# Introduction

The Lincoln, United Kingdom based is Evans Conference Management Company which arrange academic conference for third party organizations like universities in all across the United Kingdom. And according to the scenario the database system should be developed to carry out the business which should have capability to store all the information needed for the Evans Conference management company.

The data displayed in the scenario is not fully normalized so, I have to normalized and then create a database so that I can store all the information needed for the Evans Conference Management Company.

# Task-1



The graphical depiction of an information system that signifies the interactions between people, items, ideas or events in that system is known as entity relationship diagram (ERD),which is also known as an entity relationship model.

And also the given paper based database is normalized up to third normalized form and then entity relationship diagram has been drawn accordingly.



Fig 1: Entity relationship model for the proposed database system for Evans Conference Management.

B.

The process of reducing redundant data, removing insert, delete anomalies and logically splitting any compound is called normalization. There are insert and delete anomalies in the table of scenario, so I have normalized the data up to third normalize form.

For example of delete anomalies: if venue information is deleted then the conference information associated with venue will also be deleted.

To be in first normalization (1NF) the columns should contain only single values and then stored value in that column should be the same domain. Similarly, in 1NF there should not be non-repeating groups but there are repeating in document 1 like: venue ID: MMU and location: Manchester are repeating many time. So, I have divided two tables one with repeating and another with non-repeating groups.

Whereas to be in second normalization (2NF) form the table should in first normal form and should not have any partial dependencies. Here, the below shown table are in both first and second normalization so I have kept them together.

**1NF / 2NF**

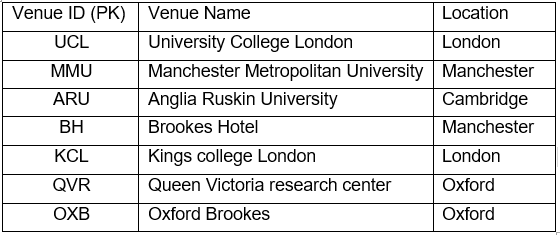
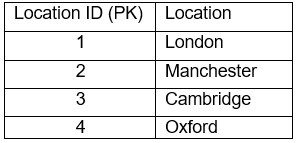


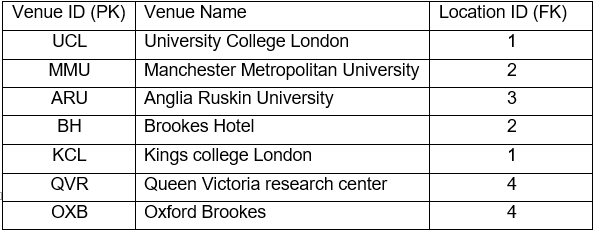


Fig 2: first and second normalization table.

Finally, to be in third normal form the table should be in first and second normal form and also should have any transitive dependencies. The below shown table are in third normal form and also does not have any transitive dependencies.

**3NF**





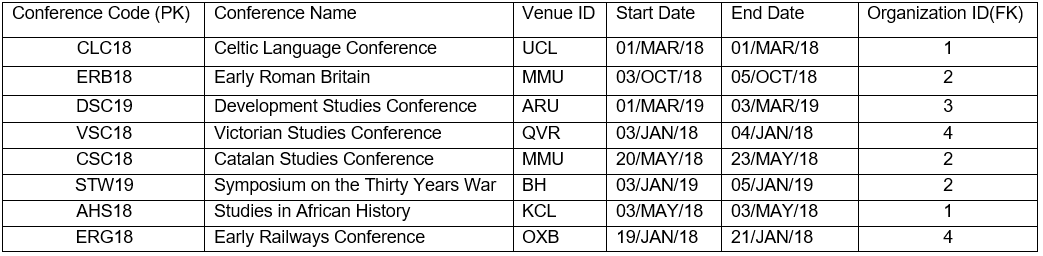


Fig 3: Third normalization table.

C.

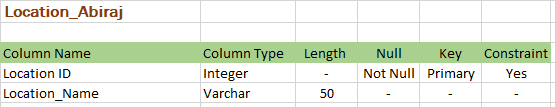


Fig 4: Data dictionary of location.

