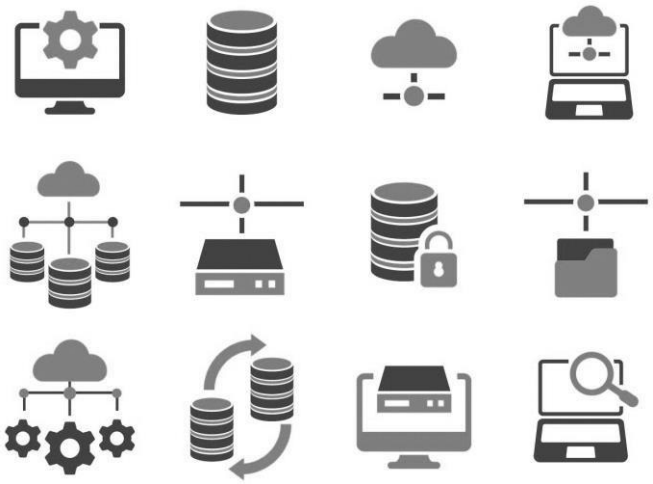
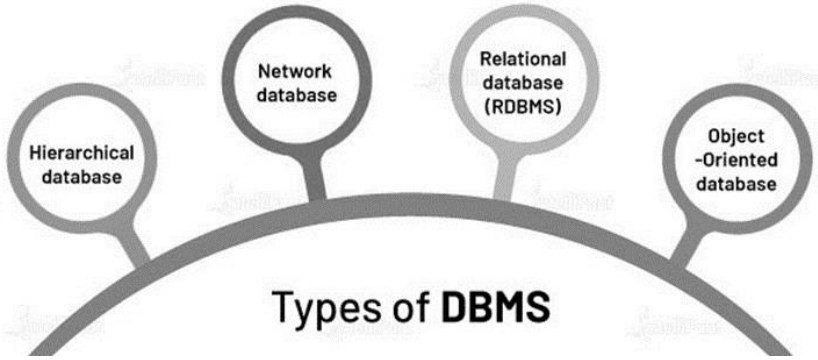
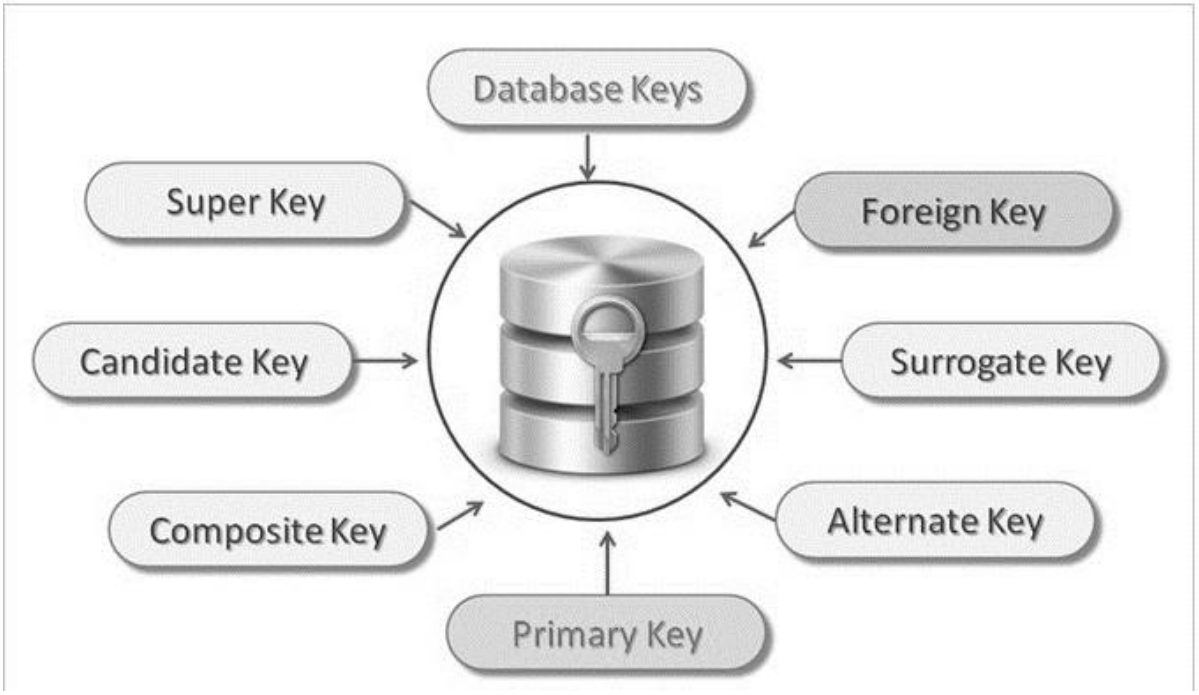
TurboTech Revision
Database I

