

DataMidWare

A Data-Oriented Middleware & Application Integration

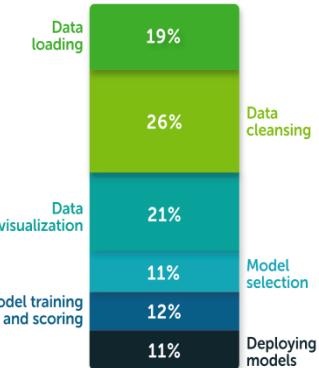
Jagriti Goswami

Outline

- Motivation
- Introduction of DataMidWare
- Architectural Overview of the Software
- Development Process Overview
- Code Architecture Representation using Component View
- Dependencies: Software, Tools, Libraries
- Environment Installation
- Code Implementation
- Testing
- Deployment & Execution
- Summary & Future Work

Motivation

- In this data-driven-age, every small and large organization is taking their business decisions based on the data insights.
- Transformation of raw data into usable cleaned data is crucial to take data-driven-decisions.
- Data cleaning and preparation takes valuable time away from real data science work – model building and deployment
 - Bulk time is spent on data preparation (According to [Anaconda survey-2020](#), on average **45%** of time is spent on data preparation, **21%** of time spent on visualization)
- Data preparation is challenging for multiple reasons :
 - Data from different sources come in various sizes and are different in nature.
 - Lack of integration between different data sources across the organization
 - Lack of proper integration between different technologies used for data science
 - Different platforms, software, libraries, and databases are used by developers
 - Managing dependencies and environments is another hurdle
 - Developing ad-hoc codes for specific use case
- Why DataMiddleWare Python Library?
 - Accelerate data preparation, analysis, and visualization tasks under a unified and single framework to overcome efficiency gap
 - Why Python for Data Analysis? According to [Anaconda's survey](#), **47%** of data scientists always use Python.



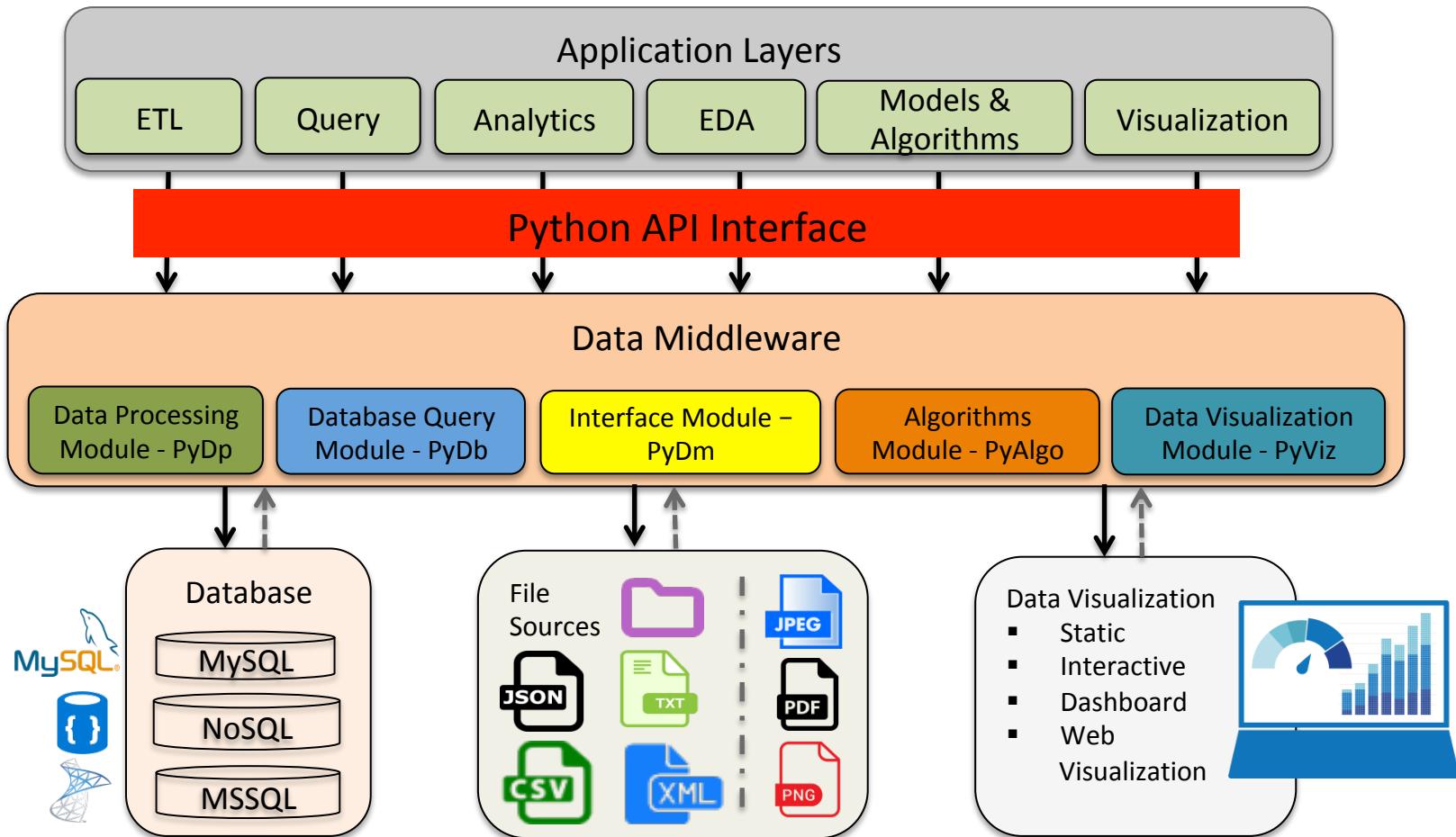
[[The State of Data Science 2020](#)]

DataMidWare is a data middleware which accelerates data preparation, analysis, and visualization tasks by integrating different technologies, software, and libraries using its APIs implemented in Python 3.

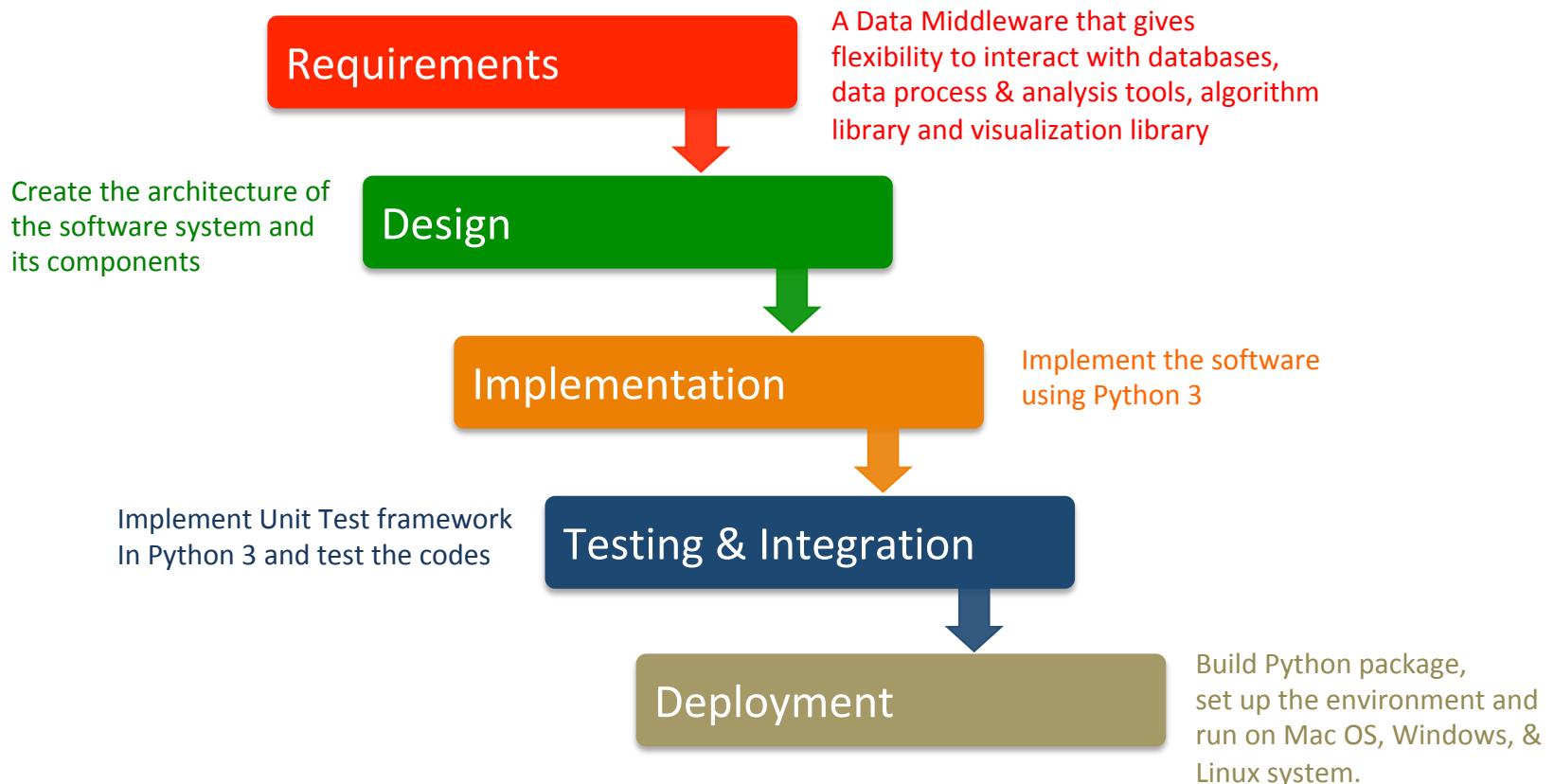
Introduction of DataMidWare

- **DataMidWare** - A data middleware implemented in Python 3.7.
 - [<https://github.com/JagritiG/data-middleware>]
- Provides direct interactions with database, data preparation & processing tool, algorithm library, and data visualization tool using its APIs.
- Imports and parses raw data (csv, json, txt, xml) from different sources and load to database (MySQL, NoSQL).
- Produces clean data from raw format and stores into database for analysis
- Performs SQL queries using its APIs
- Performs data analysis on the data stored in database
- Provides direct visualization of data stored in database
- Exports data in different format (csv, json) from database.
- *Accelerates data Preparation, Processing, and Visualization time*