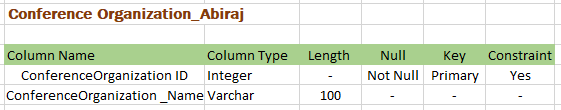


Fig 4: Data dictionary of Conference Organization.

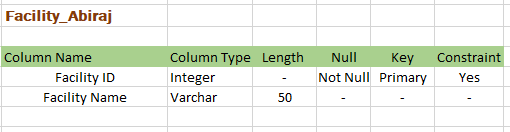


Fig 5: Data dictionary of Facility.

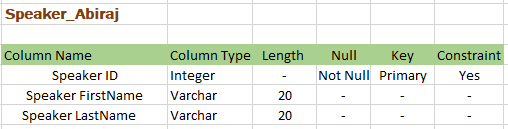


Fig 6: Data dictionary of Speaker.

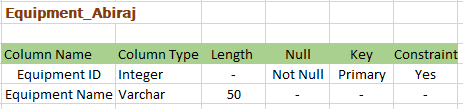


Fig 7: Data dictionary of Equipment.

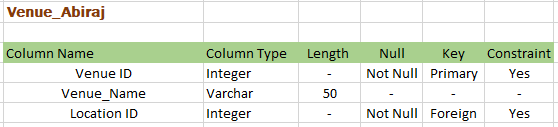


Fig 8: Data dictionary of Venue.

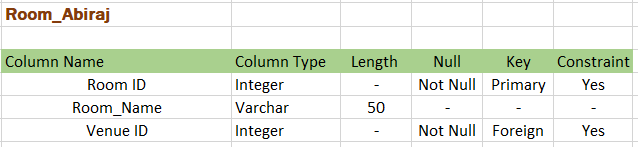


Fig 9: Data dictionary of Room.

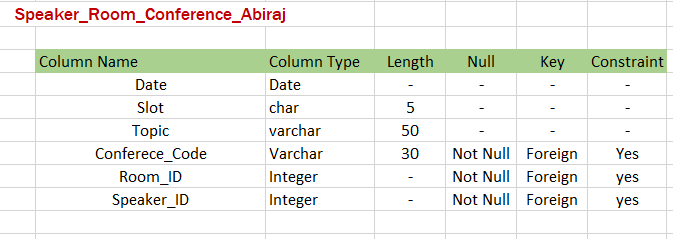


Fig 10: Data dictionary of Speaker\_Room\_Conference.

# Task-2

A.

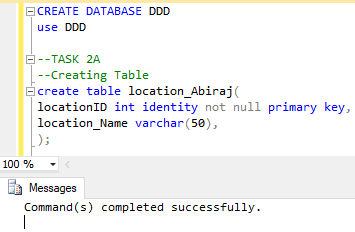


Fig 11: location table created successfully.

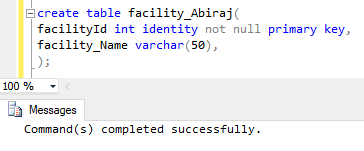


Fig 12: location table created successfully.

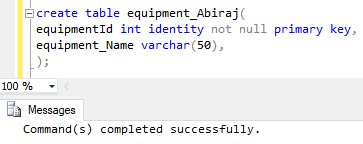


Fig 13: Equipment table created successfully.

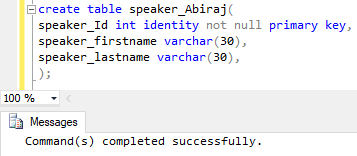


Fig 14: Speaker table created successfully.

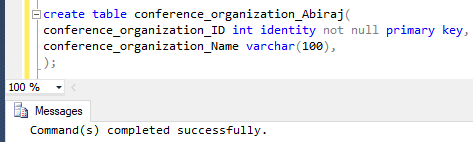


Fig 15: Conference table created successfully.

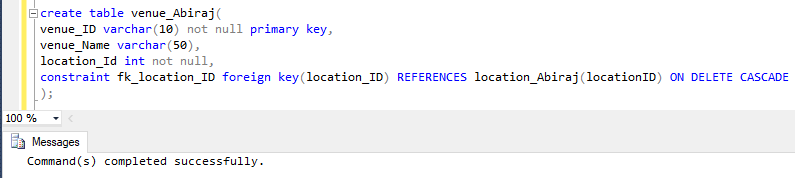


Fig 16: Venue table created successfully.

