name
user:	User name
password:	Root password to connect to the database
db_name:	Name of the database to retrieve image
tb_name:	Name of the table to retrieve image
sql_query:	Sql statement; default None (for e.g., select column using WHERE clause)
file_path:	File path to save retrieved image
image_col:	Image column name to retrieve image
ext:	File extension to save image; default “png”

12. **image2mysql.image2mysql**(host, user, password, db_name=None, tb_name=None, dir_path=None, ext="png"):

Insert images from disk or directory to MySQL table as a BLOB data
Parameter:

host:	Host name
user:	User name
password:	Root password to connect to the database
db_name:	Name of the database to insert image
tb_name:	Name of the table to insert image
dir_path:	Directory path of images
ext:	File extension of images to send to MySQL; if not specified, send all files; default None

13. **mysql_query.MySQLDatabase** (host, user, password, db_name): (Source Code)

Database query Class.
Parameters:

host:	Host name
user:	User name
password:	Root password to connect to the database
db_name:	Name of the database to perform SQL queries

Examples:

```
db = mysql_query.MySQLDatabase (host, user, password, db_name)
db.run_query(query)
db.select(tb_name, row_count=”one”)
db.rename_column(tb_name, “old_column_name”, "new_column_name", "column definition (col type)")
db.drop_column(tb_name, "column_name")
```

13.1. **mysql_query.MySQLDatabase.run_query** (query): (Source Code)

Execute SQL query. Returns number of rows affected after modification.

Parameters:

query:	SQL query to modify table

13.2. **mysql_query.MySQLDatabase.select**(tb_name, row_count="all"): (Source Code)

Select all rows (or select one row if row_count="one") from the table. Returns list of rows selected.

Execute SQL query:

```
“SELECT * FROM <table>”
```

Parameters:

tb_name:	The name of the table to modify
row_count:	"all" for all rows, "one" for one row; Default "all"

13.3. **mysql_query.MySQLDatabase.drop_column**(tb_name, col_name): (Source Code)

Drop a column in a table. Returns number of rows affected after modification

Execute SQL query:

```
“ALTER TABLE <table name>
 DROP COLUMN <column name>”
```

Parameters:

tb_name:	The name of the table to modify
col_name:	The name of the column to delete from the table

13.4. **mysql_query.MySQLDatabase.rename_column**(tb_name, old_name, new_name, col_def, col_pos=None): (Source Code)

Rename a column in a table. Returns number of rows affected after modification.

Execute SQL query:

```
“ALTER TABLE <table name>
  CHANGE COLUMN <old name> <new name>
  column_definition
  [ FIRST | AFTER column_name ]”
```

Parameters:

tb_name:	The name of the table to modify
old_name:	The column name to rename
new_name:	The new name for the column
col_def:	The data type and definition of the column (NULL or NOT NULL, etc). You must specify the column definition when renaming the column, even if it does not change
col_pos:	Optional. It tells MySQL where in the table to position the column, if you wish to change its position.