DESIGNS AND DEVELOPS DATABASE SYSTEMS TO MANAGE DATA EFFICIENTLY AND EFFECTIVELY







Name _____

School _____

Student ID _____

Designs and develops database systems to manage data efficiently and effectively

Data vs Information

Data is a raw and unorganized fact that required to be processed to make it meaningful. Data is always interpreted, by a human or machine, to derive meaning. So, data is meaningless. Data contains numbers, statements, and characters in a raw form.

Information is a set of data which is processed in a meaningful way according to the given requirement. Information is processed, structured, or presented in a given context to make it meaningful and useful. Information assigns meaning and improves the reliability of the data. It helps to ensure undesirability and reduces uncertainty. So, when the data is transformed into information, it never has any useless details.

Difference between Structured, Semi-structured and Unstructured data

Big Data includes huge volume, high velocity, and extensible variety of data. These are 3 types: Structured data, Semi-structured data, and Unstructured data.

1. **Structured data –**
   Structured data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database SQL in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information.
   Example: Relational data.
2. **Unstructured data –**
   Unstructured data is a data which is not organized in a predefined manner or does not have a predefined data model; thus, it is not a good fit for a mainstream relational database. So for Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications.
   Example: Word, PDF, Text, Media logs.

Introduction to Databases

A collected information which is in an organized form for easier access, management, and various updating is known as a database. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

Application of DBMS

Sector	Use of DBMS
Banking	For customer information, account activities, payments, deposits, loans, etc.
Airlines	For reservations and schedule information.
Universities	For student information, course registrations, colleges and grades.
Telecommunication	It helps to keep call records, monthly bills, maintaining balances, etc.
Finance	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
Sales	Use for storing customer, product & sales information.
Manufacturing	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
HR Management	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Characteristics of Database Management System

- Provides security and removes redundancy
- Self-describing nature of a database system
- Insulation between programs and data abstraction
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing
- DBMS allows entities and relations among them to form tables.
- DBMS supports multi-user environment that allows users to access and manipulate data in parallel.

Advantages of DBMS

- DBMS offers a variety of techniques to store & retrieve data
- DBMS serves as an efficient handler to balance the needs of multiple applications using the same data
- Uniform administration procedures for data
- Application programmers never exposed to details of data representation and storage.
- A DBMS uses various powerful functions to store and retrieve data efficiently.
- Offers Data Integrity and Security
- The DBMS implies integrity constraints to get a high level of protection against prohibited access to data.
- A DBMS schedules concurrent access to the data in such a manner that only one user can access the same data at a time
- Reduced Application Development Time

Disadvantage of DBMS

DBMS may offer plenty of advantages but, it has certain flaws-

- Cost of Hardware and Software of a DBMS is quite high which increases the budget of your organization.
- Most database management systems are often complex systems, so the training for users to use the DBMS is required.
- In some organizations, all data is integrated into a single database which can be damaged because of electric failure or database is corrupted on the storage media
- Use of the same program at a time by many users sometimes lead to the loss of some data.
- DBMS can't perform sophisticated calculations

Types of DBMS

Five Types of DBMS systems are:

- Hierarchical database model
- Flat File database model
- Network database model
- Relational database model
- Object-Oriented database model

Hierarchical DBMS

In a Hierarchical database, model data is organized in a tree-like structure. Data is Stored Hierarchically (top down or bottom up) format. Data is represented using a parent-child relationship. In Hierarchical DBMS parent may have many children, but children have only one parent.