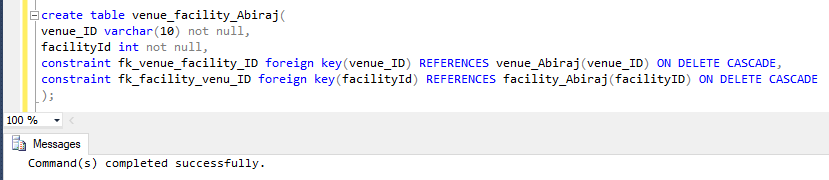


Fig 17: Venue\_Facility table created successfully.

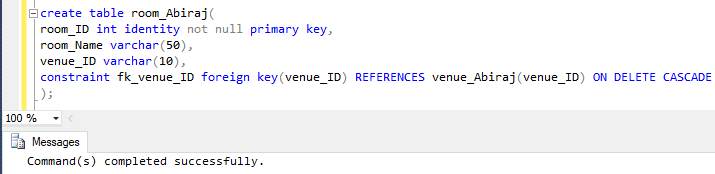


Fig 18: Room table created successfully.

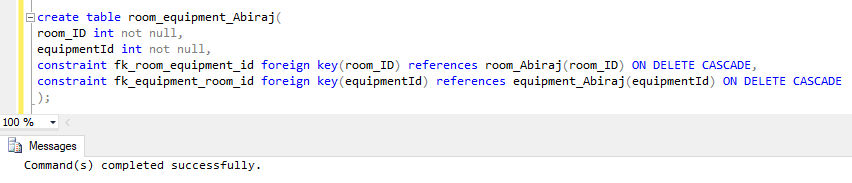


Fig 19: Room\_Eqipment table created successfully.

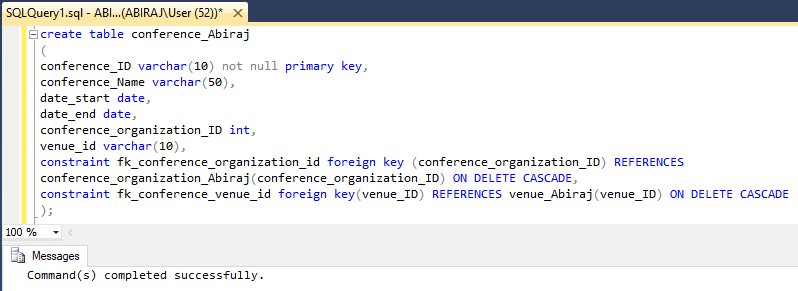


Fig 20: Conference table created successfully.

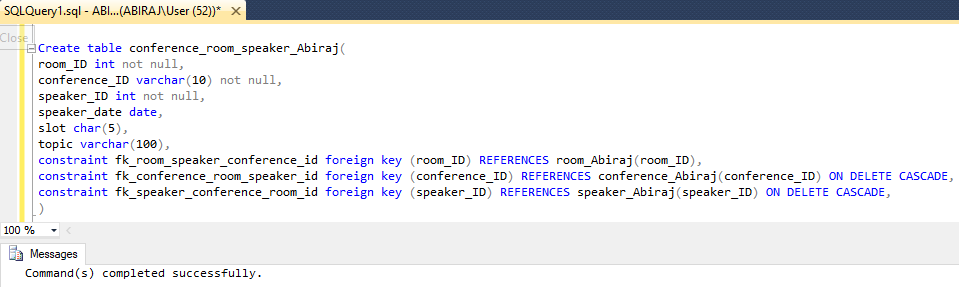


Fig 21: Conference\_Room\_Speaker table created successfully.

B.

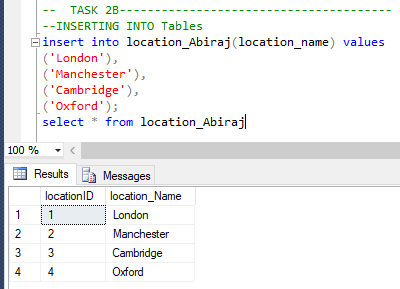


Fig 22: value successfully inserted into location.