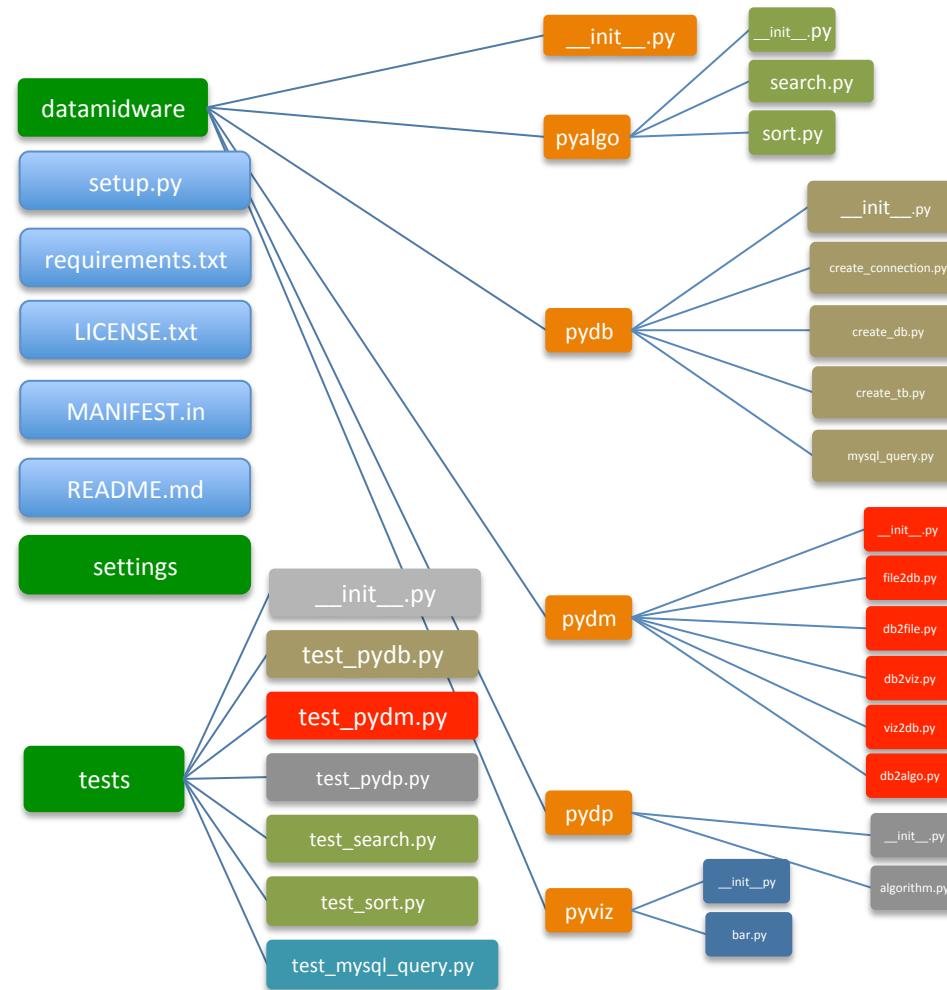
Architectural Overview



Development Process Overview



Code Architecture Representation using Component View



Dependencies: Software, Tools, Libraries

- IDE: PyCharm 18.3
- MySQL Community Server 8.0.21
- MySQL Workbench 6.3
- Programming language: Python (version 3.7)
- Python dependency packages

Package	Version
Numpy	1.19.2
Pandas	1.1.2
Plotly	4.11.0
PyMySQL	0.10.1
requests	2.24.0
SQLAlchemy	1.3.19
urllib3	1.25.10

requirements.txt

```
certifi==2020.6.20
chardet==3.0.4
idna==2.10
loguru==0.5.3
numpy==1.19.2
pandas==1.1.2
plotly==4.11.0
psutil==5.7.2
PyMySQL==0.10.1
python-dateutil==2.8.1
pytz==2020.1
requests==2.24.0
retrying==1.3.3
six==1.15.0
SQLAlchemy==1.3.19
urllib3==1.25.10
```

Environment Installation

1. PyCharm 18.3
2. MySQL Community Server 8.0.21
3. MySQL Workbench 6.3.10

Environment/IDE Name	Download and Installation Link	Video Link
PyCharm 18.3	https://www.jetbrains.com/help/pycharm/installation-guide.html	
Mac OS		https://www.youtube.com/watch?v=mDqxeCqVsOg
Windows		https://www.youtube.com/watch?v=EpjDOovzgrc
Linux		https://www.youtube.com/watch?v=hPhZOCXr_2w
MySQL Installation on Mac OS		https://www.youtube.com/watch?v=UcpHkYfWarM
MySQL Community Server 8.0.21	Download Link , How to Install?	
MySQL Workbench 6.3.10	Download Link , How to Install?	
MySQL Installation on Windows		https://www.youtube.com/watch?v=WuBcTJnluzo
MySQL Community Server 8.0	Download Link , How to Install?	
MySQL Community Server 8.0.21	Download Link	
MySQL Installation on Linux		
MySQL Community Server 8.0.21	Download Link , How to install?	https://www.youtube.com/watch?v=3qD6zy7thdE
MySQL Workbench 6.3.10	Download Link , How to install?	https://www.youtube.com/watch?v=9aYLfp5870

Code Implementation

API	Description	Parameters	Imports statement	Example Link
<u>file2db.file2db</u> (host, user, password, filename, db_name, tb_name, file_type="file_type", db_type="db_type")	Imports raw structured/semi-structured data (csv, json) into database (MySQL, NoSQL)	host: host name user: user name password: password filename: filename to send to database file_type: file type (csv, json), str db_type: database type (mysql, nosql), str tb_name: table name where data will be stored	from datamidware.pydm import file2db	Example , Doc
<u>db2file.db2file</u> (host, user, password, file_path, db_name, tb_name, file_type="file_type", db_type="db_type")	Exports table data as csv/json format from the database	host: host name user: user name password: password file_path: file path to export data from database file_type: file type (csv, json) db_type: database type (mysql, nosql) db_name: database from where data is exported tb_name: table from where data will be exported	from datamidware.pydm import db2file	Example , Doc
<u>create_mysql_db.create_mysql_db</u> (host, user, password, db_name)	Creates a new MySQL database	host: host name user: user name password: password db_name: database name to be created	from datamidware.pydm import create_mysql_db	
<u>mysql_query.MySQLData</u> <u>base.select</u> (tb_name, row_count="all")	Execute SQL query: SELECT * FROM table. Selecting all(or one if row_count="one") rows from the table.	query: SQL query to select rows: SELECT * FROM <table> row_count: "all" or "one" row. default "all". return: list of rows selected.	from datamidware.pydm import mysql_query	Example , Doc

Code Implementation

API	Description	Parameters	Imports statement	Example Link
<code>csv2mysql.csv2mysql</code> (host , user, password, filename, db_name, tb_name)	Imports csv file into mysql database	host : host name user : user name password : password 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.	from datamidware.pydm import csv2mysql	Doc
<code>mysql2csv.mysql2csv</code> (host , user, password, file_path, db_name, tb_name)	Exports csv file from mysql database table	host : host name user : user name password : password 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	from datamidware.pydm import mysql2csv	Doc
<code>db2viz.db2viz</code> (host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)	Visualize the Database table data	host : host name, user : user name password : password, db_name : database name tb_name : table name, kind : plot kind (bar, horizontal bar, hist,..), file_path : file path to save figure x : x data; list or array-like, y : y data; list or array-like title : title of the figure, label : x-label, y-label	from datamidware.pydm import db2viz	Example , Doc

Testing

Modules	Description	Execution command	Remark
test_pydm.py	Test module for pydm sub-package	python3 -m unittest tests/test_pydm.py	Pass
test_mysql_query.py	Test module for mysql_query.py module	python3 -m unittest tests/test_mysql_query.py	Pass
test_search.py	Test module for search.py module	python3 -m unittest tests/test_search.py	Pass
test_sort.py	Test module for sort.py module	python3 -m unittest tests/test_sort.py	Pass

- To test all the modules, run all the above unittest command from the top-level directory
- To successfully run the test, config.ini file is required
- To write config.ini file go to /settings
- update write_config.py with database connection credentials, for e.g.,
(for MySQL database connection)

```
config_object["MYSQL"] = {  
    "host": "hostname",  
    "user": "username",  
    "password": "password"  
}
```

Deployment & Execution

- Using the terminal or an Anaconda Prompt, create a new environment – `conda create -n env_name python=3.7`
- Activate new environment using - `conda activate env_name`
- Install package using pip – `pip install -i https://test.pypi.org/simple/ datamidware==2020.11`
- Install requirements.txt using pip – `pip install -r requirements.txt`

```
(env_dataware) S: [REDACTED] $ pip install -i https://test.pypi.org/simple/ datamidware==0.0.11
Looking in indexes: https://test.pypi.org/simple/
Collecting datamidware==0.0.11
  Downloading https://test-files.pythonhosted.org/packages/34/1d/103524d61472838c103e1411f07c1263ebf509e7a9f7242b2d72092a73be/datamidware-0.0.11-py3-none-any.whl (96 kB)
! [REDACTED] | 96 kB 685 kB/s
Installing collected packages: datamidware
Successfully installed datamidware-0.0.11
(env_dataware) S: [REDACTED] $ pip install -r requirements.txt
ERROR: Invalid requirement: '-r'
(env_dataware) S: [REDACTED] $ pip install -r requirements.txt
Requirement already satisfied: certifi==2020.6.20 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (2020.6.20)
Requirement already satisfied: chardet==3.0.4 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 2)) (3.0.4)
Requirement already satisfied: idna==2.10 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 3)) (2.10)
Requirement already satisfied: loguru==0.5.3 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (0.5.3)
Requirement already satisfied: numpy==1.19.2 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 5)) (1.19.2)
Requirement already satisfied: pandas==1.1.2 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (1.1.2)
Requirement already satisfied: plotly==4.11.0 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (4.11.0)
Requirement already satisfied: psutil==5.7.2 in ./anaconda3/envs/env_dataware/lib/python3.7/site-packages (from -r requirements.txt (line 8)) (5.7.2)
```