This model structure allows the one-to-one and a one-to-many relationship between two various types of data. This structure is very helpful in describing many relationships in the real world; table of contents, any nested and sorted information.

The hierarchical structure is used as the physical order of records in storage. One can access the records by navigating down through the data structure using pointers which are combined with sequential accessing. Therefore, the hierarchical structure is not suitable for certain database operations when a full path is not also included for each record.

Data in this type of database is structured hierarchically and is typically developed as an inverted tree. The "root" in the structure is a single table in the database and other tables act as the branches flowing from the root. The diagram below shows a typical hierarchical database structure.

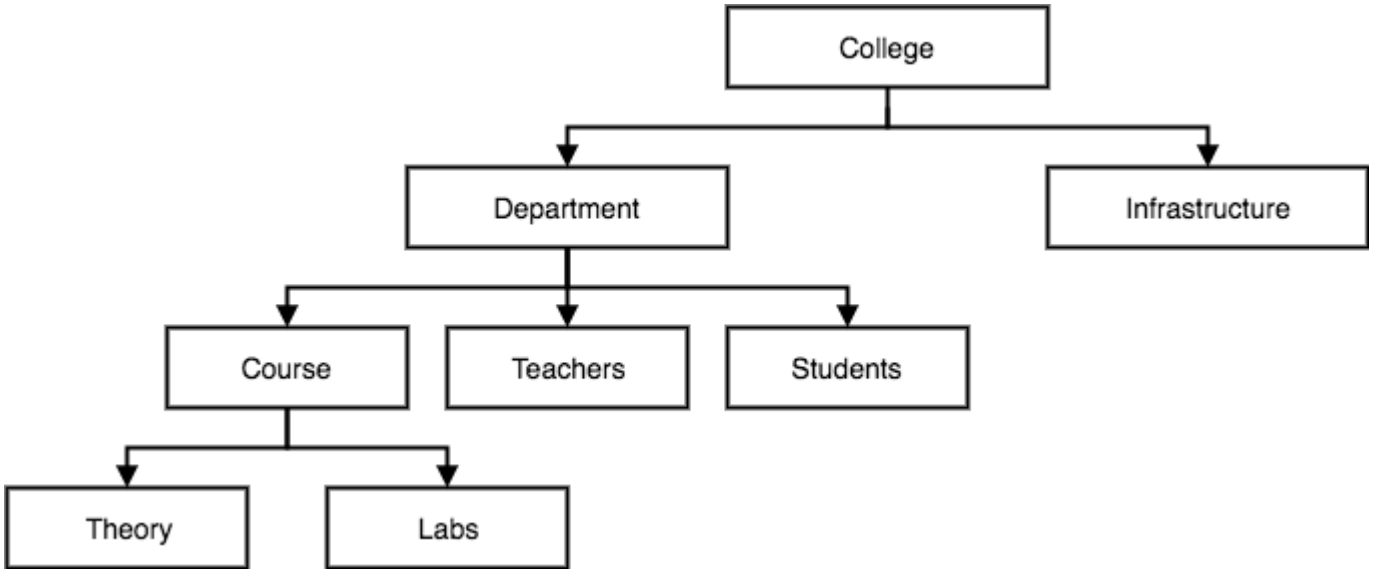


Figure 1- HM

Flat File database model

THE FLAT (OR TABLE) MODEL is most conventional and simple data model, which consists of a single, two-dimensional array of data elements, in which all members of a given column represents similar values, and all members of a row represent relations to one another. For example, columns are used for name and password that are a part of a system security database. Each row contains specific password associated with a specific user. Columns of the table contain a type which defines a character data, date or time information, integers, or floating-point numbers.

