

***DETERIORATING QUALITY OF AIR QUALITY***



**Table of Contents**

Introduction1

File v/s DBMS4

Relational Model5

ER Model6

**REPORT**

**Introduction**

We often read in newspapers about the deteriorating air quality of the metro cities. This has become the major concern recently and several steps are being taken in accordance with its improvement. The Environmentalist has declared an alarming situation in national capital of Delhi.

What is DBMS?

Database management system is a file handling software used for storing data and retrieving it whenever required. This system has helped us immensely in collecting and storing data in an organized format. These are easily accessible.

**File System v/s DBMS**

## DATA REDUNDANCY

1. Data redundancy occurs when the same piece of data exists at several points or places at a time. Same data is stored at multiple locations.
2. The air quality data is collected at the various stations. They store their own data and also share it with the centres at the state levels. This data is again shared by the state at the national level.
3. If the need arises that some other department has to access the same data, the same data has to be shared all over again.
4. Sensors at the stations keep on updating the data almost every minute, which leads to generation of huge amounts of data.
5. If any private institutions or individuals required data for research purposes then, the data has to be sent to them individually.

## DATA INCONSISTENCY

1. Data at different places is supposed to be the same but due to negligence or error the data differs from each other.
2. If the data of the sensors are being stored in the centralised system, then this is easier to manage.
3. For example, today’s data set has been sent to the state level but while forwarding it to the national station, the state level in charge sends yesterdays’ data. Now because of this the data at the national level for the respective state would be duplicated for two consecutive days.
4. For each sensor a different table has to be created when implementing a file system.
5. If some failure occurs while sending the files from the station to the state level then the files have to be shared all over again.

## EASY ACCESS TO DATA

1. The sensors generate a lot of data as it is working every minute.
2. The data will be updated every minute which would be difficult to be analysed manually.
3. Hence any GUI System is created for visualising and analysing data which can work with a centralised database and the memory is easily accessible.
4. The data can be shared through API directly with the institutions or individuals for research purposes.
5. Various insights will be available to us because of easy access which would help in stopping the deteriorating conditions of the environment.

## DATA ISOLATION

1. The latest updated data will not be available.
2. In case a programmer writes a code, then he’ll not receive the latest data rather he’ll have to work with stale data.
3. If there is a centralised data, then a single code can be created to analyse data from all the different levels and stations.
4. Programmers at different levels will have to write their own individual codes.
5. The files at the station level may have different formats, then there will be a problem in processing the data.

## DATA INTEGRITY

1. The government can be a main cause of data being hampered. They can alter the data to demonstrate a clean image in front of the public.
2. If DBMS is used then, this can surely be avoided and the data integrity can be maintained.
3. As we know that the sensors are the one going to write the data and hence all users should only be granted read access.
4. In the file system any one can modify the data for whatever reasons.
5. The concept of foreign keys cannot be implemented in a file system and hence the data from various stations if necessary, cannot be grouped together based on date, time, etc. Example : If every hour we want to calculate city and state averages for the concentration levels the data from all stations has to be grouped and then it has to be calculated.

## ATOMICITY OF UPDATES

1. If the sensors have got into some malfunctioning, then Null data will be updated in the file system continuously which cannot be altered.
2. This would create a serious problem in the analysis system. If suppose the ozone sensor has stopped working, then the file system will receive Null values till the time it is repaired. This would alter the data analysis.
3. Hence if a database is used then a condition can be put up such that if a sensor returns NULL values then those should be discarded and some other values should be substituted instead using some statistical analysis, based on time and location a time series analysis can be performed.
4. If any transaction has just been completed and power failure occurs, still the transaction has been saved. But if a power failure occurs while writing the data in a file then it has to be saved first to save the transactions permanently.