Example:`file2db`.`file2db(host, user, password, filename, db_name, tb_name, file_type="file_type", db_type="db_type")`

- Execute `file2db`.`file2db(host, user, password, filename, db_name, tb_name, file_type="file_type", db_type="db_type")` to load csv file into MySQL DB
- Example data: titanic.csv

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view has 'titanic' selected and highlighted with a red box. The central workspace contains a terminal window showing a Python session. The command `>>> file2db.file2db(host, user, password, csv_file, db_name, "csv", "mysql")` is highlighted with a red box. Below the terminal, the 'Result Grid' shows the data from the 'titanic' table, and the 'Action Output' pane at the bottom lists the SQL commands executed: `SELECT * FROM titanic.titanic LIMIT 0, 50`, `DROP DATABASE 'titanic'`, and `SELECT * FROM titanic.titanic LIMIT 0, 50`.

```
(env_dataware) Santanu-MacBook-Pro:~ santanu$ python
Python 3.7.9 (default, Aug 31 2020, 07:22:35)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import datamiddleware
>>> from datamiddleware.pydm import file2db
>>> host = "localhost"
>>> password = "MySql@2020"
>>> user = "root"
>>> csv_file = "/Users/santanu/Downloads/Programming/Data Analysis/data_integration_middleware/tests/test_data/titanic_short.csv"
>>> db_name = "titanic"
>>> tb_name = "titanic"
>>> file2db.file2db(host, user, password, csv_file, db_name, "csv", "mysql")
Connected to DB: localhost
Successfully created pydb titanic
{'Database': 'ca_pop'}
{'Database': 'consumer'}
{'Database': 'country'}
{'Database': 'housing'}
{'Database': 'information_schema'}
{'Database': 'mysql'}
{'Database': 'performance_schema'}
{'Database': 'product'}
{'Database': 'sys'}
{'Database': 'testdb'}
{'Database': 'testdb2'}
{'Database': 'titanic'}
{'Database': 'tmdb'}
```

Surviv...	Pclass	Name	Sex	Age	Siblings_Spouses_A... _A	Parents_Children_A... _A	Fare
0	3	Mr. Owen Harris Braund	male	22	1	0	72.25
1	1	Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	71.2833
1	3	Mrs. Laina Hekkinen	female	26	0	0	7.925
1	1	Mr. Jacques Heath (Ily May Peel) Futrelle	female	35	1	0	53.1
0	3	Mr. William Henry Allen	male	35	0	0	8.05
0	3	Mr. James Moran	male	27	0	0	8.4683
0	1	Mr. Timothy J McCarthy	male	54	0	0	51.8625
0	3	Master. Gosta Leonard Palsson	male	2	3	1	21.075
1	3	Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johansson	female	27	0	2	11.1333

Action Output

Time	Action	Response	Duration / F
14:54:37	SELECT * FROM titanic.titanic LIMIT 0, 50	50 row(s) returned	0.00049 sec
14:54:47	DROP DATABASE 'titanic'	1 row(s) affected	0.033 sec
15:12:12	SELECT * FROM titanic.titanic LIMIT 0, 50	Error Code: 1049. Unknown database 'titanic'	0.0026 sec
15:18:40	SELECT * FROM titanic.titanic LIMIT 0, 50	9 row(s) returned	0.00037 sec

[Back](#)

Example:**db2file.db2file(host, user, password, file_path, db_name, tb_name, file_type="file_type", db_type="db_type")**

- Execute **db2file.db2file(host, user, password, file_path, db_name, tb_name, file_type="file_type", db_type="db_type")** to export csv file from MySQL
- Example data: MySQL database name : titanic, table name = titanic

```
(env_dataaware) Santanu-MacBook-Pro: santanusrarma$ python
Python 3.7.9 (default, Aug 31 2020, 07:22:35)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> import datamidware
[>>> from datamidware.pydm import db2file
[>>> host = "localhost"
[>>> user = "root"
[>>> password = "*****"
[>>> output_filepath = "/Users/santanusrarma/Dropbox/Jagriti/Programming/Data Analysis/data_integration_middleware/tests/test_result/"
[>>> db_name = "titanic"
[>>> tb_name = "titanic"
[>>> db2file.db2file(host, user, password, output_filepath, db_name, tb_name, "csv", "mysql")]
2020-11-06 15:22:37,038 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'sql_mode'
2020-11-06 15:22:37,039 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:22:37,042 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,076 INFO sqlalchemy.engine.base.Engine SHOW CREATE TABLE `titanic`
2020-11-06 15:22:37,042 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:22:37,044 INFO sqlalchemy.engine.base.Engine ['Survived', 'Pclass', 'Name', 'Sex', 'Age', 'Siblings_Spouses_Aboard', 'Parents_Children_Aboard', 'Fare']
2020-11-06 15:22:37,044 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,080 INFO sqlalchemy.engine.base.Engine DESCRIBE `SELECT * FROM titanic`
2020-11-06 15:22:37,044 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,080 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:22:37,045 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,082 INFO sqlalchemy.engine.base.Engine ROLLBACK
2020-11-06 15:22:37,045 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,082 INFO sqlalchemy.engine.base.Engine SELECT * FROM titanic
Collation` = 'utf8mb4_bin' 2020-11-06 15:22:37,082 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:22:37,045 INFO sqlalchemy.engine.base.Engine 2020-11-06 15:22:37,082 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine   Survived  Pclass ... Parents_Children_Aboard    Fare
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine 0      0     3 ...          0    7.2500
anon_1
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine 1      1     1 ...          0   71.2833
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine 2      1     3 ...          0   7.9250
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine 3      1     1 ...          0   53.1000
AS anon_1
2020-11-06 15:22:37,049 INFO sqlalchemy.engine.base.Engine 4      0     3 ...          0   8.0500
[5 rows x 8 columns]
File, /Users/santanusrarma/Dropbox/Jagriti/Programming/Data Analysis/data_integration_middleware/tests/test_result/titanic.csv, has been created successfully
>>>
```

Example: *mysql_query.MySQLDatabase.select(tb_name, row_count="all")*

- Execute *mysql_query.MySQLDatabase.select(tb_name, row_count="all")* to select all rows from MySQL table
- Example data: MySQL database name : titanic, table name = titanic

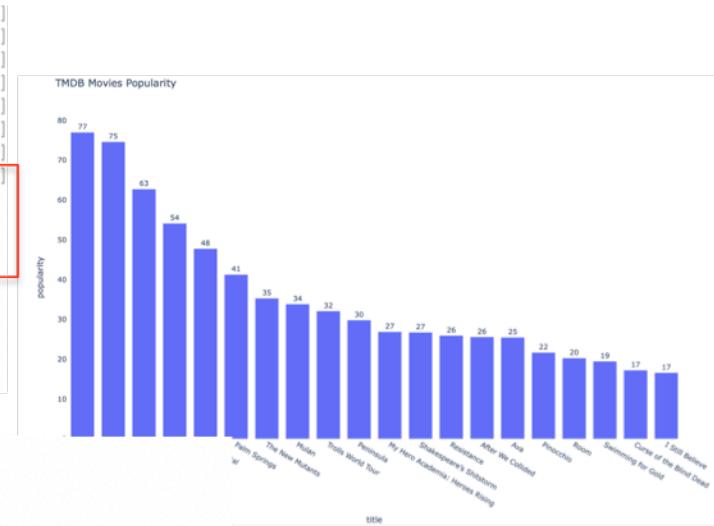
```
[env_dataware] ~ % [root@localhost ~] $ python
Python 3.7.9 (default, Aug 31 2020, 07:22:35)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> import datamidware
[>>> from datamidware.pydm import mysql_query
[>>> host = "localhost"
[>>> user = "root"
[>>> password = "*****"
[>>> db_name = "titanic"
[>>> tb_name = "titanic"
[>>> db = mysql_query.MySQLDatabase(host, user, password, db_name)
[>>> rows = db.select(tb_name, row_count="all")
[2020-11-06 15:33:27.254 | INFO    | datamidware.pydm.mysql_query:open_connection:44 - Connection opened successfully.
[2020-11-06 15:33:27.256 | INFO    | datamidware.pydm.mysql_query:select:78 - Database connection closed.
[>>> rows
[{'Survived': 0, 'Pclass': 3, 'Name': 'Mr. Owen Harris Braund', 'Sex': 'male', 'Age': 22, 'Siblings_Spouses_Aboard': 1, 'Parents_Children_Aboard': 0, 'Fare': 7.25}, {'Survived': 1, 'Pclass': 1, 'Name': 'Mrs. John Bradley (Florence Briggs Thayer) Cumings', 'Sex': 'female', 'Age': 38, 'Siblings_Spouses_Aboard': 1, 'Parents_Children_Aboard': 0, 'Fare': 71.2833}, {'Survived': 1, 'Pclass': 3, 'Name': 'Miss. Laina Heikkinen', 'Sex': 'female', 'Age': 26, 'Siblings_Spouses_Aboard': 0, 'Parents_Children_Aboard': 0, 'Fare': 7.925}, {'Survived': 1, 'Pclass': 1, 'Name': 'Mrs. Jacques Heath (Lily May Peel) Futrelle', 'Sex': 'female', 'Age': 35, 'Siblings_Spouses_Aboard': 1, 'Parents_Children_Aboard': 0, 'Fare': 53.1}, {'Survived': 0, 'Pclass': 3, 'Name': 'Mr. William Henry Allen', 'Sex': 'male', 'Age': 35, 'Siblings_Spouses_Aboard': 0, 'Parents_Children_Aboard': 0, 'Fare': 8.05}, {'Survived': 0, 'Pclass': 3, 'Name': 'Mr. James Moran', 'Sex': 'male', 'Age': 27, 'Siblings_Spouses_Aboard': 0, 'Parents_Children_Aboard': 0, 'Fare': 8.4583}, {'Survived': 0, 'Pclass': 1, 'Name': 'Mr. Timothy J McCarthy', 'Sex': 'male', 'Age': 54, 'Siblings_Spouses_Aboard': 0, 'Parents_Children_Aboard': 0, 'Fare': 51.8625}, {'Survived': 0, 'Pclass': 3, 'Name': 'Master. Gosta Leonard Palsson', 'Sex': 'male', 'Age': 2, 'Siblings_Spouses_Aboard': 3, 'Parents_Children_Aboard': 1, 'Fare': 21.075}, {'Survived': 1, 'Pclass': 3, 'Name': 'Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson', 'Sex': 'female', 'Age': 27, 'Siblings_Spouses_Aboard': 0, 'Parents_Children_Aboard': 2, 'Fare': 11.1333}]
>>> ]
```