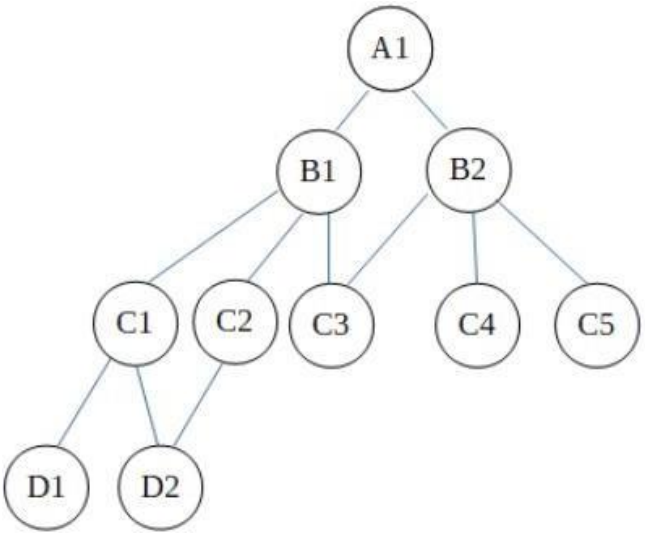
a flat file database is a database which stored an ordinary unstructured file also known as a "flat file". This flat model suits best for the small, simple databases. As the size of data grows, the memory access becomes difficult and requires more sophisticated databases. A list of names, contact number, address, a city which is written by hand is a flat file database, If the same information gets recorded on a spreadsheet then it can be used online for improved search capabilities. One can also transfer the data by using a flat file database model.

Network Model

The network database model allows each child to have multiple parents. It helps you to address the need to model more complex relationships like as the orders/parts many-to-many relationship. In this model, entities are organized in a graph which can be accessed through several paths.

The network database model was created to solve the shortcomings of the hierarchical database model. In this type of model, a child can be linked to multiple parents, a feature that was not supported by the hierarchical data model. The parent nodes are known as owners and the child nodes are called members.

The network data model can be represented as −



Advantages of Network Model

The network model can support many to many relationships as seen in the diagram. D2 and C3 each have multiple masters. The masters for D2 are C1 and C2 while for C3 are B1 and B2. In this way, the network data model can handle many to many relationships where the hierarchical data model didn't.

Disadvantages of Network Model

There are some disadvantages in the network model even though it is an improvement over the hierarchical model. These are −

- The network model is much more complicated than the Hierarchical model. As such, it is difficult to handle and maintain.
- Although the Network model is more flexible than the Hierarchical model, it still has flexibility problems. Not all relations can handle by assigning them in the form of owners and members.
- The structure of the Network Model is quite complicated and so the programmer has to understand it well in order to implement or modify it.

Object-Relational Model

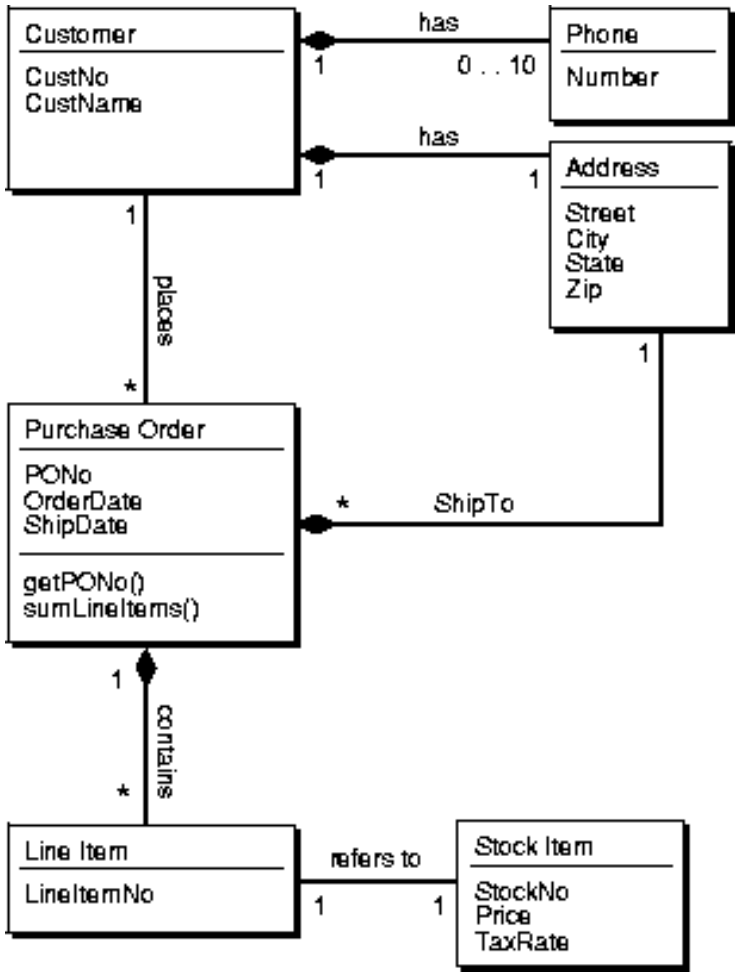
An Object relational model is a combination of Object oriented database model and a Relational database model. So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.