## SECURITY

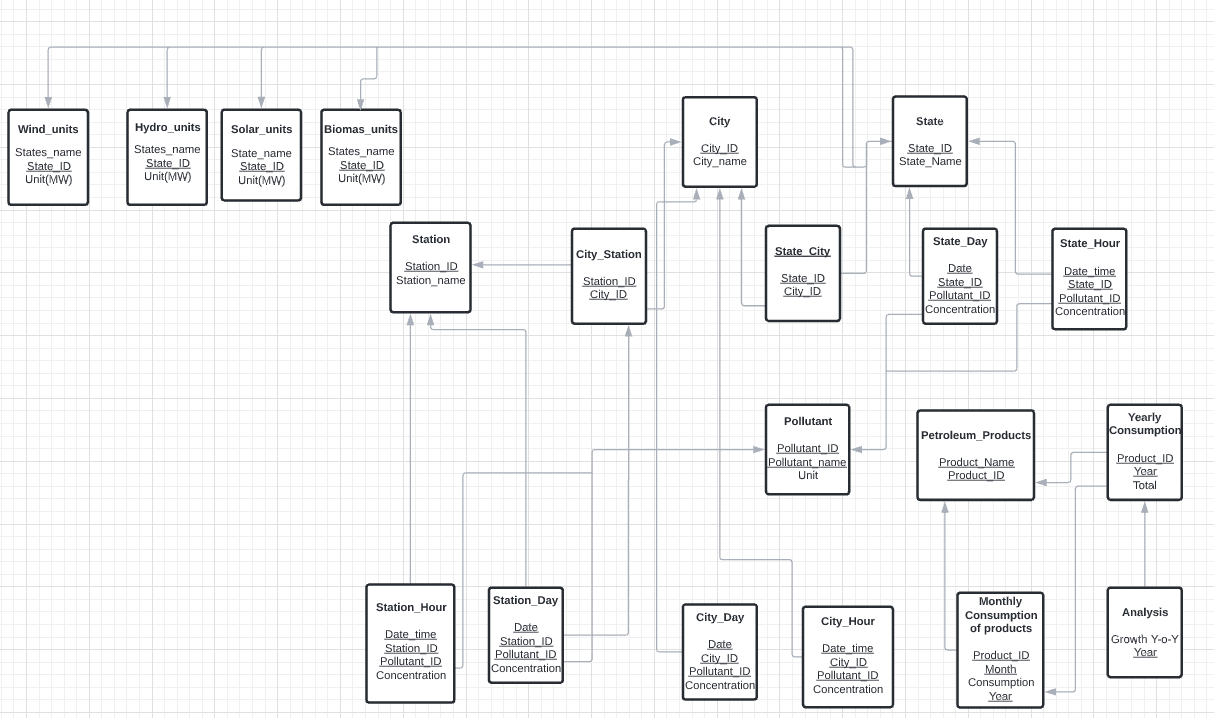
1. In the case of a file system, the data will be stored in separate computers in various stations. That computer will be having several users. Hence, data can be easily hampered or altered.
2. Also, the computers can be hacked easily. Hundreds of computers cannot be provided with high security.
3. In DBMS the central system will be providing high security. All the data will be controlled using a single security system and the legitimacy of data can be thus maintained.
4. No user needs to have an update access on the data.
5. In a database system all stations can view only the data shared by their respective station and same with the state level. For accessing the whole database they too would have to use an API.

## CONCURRENT ACCESS

1. Several sensors can write data at the same time in a DBMS system. This is not possible in the file system. In that case we’ll have to compile system files separately.
2. In a DBMS system all the sensors can write the data at the same place and at the same time. The data from all the stations, states and nations can be combined, stored and accessed from this system itself. Whereas in the case of a file system the data has to be individually combined at different levels and then be shared.
3. The data stored in the file system will be available to only a few people. If some other person wants access to it, then he’ll have to undergo long procedures in order to receive the data.
4. If we create an API of the dataset, then the users throughout the nation can easily access it and get the latest updated data.