Example: **db2viz.db2viz(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)**

- Execute **db2viz.db2viz(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)** to plot bar chart to visualize movies by popularity from tmdb_results table from MySQL database tmdb.
- Example data: database name - tmdb, table name - tmdb_results

```
>>> import datamidware
>>> from datamidware.pydm import db2viz
>>> host = "localhost"
>>> user = "root"
>>> password = "*****"
>>> db_name = "tmdb"
>>> tb_name = "tmdb_results"
>>> db2viz.db2viz(host, user, password, db_name, tb_name, db_type="mysql", kind="bar", y="popularity", x="title", title="TMDB Movies Popularity", labels={"y": "popularity", "x": "title"}, update_trace_text=True, file_path="/Users/*****/Desktop/*****/Programming/Data Analysis/data_integration_middleware/tests/test_result/", save2db=dict(host="localhost", user="root", password="*****", db_type="mysql", db_name="tmdb", tb_name="image"), show=True)
2020-11-06 15:53:32,291 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'sql_mode'
2020-11-06 15:53:32,291 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:53:32,293 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'lower_case_table_names'
2020-11-06 15:53:32,293 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 15:53:32,294 INFO sqlalchemy.engine.base.Engine SELECT DATABASE()
```

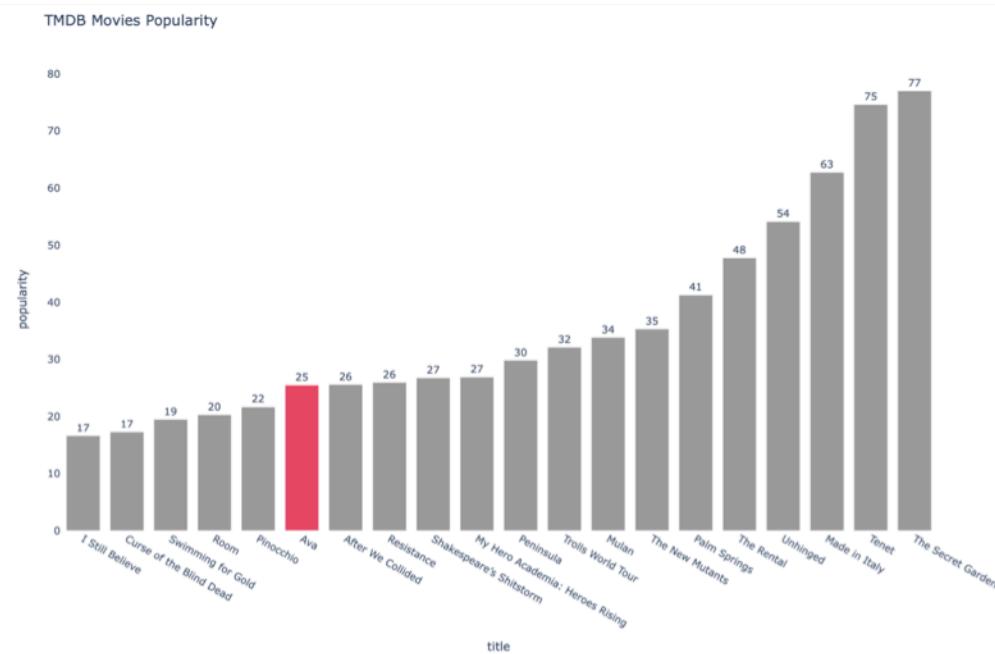
```
_result/tmdb_movies_popularity.png, has been created successfully
localhost root ***** tmdb image
('image',)
('tmdb_results',)
Inserting BLOB into table
Image inserted successfully as a BLOB into image table 1
MySQL connection is closed
```



Example: ***db2viz.db2viz(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)***

- Execute ***db2viz.db2viz(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)*** to plot bar chart for movies by popularity from tmdb_results table in ascending order and highlighting movie title - "Ava"

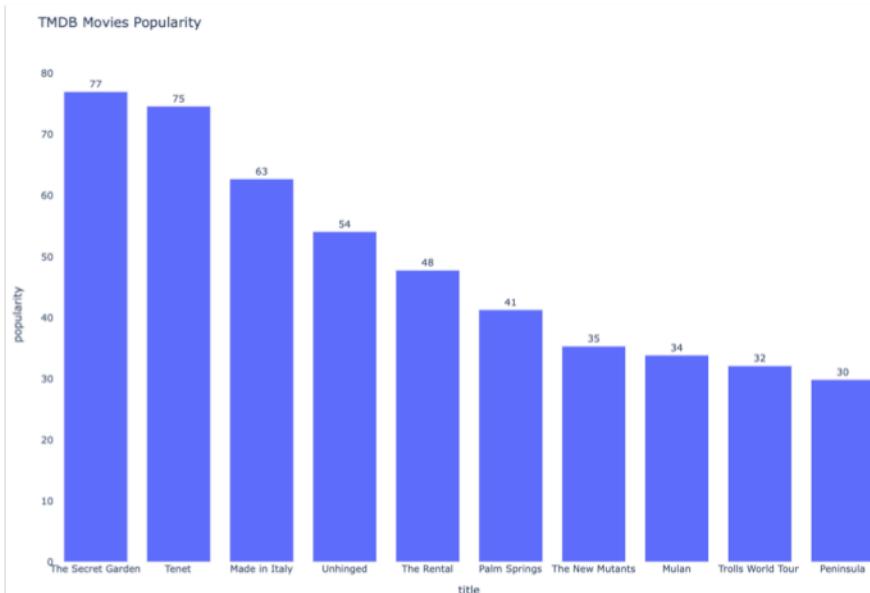
```
>>> db2viz.db2viz(host, user, password, db_name, tb_name, db_type="mysql", kind="bar", y="popularity", x="title", title="TMDB Movies Popularity", labels={"y": "popularity", "x": "title"}, update_trace_text=True, set_col_color="Ava", sort_asc=True, file_path="/Users/.../Desktop/PyCharm/Programming/Data Analysis/data_integration_middleware/tests/test_result/", save2db=dict(host="localhost", user="root", password="..._..._..._...", db_type="mysql", db_name="tmdb", tb_name="image"), show=True)
2020-11-06 16:02:53,064 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'sql_mode'
2020-11-06 16:02:53,064 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 16:02:53,065 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'lower_case_table_names'
```



Example: ***db2viz.db2viz***(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...)

- Execute ***db2viz.db2viz***(host, user, password, db_name, tb_name, kind=None, x=None, y=None, ...) to plot bar chart for 10 highest movies by popularity from tmdb_results table of tmdb database.

```
>>> db2viz.db2viz(host, user, password, db_name, tb_name, db_type="mysql", kind="bar", y="popularity", x="title", title="TMDB Movies Popularity", labels={"y": "popularity", "x": "title"}, update_trace_text=True, N_largest=10, file_path="/Users/[REDACTED]/Desktop/PyCharm/Programming/Data Analysis/data_integration_middleware/tests/test_result/", save2db=dict(host="localhost", user="root", password="[REDACTED]", db_type="mysql", db_name="tmdb", tb_name="image"), show=True)
2020-11-06 16:06:21,103 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'sql_mode'
2020-11-06 16:06:21,104 INFO sqlalchemy.engine.base.Engine {}
2020-11-06 16:06:21,105 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'lower_case_table_names'
2020-11-06 16:06:21,105 INFO sqlalchemy.engine.base.Engine {}
```



Summary & Future Work

- Summary
 - A data middleware that gives flexibility to directly interact with database, data preparation & processing tool, algorithm library, and data visualization tool using its APIs; *accelerates data preparation, processing, & Visualization time*
 - Imports and parses raw data (csv, json, txt, xml) and load to database (MySQL, NoSQL).
 - Produces clean data from raw format and stores into database for analysis
 - Performs SQL queries
 - Provides data analysis and direct visualization of data stored in database
 - Exports data in different format (csv, json) from database.
- Future Work
 - Extend all the libraries – (data processing, algorithms, and data visualization library)
 - Integrate Exploratory and Inferential data analysis library
 - Integrate Machine Learning algorithms library
 - Integrate Web Visualization
 - Integrate with other Databases ([MSSQL](#))
 - Integrate with Splunk

MySQL Installation on Mac OS X

1. MySQL Community Server 8.0.21 Download (MacOS Mojave version 10.14.4):

1. Open the below link on <https://www.mysql.com/downloads/> and Click on – MySQL Community (GPL) Downloads
2. Or directly open the below link to download MySQL community version <https://dev.mysql.com/downloads/>
3. Using step-1 or step-2 navigate to MySQL Community Downloads page (**fig-1**) and Open MySQL Community Server (**fig-2**)
4. Select operating system from dropdown menu - **macOS**
5. To download latest version (8.0.22) - **macOS 10.15 (x86, 64-bit), DMG Archive** (mysql-8.0.22-macos10.15-x86_64.dmg), click on <**download**>
6. To download version 8.0.21 click on **Archives**
7. To download **macOS 10.15 (x86, 64-bit), DMG Archive** (mysql-8.0.21-macos10.15-x86_64.dmg), click on download
8. Login Now or Sign Up window (**fig-3**) will appear – (No need to click on Login or Sign Up)
9. Click on -> **No Thanks, just start my download.**