The benefits provided by the object-relational model include:

- Support for many geometry types, including arcs, circles, compound polygons, compound line strings, and optimized rectangles
- Ease of use in creating and maintaining indexes and in performing spatial queries
- Index maintenance by the Oracle database
- Geometries modeled in a single column
- Optimal performance

The ORDBMS approach has the obvious disadvantages of complexity and associated increased costs. Further, there are the proponents of the relational approach that believe the· essential simplicity' and purity of the. relational model is lost with these types of extension.

ORDBMS vendors are attempting to portray object models as extensions to the relational model with some additional complexities. This potentially misses the point of object orientation, highlighting the large semantic gap between these two technologies. Object applications are simply not as data-centric as relational-based ones.



Install Microsoft SQL Server Management Studio

Step 1

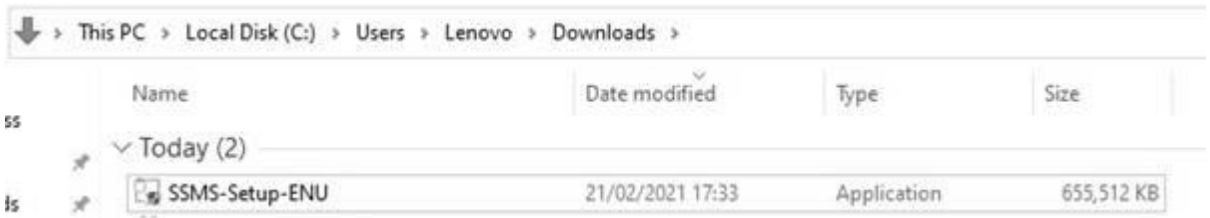
To install SQL Server Management Studio, you need to download it from this link.

Step 2

Click the "Download" button for downloading the SSMS executable file on the downloaded path.

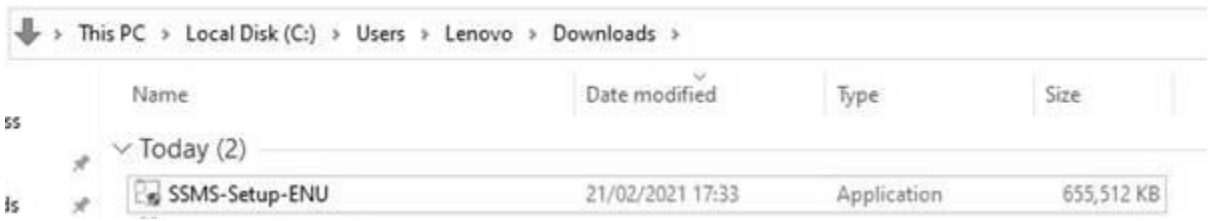
Step 3

Open your system download path and find the .exe file. SSMS-Setup-ENU.exe



Step 4

Double-click the .exe file SSMS-Setup-ENU.exe to start installing. The installation process of SMSS is straightforward and you just need to follow the screen sequence.