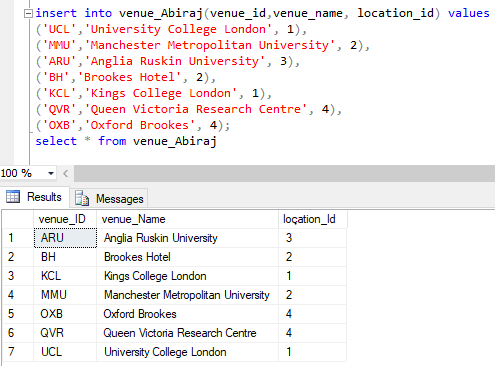


Fig 23: value successfully inserted into Venue.

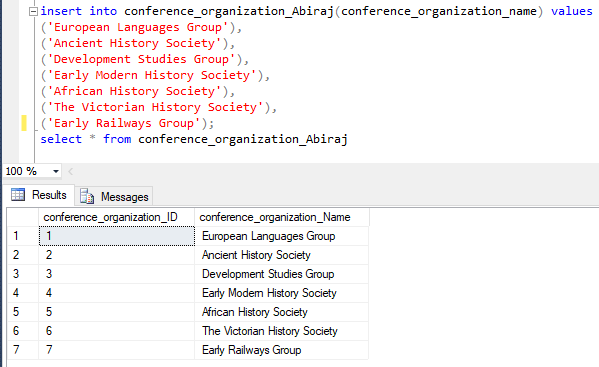


Fig 24: value successfully inserted into Conference Organization.

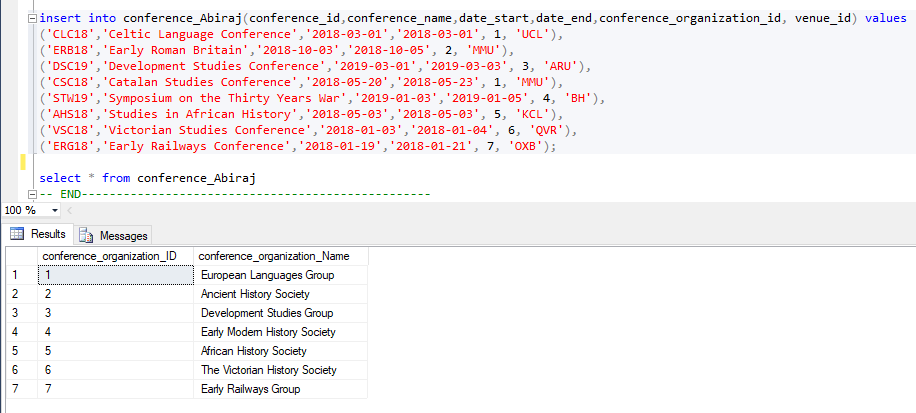


Fig 25: value successfully inserted into Conference.

C.

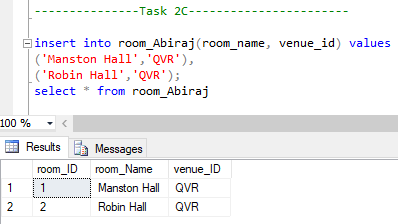


Fig 26: value successfully inserted into room.

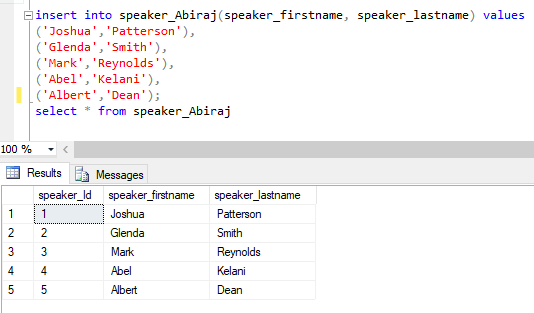


Fig 27: value successfully inserted into speaker.