MySQL Installation on Mac OS X

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Installer for Windows
- MySQL for Visual Studio
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/.NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives

Fig-1

MySQL Community Downloads

MySQL Community Server 8.0.22

Select Operating System: macOS

Looking for previous GA versions?

macOS 10.15 (x86, 64-bit), DMG Archive	8.0.22	401.5M	Download
(mysql-8.0.22-macos10.15-x86_64.dmg)	MDS: 6fa3a385100e474dc2a6a4e848b3f9284 Signature		
macOS 10.15 (x86, 64-bit), Compressed TAR Archive	8.0.22	160.6M	Download
(mysql-8.0.22-macos10.15-x86_64.tar.gz)	MDS: e4c8da935e75e3a943609c74e28e737 Signature		

Fig-2

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)
using my Oracle Web account

[Sign Up »](#)
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

Fig-3

[Back](#)

MySQL Installation on Mac OS X

2. Installation and Setup:

1. Double click on .dmg file (mysql-8.0.21-macos10.15-x86_64.pkg file will appear)(**fig-4**) (if latest version is downloaded, then file name will be different)
2. To start installing double click on -> **mysql-8.0.21-macos10.15-x86_64.pkg**
3. Install window will appear (**fig-5**) -> click on <**Continue**>
4. Agree the License agreement and click <**Continue**>
5. Change destination if you want or keep it as default and click on <**Continue**>
6. In the installation window, give Mac OS password and click on <**Install Software**>
7. Now Installation starts
8. After installation is done, a **temporary password** will be generated (**fig-6**)
9. Copy and save this password. It is required to build database connection for the first time.
10. Close all the windows and check whether MySQL is installed correctly on Mac OS or not
11. To check correctly installed or not, click on **Apple icon** on left upper corner -> go to **System Preferences**
12. Check for **MySQL Logo** (**fig-7**)- if present, that means MySQL is installed correctly.
13. Click on **MySQL Logo** -> **MySQL Server Status** window appears (**fig-8**) -> click on <**Start MySQL Server**>
14. MySQL Server is now running

MySQL Installation on Mac OS X

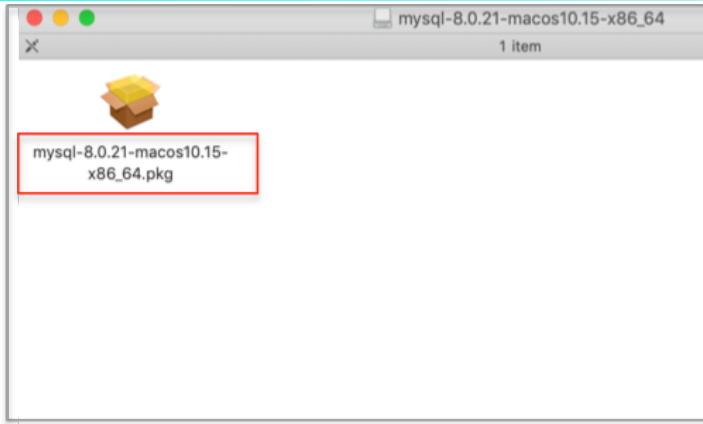


Fig-4

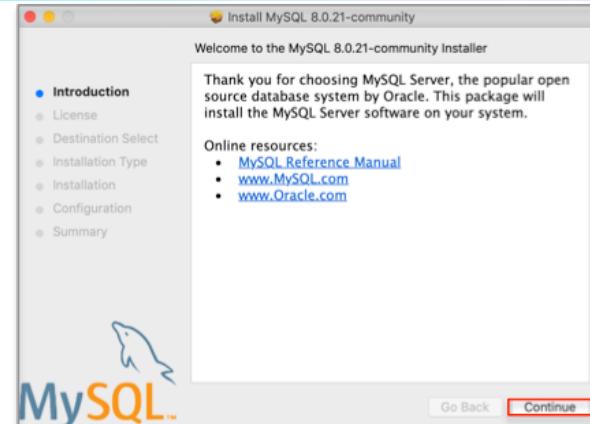


Fig-5

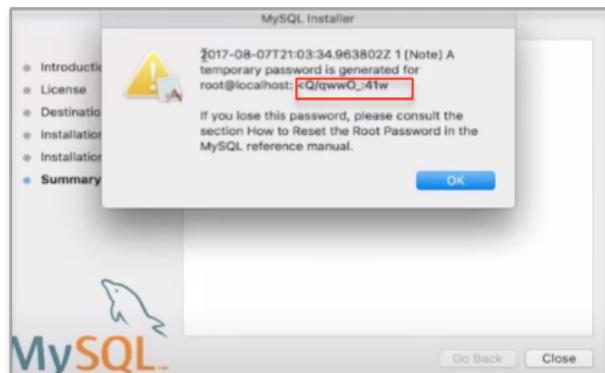


Fig-6



Fig-7



Fig-8

MySQL Installation on Mac OS X

3. Connect to the MySQL Server using MySQL Client

1. To connect to the server using Terminal, follow the below steps:

1. Set MySQL location in .bash_profile
2. Open Terminal, make sure you are in home directory, to confirm this type command – **pwd**. Then type command – **ls -all**
3. Open **.bash_profile** using command – **open -t .bash_profile**
4. If **.bash_profile** is not present, create one using command – **touch .bash_profile** and open it
5. Copy **export PATH=\${PATH}:/usr/local/mysql/bin/** to **.bash_profile**. Save this and close the file
6. Connect to the MySQL server using terminal – open terminal and type command **mysql -u root -p** (connects to the MySQL server using root user and **-p** flag, the saved temporary password)
7. Give the **temporary password** saved during installation and press **<Enter>** → MySQL terminal will open (**fig-9**)
8. First change temporary password using command **ALTER USER 'root'@'localhost' IDENTIFIED BY 'newpassword'** (**fig-10**)
9. Open a new terminal and again type command **mysql -u root -p** and give new password to open MySQL terminal
10. Now type SQL statement **<show databases;>** to see default databases (**fig-11**). See MySQL tutorial
<https://www.w3schools.com/sql/> for more

MySQL Installation on Mac OS X

```
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 272  
Server version: 5.7.19  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> 
```

Fig-9

```
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 272  
Server version: 5.7.19  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass';
```

Fig-10

```
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.14 sec)  
  
mysql> 
```

Fig-11

MySQL Installation on Mac OS X

2. To connect to server using MySQL Workbench, follow the below installation and setup steps

1. Go to MySQL Community Downloads page <https://dev.mysql.com/downloads/> and click on MySQL Workbench (fig-12)
2. Download MySQL Workbench in the similar way as MySQL Community Server
3. Open MySQL Workbench dmg file (fig-13) and drag & drop MySQL Workbench into Application folder
4. Open MySQL Workbench from Launch Pad or Application folder – (fig-14)
5. By default MySQL server starts on port 3306, click on this and give newpassword reset from terminal and check <Save password in keychain>box(fig-14) → click ok to open MySQL interface (fig-15)

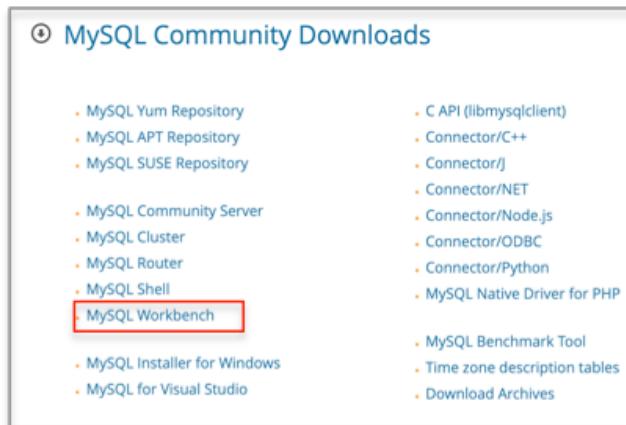


Fig-12

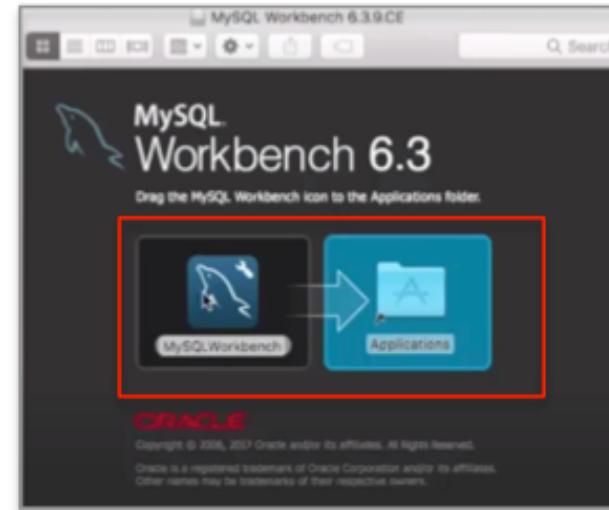


Fig-13

[Back](#)