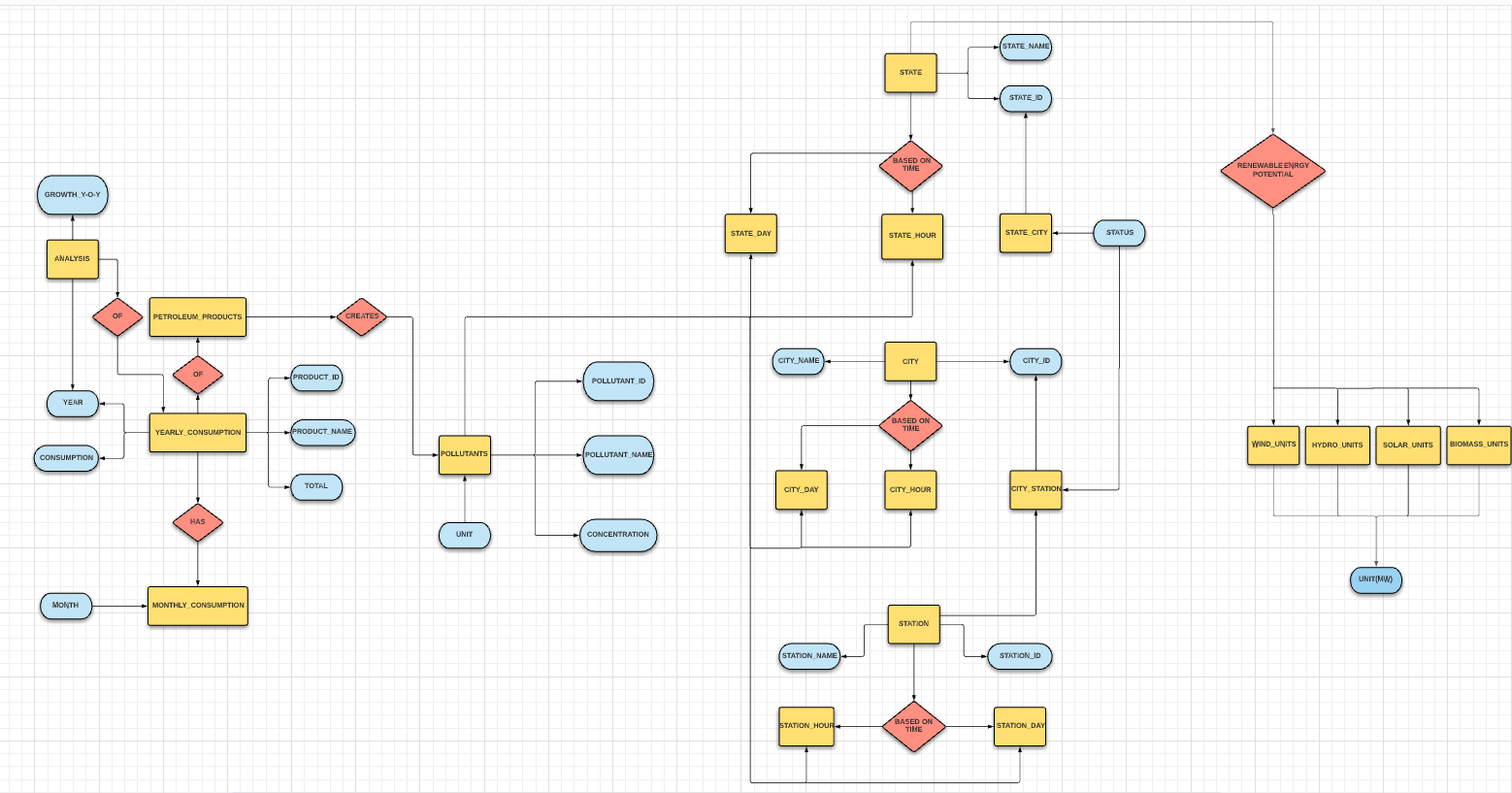
**RELATIONAL MODEL**

Relational Model is the data representation that demonstrates the relation in tables. The rows of the table are connected with some real-world relation.