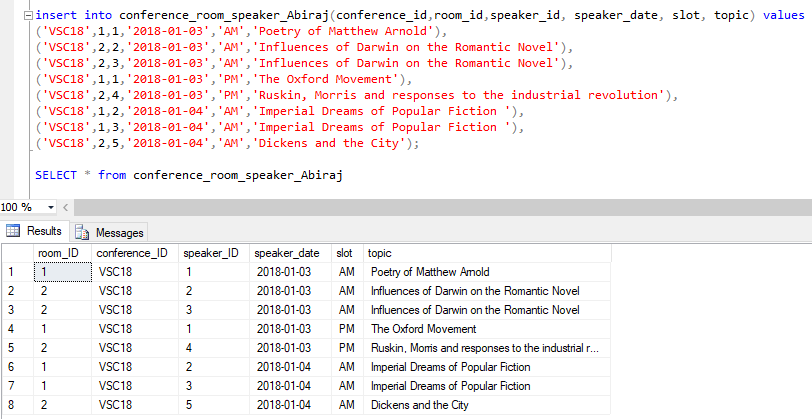


Fig 28: value successfully inserted into Conference\_Room\_Speaker.

D.

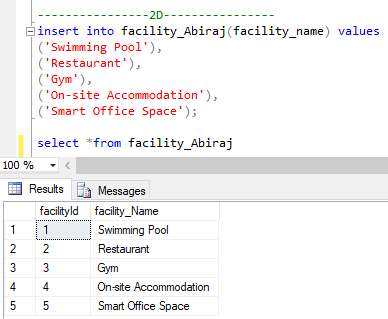


Fig 29: value successfully inserted into the facility.

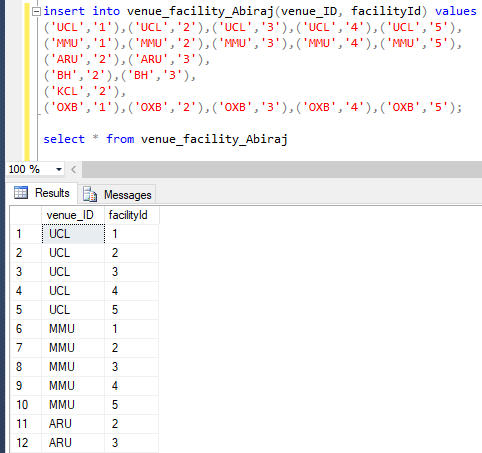


Fig 30: value successfully inserted into the Venue\_Facility.

E.

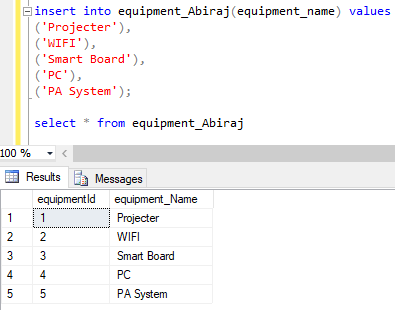


Fig 31: value successfully inserted into the equipment.

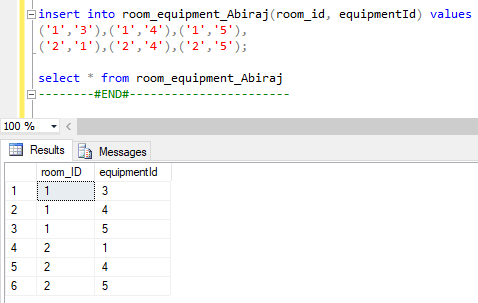


Fig 32: value successfully inserted into the Room\_Eqipment.

F.

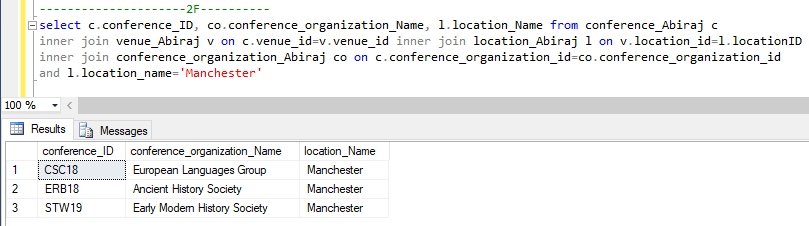


Fig 33: Query successfully run which select the conference code and conference organization for conference in Manchester.

G.