MySQL Installation on Mac OS X

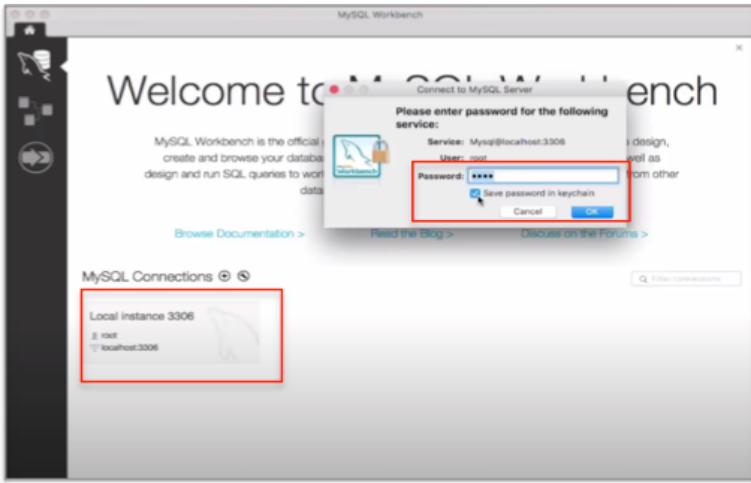


Fig-14

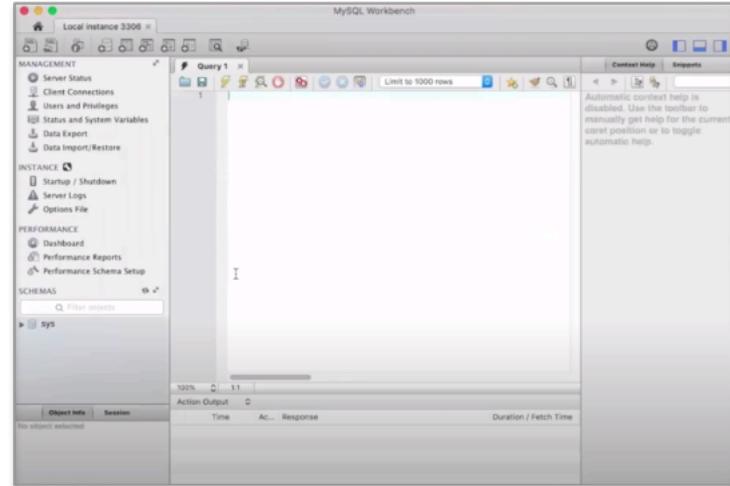


Fig-15

MySQL Installation on Windows

1. MySQL Community Server Download:

1. Open the given link to download MySQL community version <https://dev.mysql.com/downloads/>
2. Click on **MySQL Community Server**
3. Select operating system from dropdown menu – **Microsoft Windows (fig-16)**
4. Click on <**Go to Download Page**> tab next to **Windows (x86, 32 & 64-bit), MySQL Installer MSI (fig-16)**
5. Download (**mysql-installer-community-8.0.22.0.msi**) (fig-17)
6. To install **version 8.0.21**, click on **Archives** → download (**mysql-installer-community-8.0.21.0.msi**)
7. Login Now or Sign Up window (**fig-18**) will appear – (No need to click on Login or Sign Up)
7. Click on -> **No Thanks, just start my download.**

MySQL Installation on Windows

The screenshot shows the MySQL Community Downloads page for MySQL Community Server 8.0.22. A red box highlights the 'Select Operating System' dropdown menu, which is set to 'Microsoft Windows'. Below it, a 'Recommended Download' section features the MySQL Installer for Windows, with its icon and download link. Another red box highlights the 'Windows (x86, 32 & 64-bit), MSI Installer' download link.

Fig-16

The screenshot shows the MySQL Community Downloads page for MySQL Installer 8.0.22. A red box highlights the 'Select Operating System' dropdown menu, which is set to 'Microsoft Windows'. Below it, two download options are listed: 'Windows (x86, 32-bit), MSI Installer' and 'Windows (x86, 32-bit), MSI Installer'. The second option is highlighted with a red box. Both entries include file size (8.0.22, 2.5M and 405.2M respectively) and a 'Download' button. A note at the bottom suggests using MDS checksums and GnuPG signatures for integrity verification.

Fig-17

The screenshot shows the MySQL Community Downloads page with a focus on user authentication. It features 'Login Now or Sign Up for a free account.' buttons. A note explains that an Oracle Web Account provides advantages like fast access to MySQL software downloads and the ability to post messages in forums. A red box highlights the 'No thanks, just start my download.' button at the bottom.

Fig-18

MySQL Installation on Windows

2. Installation and Setup: Follow the steps shown in the video ->

<https://www.youtube.com/watch?v=WuBcTJnluzo>

1. Open the downloaded **msi** file
2. To start installing double click on the **msi** file
3. User Account Control warning may appear, read carefully and click yes to continue
4. MySQL Installer wizard will appear (**fig-19**)
5. Agree the License agreement and click <**Next**>
6. On <**Choosing a Setup Type**> select <**Developer Default**> click <**Next**> (**fig-20**) (warning may appear, read carefully and click yes to continue)
7. On <**Installation**> wizard click <**Execute**> to start installing products one by one → click <**Next**>
8. On <**Product Configuration**> click <**Next**>
9. On <**Type and Networking**> wizard select <**Standalone MySQL Server/Classic MySQL Replication**> and click <**Next**>
10. For server configuration type, select <**Development machine**> and click <**Next**> (**fig-21**)
11. On <**Accounts and Roles**> wizard give **MySQL root password** and store it in a secure place (**fig-22**). You can also set user account. <**Next**>
12. On <**Windows Service**> wizard, click <**Next**> with default settings
13. On <**Plugins and Extensions**> wizard, click <**Next**> with default settings
14. On <**Apply Configuration**> wizard, click <**Execute**> -> click <**Finish**> → <**Product Configuration**> wizard will appear again
15. Click <**Next**> for **MySQL Router Configuration** --> click <**Finish**> → <**Product Configuration**> wizard will appear again
16. Click <**Next**> for **Samples and Examples**--> <**Connect to Server**> wizard will appear. Select the server and **Check it** (**fig-23**) → click <**Next**>
17. On <**Apply Configuration**> wizard, click <**Execute**> -> click <**Finish**> → <**Product Configuration**> wizard will appear again → click <**Next**>
18. <**Installation Complete**> wizard will appear → click <**Finish**>
19. MySQL Workbench and MySQL Shell will appear