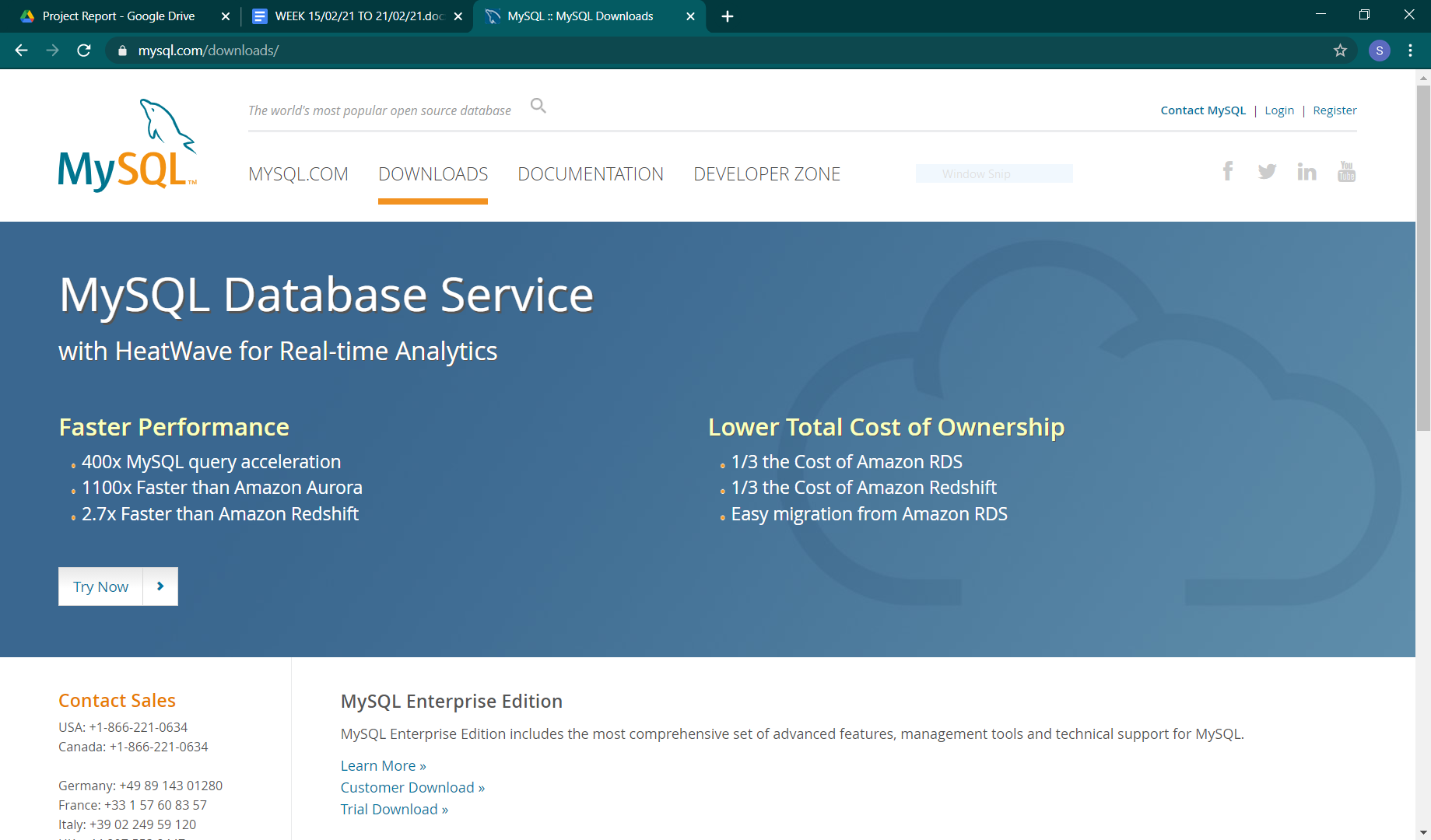
**ER MODEL**

The entity relationship diagram is basically considered to be the blueprint of the database which is finally converted to a database. It defines the relationship between elements and is a high-level data model.