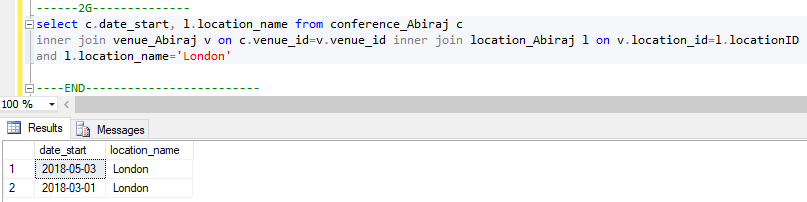


Fig 34: Query successfully run that selects start dates for all London based conferences.

H.

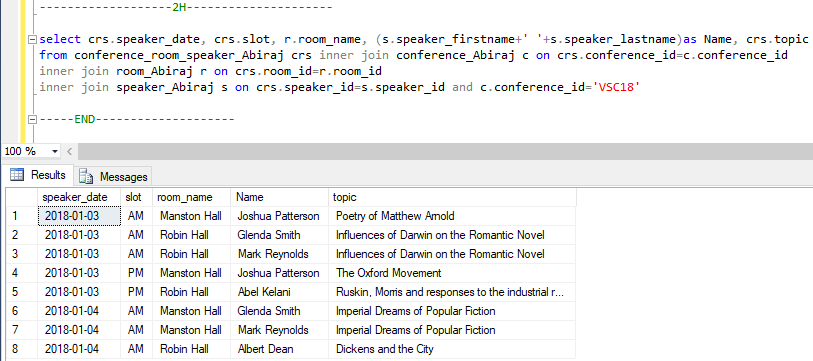


Fig 35: Query successfully run that shows the details for conferences VSC18.

I.

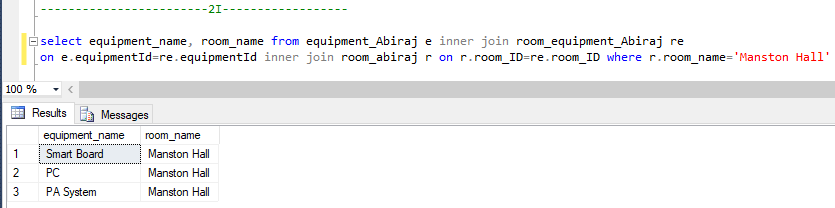


Fig 36: Query successfully run that shows the equipment in Manston Hall.

J.

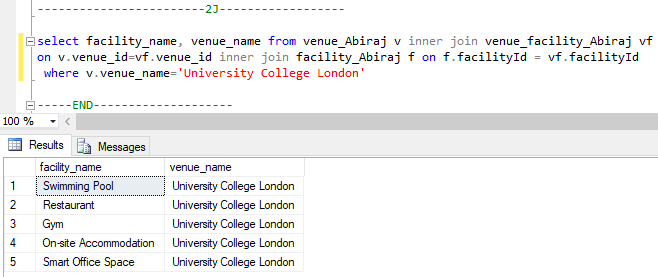


Fig 37: Query successfully run that shows the Facilities available at University College London.

K.

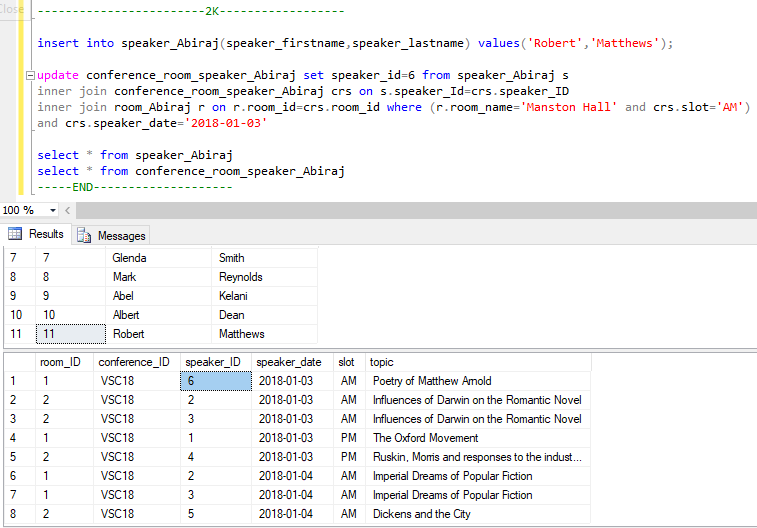


Fig 38: Query successfully updated for the AM slot in Manston Hall on the 03/JAN/2018 to Robert Matthews

L.