MySQL Installation on Windows

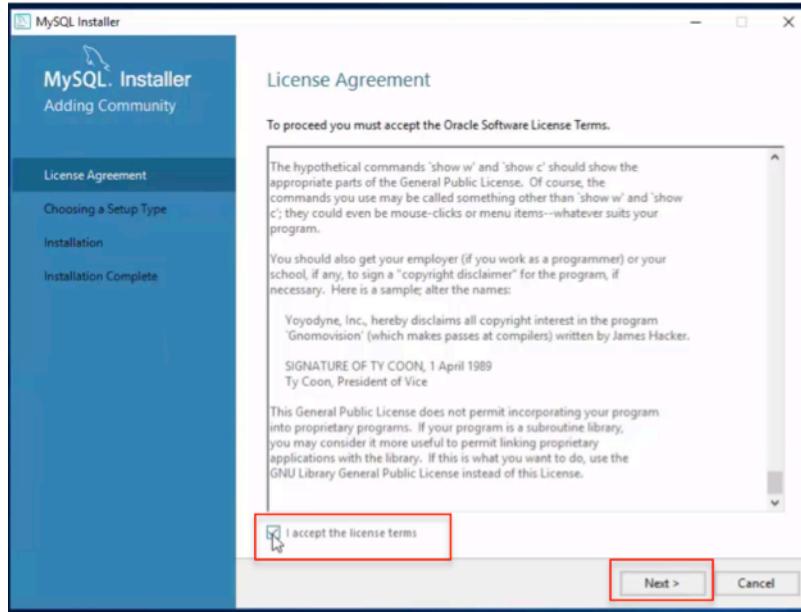


Fig-19

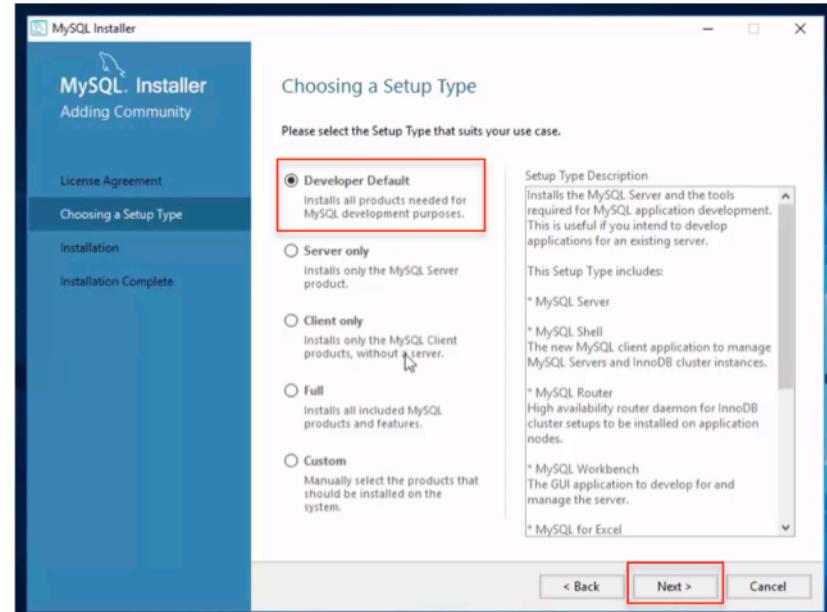


Fig-20

MySQL Installation on Windows

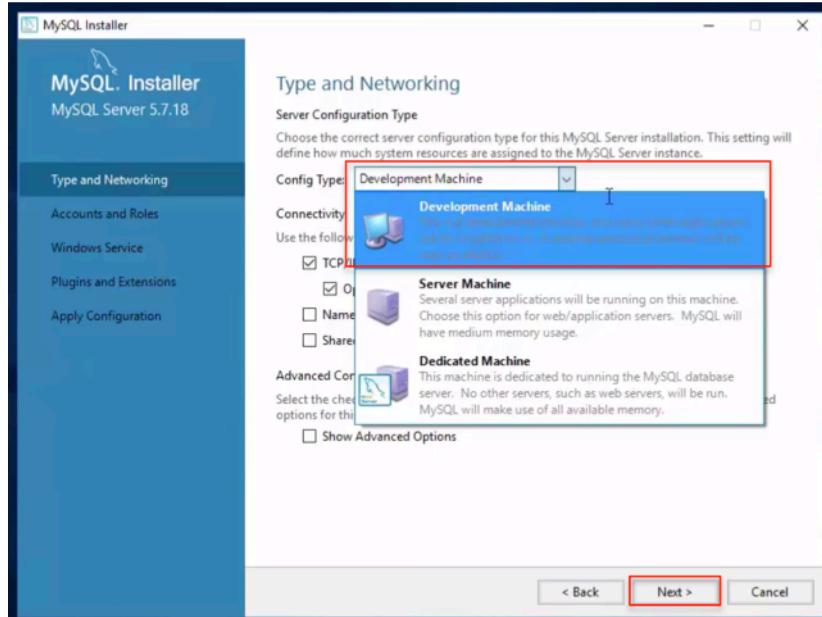


Fig-21

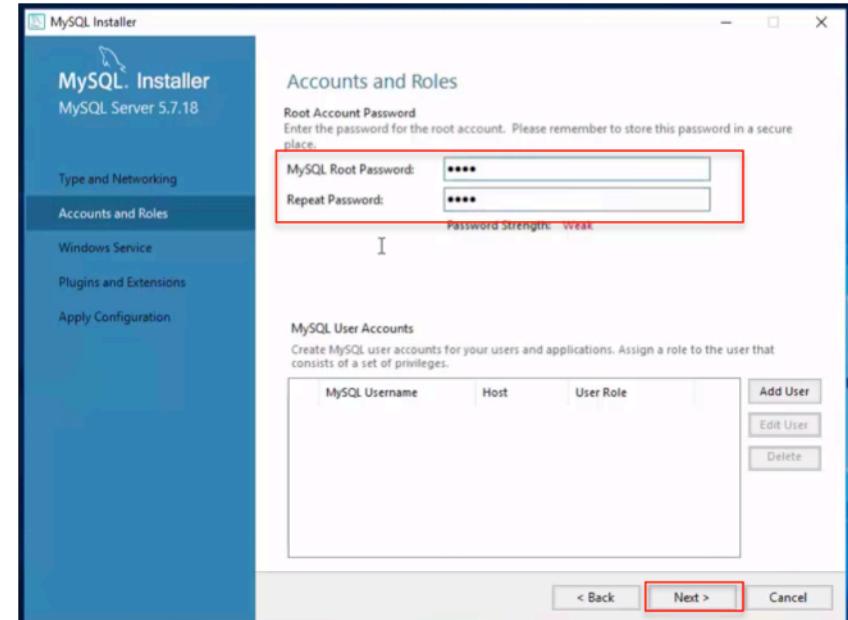


Fig-22

MySQL Installation on Windows

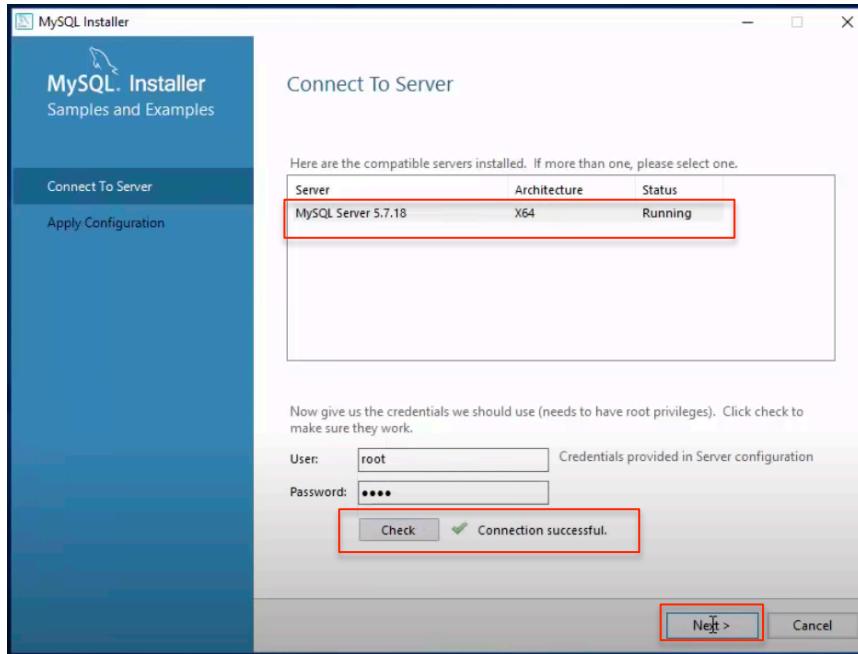


Fig-23

MySQL Installation on Windows

3. Connect to the MySQL Server using MySQL Client

1. Using MySQL Command Line

1. Go to <Start> menu → go to <All apps> → Expand <MySQL> → open <MySQL Command Line Client>
2. Give root password
3. Now it's set to interact with the databases
4. Write SQL command <**show databases;**> to see the databases present in the MySQL Server (**fig-24**)
5. See MySQL tutorial <https://www.w3schools.com/sql/> for more SQL statements

2. Using GUI - MySQL Workbench

1. Go to <Start> menu → go to <All apps> → Expand <MySQL> → open <MySQL Workbench>
2. Go to <Database> tab → select <**Connect to database**> (**fig-25**)
3. Select <**Local Instance MySQL Router**> in Stored Connection
4. Give root password and click <**Ok**> (**fig-26**)
5. Click <**Ok**> to connect to the MySQL Server
6. MySQL Workbench interface will appear (**fig-27**)