|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Super key** | **Candidate key** | **Primary key** |
| Station | Station\_Id, Station\_name, [Station\_Id, station name] | Station\_ID,State\_name | Station\_ID |
| City\_Station | [Station\_ID, City\_ID],  [Station\_ID, City\_ID] | [Station\_ID, City\_ID] | [Station\_ID, City\_ID] |
| City | City\_ID, City\_name, [City\_ID, City\_name] | City\_ID, City\_name | City\_ID |
| State | State\_ID, State\_name | State\_ID, State\_name | State\_ID |
| State\_City | State\_ID, City\_ID, [State\_ID, City\_ID] | [State\_ID, City\_ID] | [State\_ID, City\_ID] |
| State\_Day | [Date, Pollutant\_ID, State\_ID] | [Date, Pollutant\_ID, State\_ID ] | [Date, State\_ID,Pollutant\_ID] |
| State\_ Hour | [Date\_Time, State\_ID, Pollutant\_ID] | [Date\_Time, State\_ID, Pollutant\_ID] | [Date\_Time, State\_ID, Pollutant\_ID] |
| Pollutant | Pollutant\_ID, Pollutant\_name, [Pollutant\_ID, Pollutant\_name],  [Pollutant\_ID,Pollutant\_name,  unit],[Pollutant\_ID,Unit],  [Pollutant\_name,Unit] | Pollutant\_ID, Pollutant\_name | Pollutant\_ID |
| Station Hour | [Date\_time, Pollutant\_ID, Station\_ID]  [Date\_time,Pollutant\_ID,Station\_ID,concentration] | [Date\_time,Station\_ID, Pollutant\_ID] | [Date\_time,Station\_ID, Pollutant\_ID] |
| Station Day | [Date,Station\_ID,Pollutant\_ID],  [Date,Station\_ID,Pollutant\_ID,concentration] | [Date,Station\_ID,Pollutant\_ID] | [Date,Station\_ID,Pollutant\_ID] |
| City Hour | [Date\_time, Pollutant\_ID, City\_ID]  [Date\_time,Pollutant\_ID,City\_ID,concentration] | [Date\_time,City\_ID, Pollutant\_ID] | [Date\_time,City\_ID, Pollutant\_ID] |
| City Day | [Date,City\_ID,Pollutant\_ID],  [Date,City\_ID,Pollutant\_ID,concentration] | [Date,City\_ID, Pollutant\_ID] | [Date,City\_ID, Pollutant\_ID] |
| Petroleum Products | Product\_Name, Product\_ID  [Product\_ID,Product\_Name] | Product\_Name, Product\_ID | Product\_ID |
| Yearly Consumption | [Product\_ID, Year] | [Product\_ID, Year] | [Product\_ID,Year] |
| Monthly Consumption | Product\_ID, Month, Year, [Product\_ID, Year], [Month, Year, Product\_ID] | Product\_ID, Year, Month | Product\_ID |
| Analysis | Year,[Year,Growth Y-o-Y] | Year | Year |
| Wind Units | State\_ID, State\_name, [State\_ID, State\_name] | State\_ID, State\_name | State\_ID |
| Biomass Units | State\_ID, State\_name, [State\_ID, State\_name] | State\_ID, State\_name | State\_ID |
| Hydro Units | State\_ID, State\_name, [State\_ID, State\_name] | State\_ID, State\_name | State\_ID |
| Solar Units | State\_ID, State\_name, [State\_ID, State\_name] | State\_ID, State\_name | State\_ID |

**INSTALLATION OF SOFTWARE**



We have downloaded and installed MySQL software for performing our query operations in DBMS.