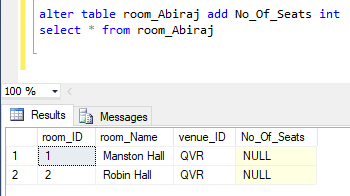


Fig 39: database successfully altered to keep a record of number of seats available in each room.

# Task - 3

The value that is calculated from the columns of same table is known as derived data. And if we want to find the data being based on another table then in this case derived data is used. For example wage of a person can be found by the time he has worked **OR** age can be found from the date of birth.

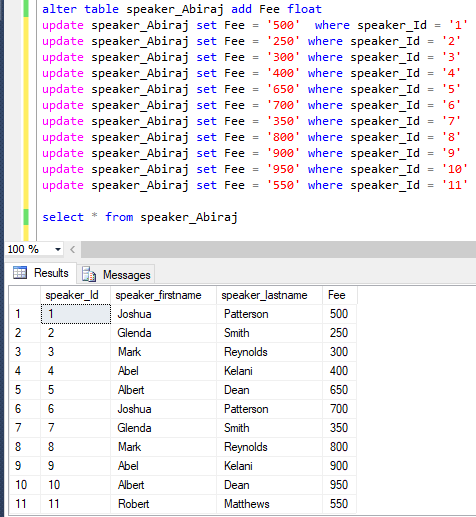


Fig 40: successfully added fee in venue for the process of total extra expenses cost.

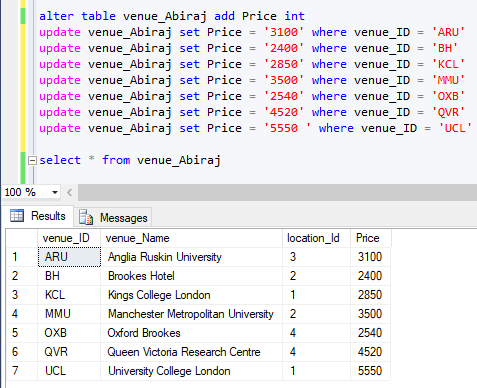


Fig 41: adding price in venue for the process of total extra expenses cost.

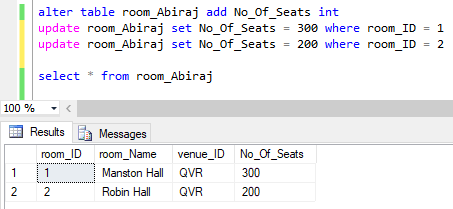


Fig 42: successfully added number of seats available in room.

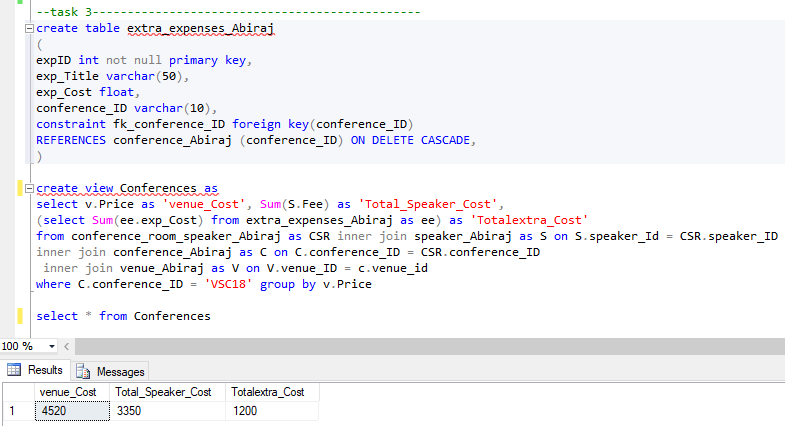


Fig 43: Successfully created extra expenses table and then derived costing information.

# Task-4

A database that involves of more files located in different locations that interconnect passing through the computer networks is known as distributed database system. Also, a distributed database system do not have any multiprocessing arrangement.