MySQL Installation on Windows

```
MySQL 5.7 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.18-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help,' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

Fig-24

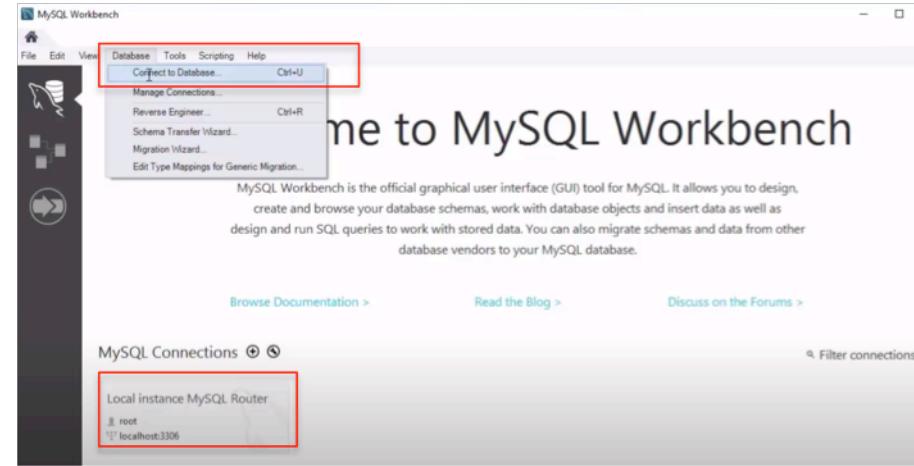


Fig-24

MySQL Installation on Windows

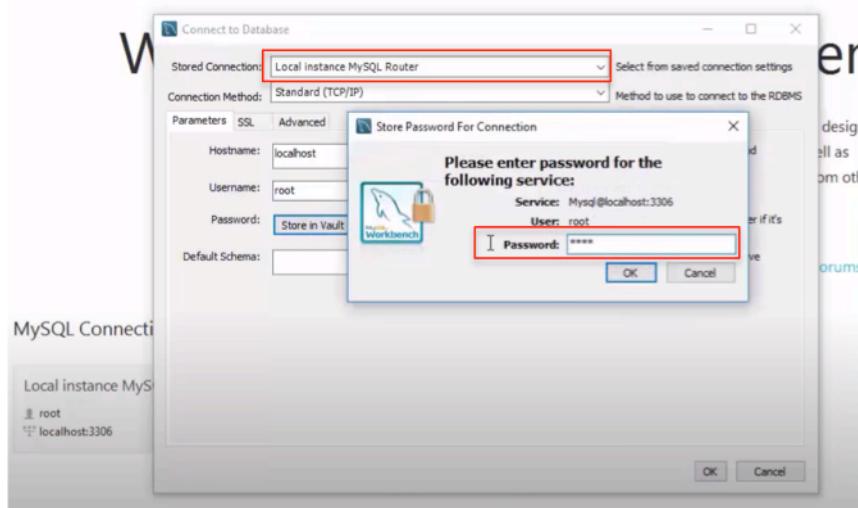


Fig-26

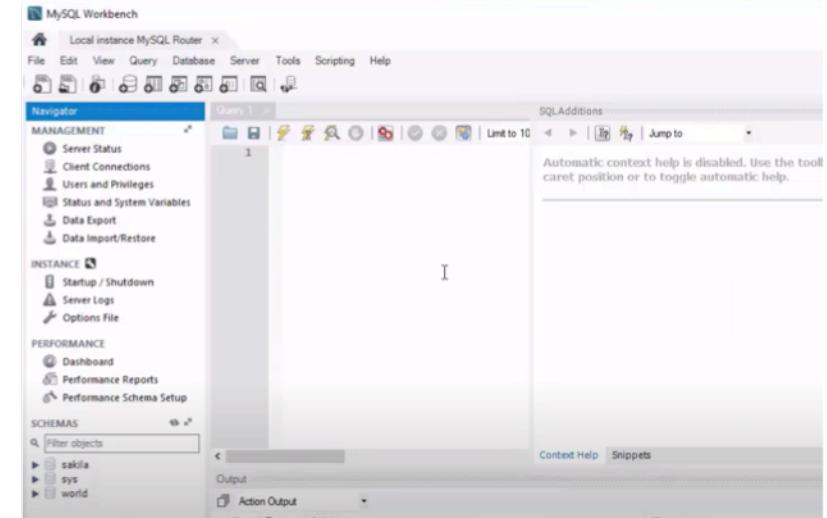


Fig-27

MySQL Installation on Linux

1. MySQL Community Server Installation:

Follow the installation steps show in video - <https://www.youtube.com/watch?v=3qD6zv7thdE>

1. Open up terminal and update repository using command - **\$ sudo apt update**
2. Once updated is done, Upgrade repository using command - **\$ sudo apt upgrade**
3. Install MySQL Server using command - **\$ sudo apt install mysql-server**
4. Check whether it is installed or not using command – **\$ mysql – version**
5. Set root password using command - **\$ sudo mysql_secure_installation**
 - Press **y** to setup VALIDATE PASSWORD component
 - 3 levels of password validation policy
 - Enter 0 for low, 1 for medium, and 2 for strong password
 - Now enter **new password**
 - It will ask to continue with password provided – press **y** to continue
 - It will ask to remove anonymous users – press **y** to remove
 - Disallow root login remotely? – press **y** to disallow
 - Remove test database and access to it? – press **y** to remove privileges on the databases
 - Reload privilege tables now? – press **y**
6. Go to mysql console and type command - **\$ sudo mysql –u root**
7. Now MySQL interface will appear
8. To check – type SQL statement **<show databases;>**
9. See MySQL tutorial <https://www.w3schools.com/sql/> for more SQL statements

MySQL Installation on Linux

2. To connect to server using MySQL Workbench, follow the below installation and setup steps

Follow the installation steps show in video - <https://www.youtube.com/watch?v=9aYLfp5870>

1. Go to MySQL Community Downloads page <https://dev.mysql.com/downloads/> and click on **MySQL Workbench** (**fig-28**)
2. Select operating system from dropdown menu – **Ubuntu Linux** (**fig-29**)
3. Click on <Download> – (**fig-29**)
4. Login Now or Sign Up window (**fig-30**) will appear – (No need to click on Login or Sign Up)
5. Click on -> **No Thanks, just start my download.**
6. Open downloaded file -> click on <Install> (**fig-31**)
7. Give system password for authentication – starts installing
8. Open **MySQL Workbench**
9. By default MySQL server starts on port 3306, click on this or add sign (+) – (**fig-32**)
10. Give root password set from terminal and check <Save password in keychain> box(**fig-32**)
11. Click <Ok> to open **MySQL interface** (**fig-33**)

MySQL Installation on Linux

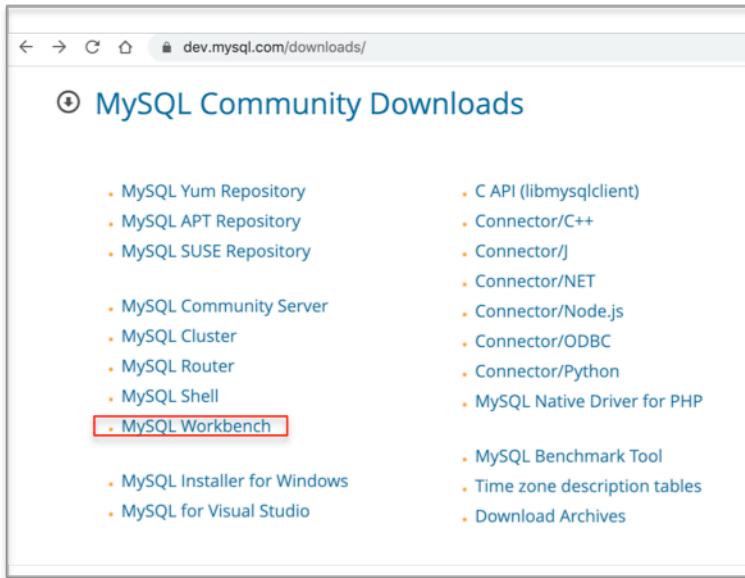


Fig-28

The screenshot shows the MySQL Workbench 8.0.22 download page. At the top, there are tabs for "General Availability (GA) Releases" (which is selected), "Archives", and a link to "MySQL Home". Below the tabs, the heading is "MySQL Workbench 8.0.22". The "Select Operating System" dropdown is set to "Ubuntu Linux" and is highlighted with a red box. The "Select OS Version" dropdown is set to "All". The page lists three available packages for "Ubuntu Linux 20.10 (x86, 64-bit), DEB Package":

Package	Version	Size	Action
Ubuntu Linux 20.10 (x86, 64-bit), DEB Package	8.0.22	29.9M	Download
(mysql-workbench-community_8.0.22-1ubuntu20.10_amd64.deb)			MD5: 372886070f55ba2c4dfd69710bfe11f0 Signature
Ubuntu Linux 20.10 (x86, 64-bit), DEB Package	8.0.22	21.5M	Download
(mysql-workbench-community_8.0.22-1ubuntu20.10_amd64.deb)			MD5: b85d8cace721f2434daced33f21c008d Signature
Ubuntu Linux 20.04 (x86, 64-bit), DEB Package	8.0.22	22.0M	Download
(mysql-workbench-community_8.0.22-1ubuntu20.04_amd64.deb)			MD5: 180fd775c5f628e2984935b0e6daf3df Signature

Fig-29

MySQL Installation on Linux

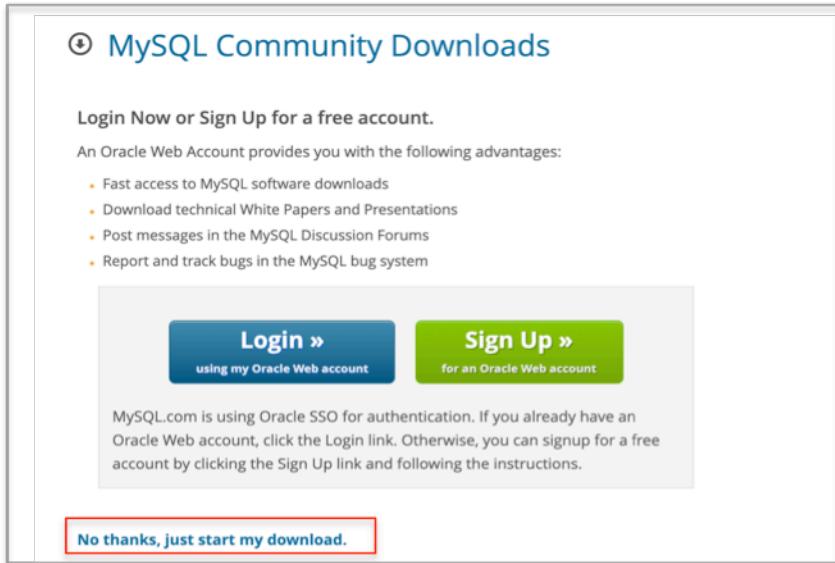


Fig-30

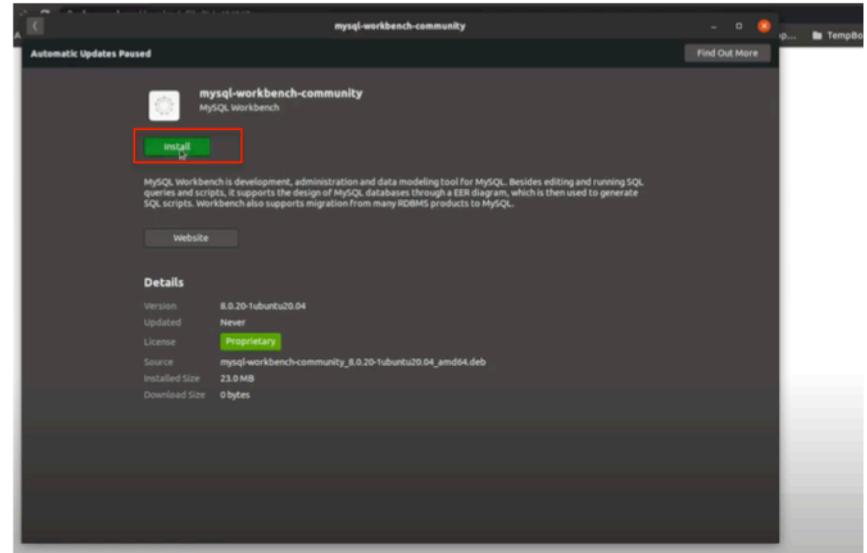


Fig-31

MySQL Installation on Linux

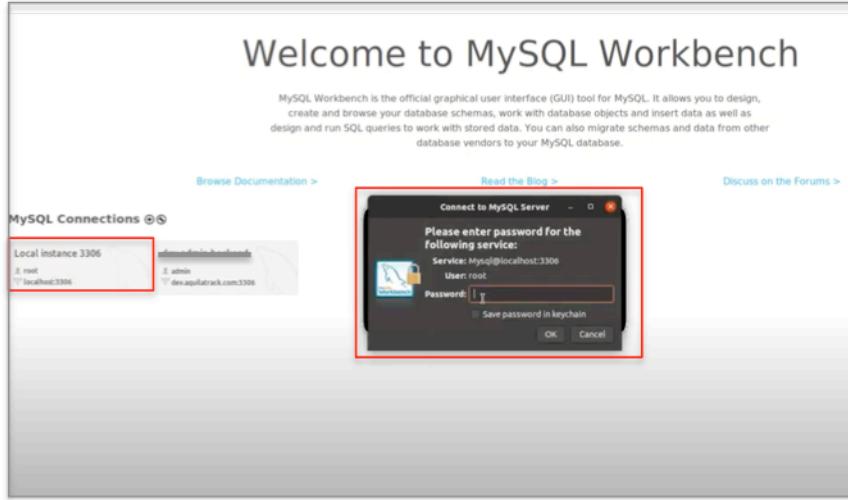


Fig-32

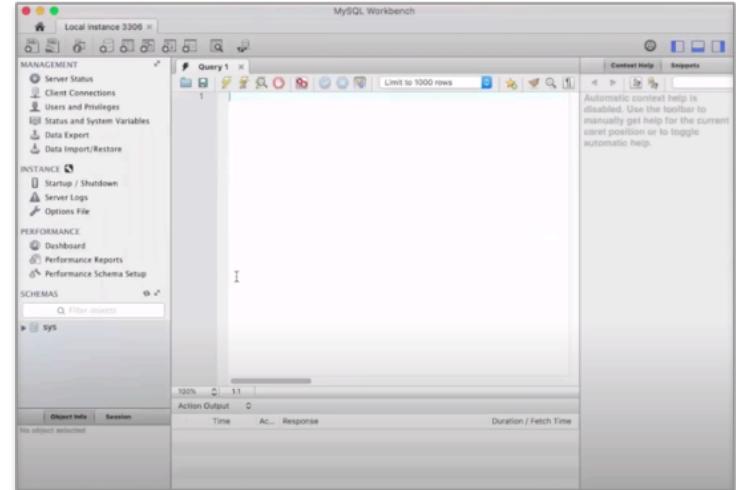


Fig-33