Following are the steps to be followed:

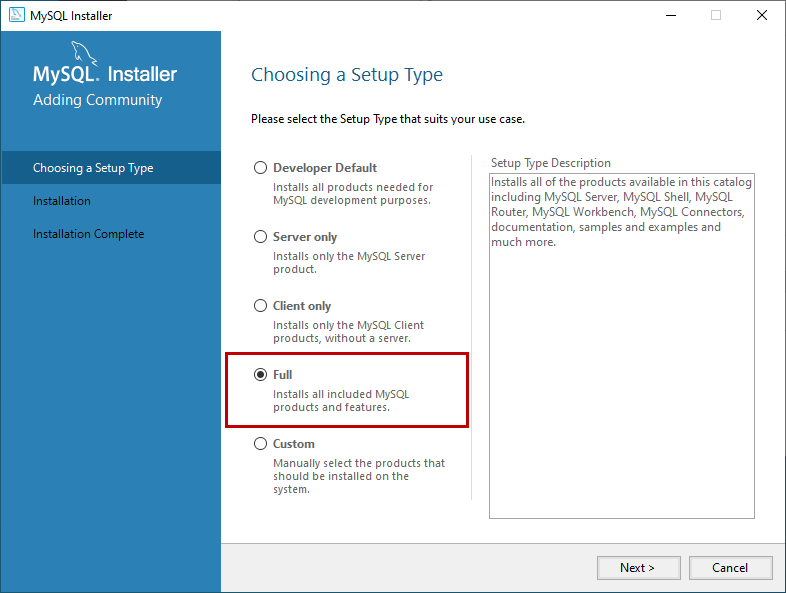
Step1

Download the Mysql software from MySql.com

Step2

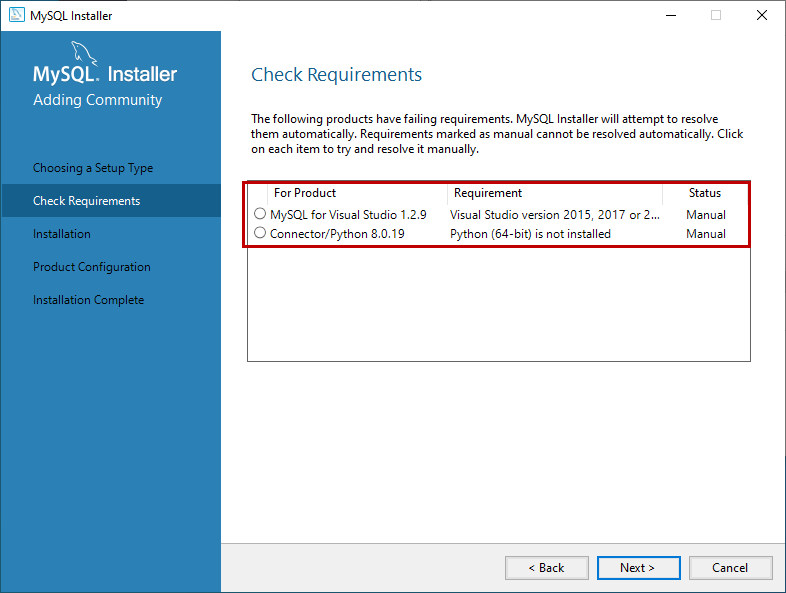
Extract the files from the zip and run the setup.

Step3



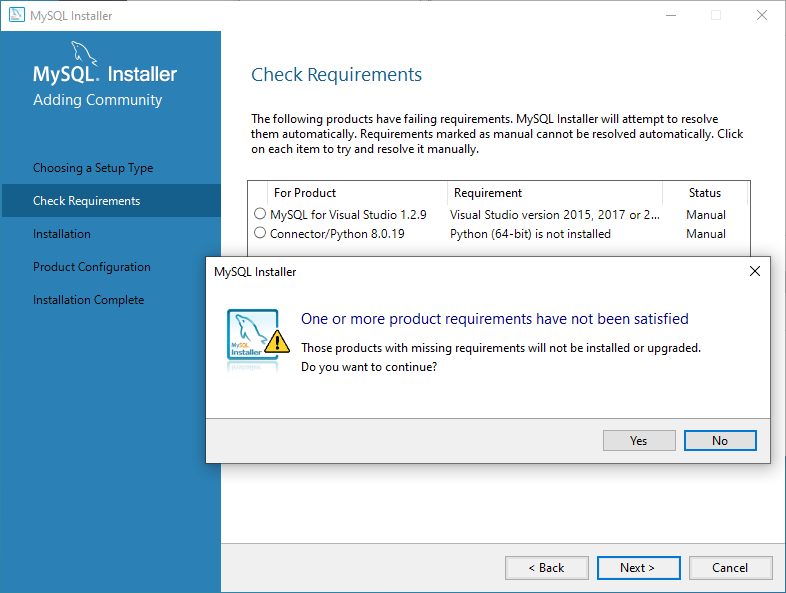
Step4

Before installation the software would check for the libraries required and whether the settings are appropriate or not.