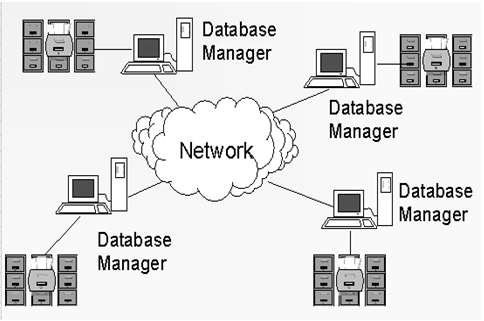


Fig 44: distributed database system.

There are several advantages of distributed database system which are explained below:

* **Quick response:** If all the data is kept in one location and when users try to fetch that data from multiple locations then it takes time to response for multiple users who are fetching data from multiple time, due to over requesting load. Instead of that if data is kept in different sites then the response time to user is less, due to less requesting load.
* **Flexible growth:** The system can be easily expanded in future than centralized database system. If a new system need to be expanded in centralized database system then it requires hard work whereas in distributed system we can easily add new system with files and connecting them to the distributed system without any large labors and interference in current function.
* **Provide improved availability and reliability:** if one system is failed then the whole system will be down in centralized database system. But in distributed system the other system will work continually even the one part of the system fails.
* **Decrease operating cost:** if the one part of the system needs to be maintain or update in distributed system then no need to down the whole system. If we have to down the whole system and again make up for maintenance or update then it will add to your additional operating cost. As this happens in centralized system.

The disadvantages of distributed database system are as follows:

* It is costly as it is complex and also it is difficult to maintain.
* Database need to be secured in all the sites where the database are stored. So, it is challenging to provide security in distributed database system and also tough to secured the setup joining all the nodes.
* There could be data redundancy because of the data is kept in different locations and also it will be problematic to maintain the reliability of data.
* It is somewhat complex and because of data is distributed in different locations it is difficult to ensure that the user got the homogenous outlook of database.

According to the scenario Evans Conference Management Company manages academic conference for third party organization across all over the United Kingdom. Hence, they might have branches too in across all over the United Kingdom.

* So, to implement that to store the data from all the branches of Evans Conference Management Company then they have to implement distributed database system.
* After implementing distributed database system all the data stored in databases in different location should be secured for data protection as well as they have to secure the setup connections in all the nodes.
* And to expand the system in future they do not have to worry about that just they need to add the computer with data and connect it to the distributed database system.

In this way the distributed database system can be contributed according to the scenario.

# Task – 5

A.

Evaluation of work.

* **Conceptual work:** the Lincoln, United Kingdom based is Evans Conference Management Company which arrange academic conference for third party organizations like universities in all across the United Kingdom. And according to the scenario the database system should be developed to carry out the business which should have capability to store all the information needed for the Evans Conference management company.
* **Logical work:** after understanding the requirements I have made entity relationship diagram and then did normalization to reduce the data redundancy. So, to make entity relationship diagram I have used visual paradigm software from where I easily developed this and after that did normalization. In normalization I have break down the documentation 1 table into three parts 1NF, 2NF and 3NF after that those data which came after normalizing the documentation 1 table up to 3Nf it became the final result to implement it into the query.
* **Physical work:** Finally, after making entity relationship diagram and doing normalization I wrote that final outcome of normalization into structure query language (SQL) to make able to store all the information needed for the Evans Conference management company into the database. So, I have used SQL Server management studio 2014 software to implement that final outcome of normalization into the SQL query.

Likewise, I created database name as DDD and then started to create the table according to the requirements like: room, venue, conference, speaker, location, facility and so on also then inserted all the data given in the scenario. Except this I also updated query in some cases like updating speaker and then altering the table room adding new column as number of seats available and so on.

Finally, in this way I created the database that has capability to store all the information needed for the Evans Conference management company.

# Conclusion

In conclusion, the whole requirements of Evans Conference Management has been provided accordingly. By breaking the table up to third normalization form which is done by removing the anomalies and then partial and transitive dependencies and making the entity relationship diagram of that normalization table and finally did activities like creating table, updating and altering it and so on.

In this way all the requirements of Evans Conference Management successfully completed.