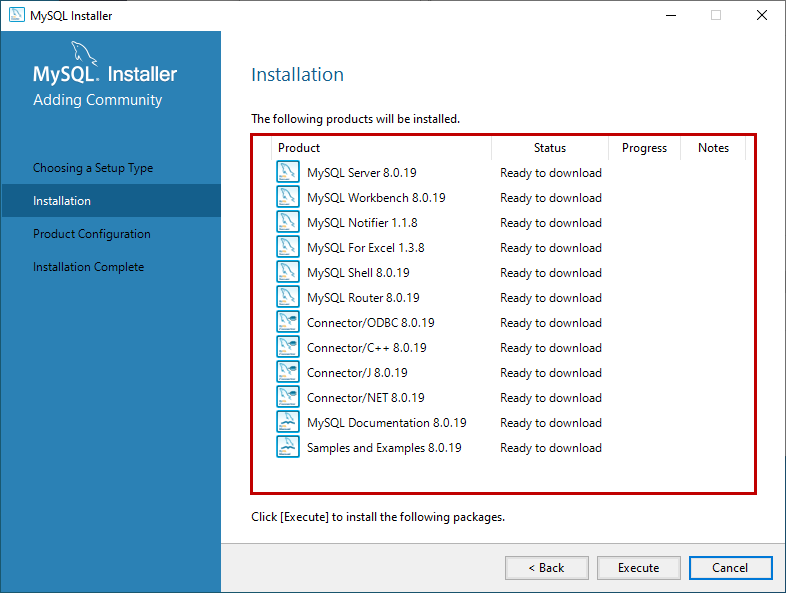
Step5

A pop up window appears giving us a warning regarding the presence of VSC or Python. Click on Yes..



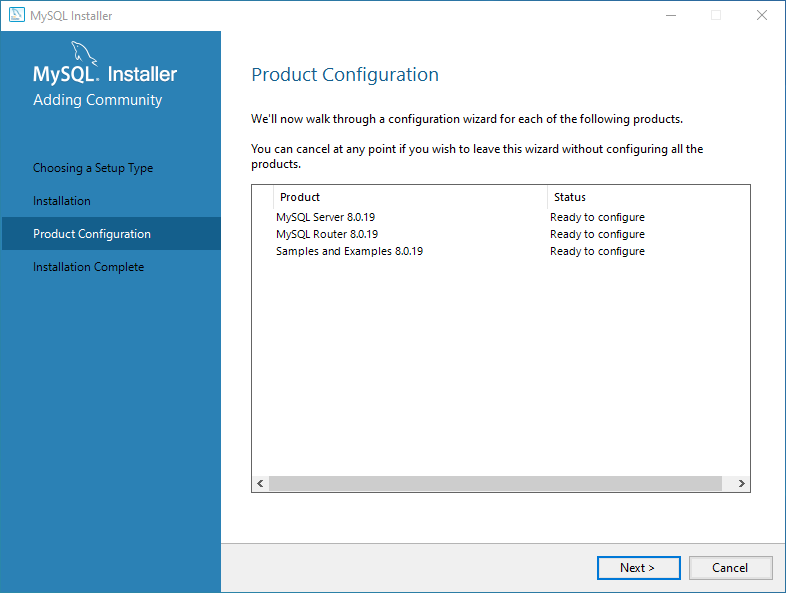
Step6

Then Click on Execute to install all the packages.



Step7

After the installation of packages is completed. A dialog box appears. Then Click on Next to continue.



Step8

After this step your Sql would be ready to resolve the queries. Just click on Finish, to complete the installation process.