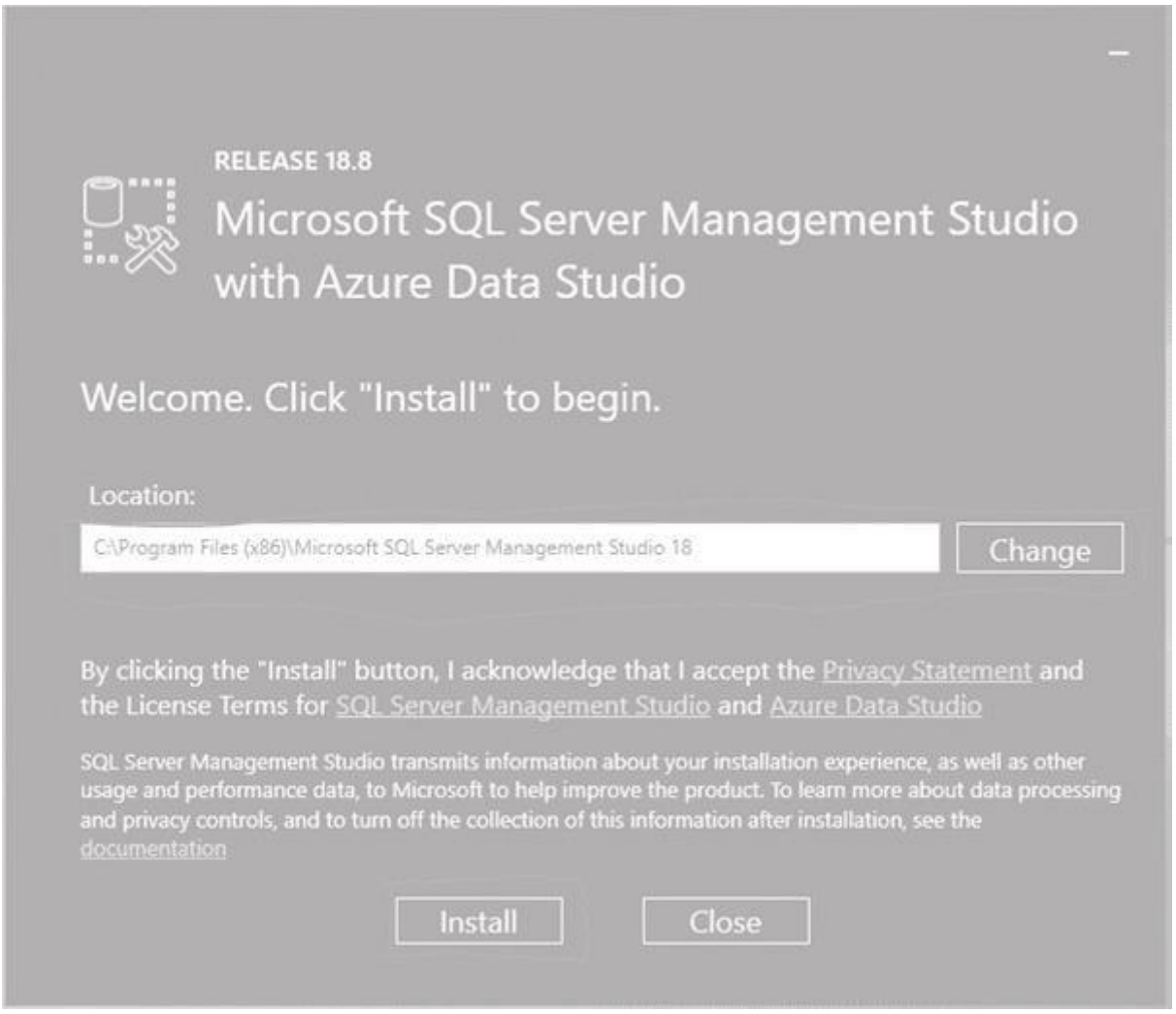
Step 5

After double-clicking, the system will ask the permission: “Do you want to allow the following to make a change to this computer? Click yes to continue installing the SQL Server Management Studio, or...

Click “Yes” on any security prompt.

Step 6

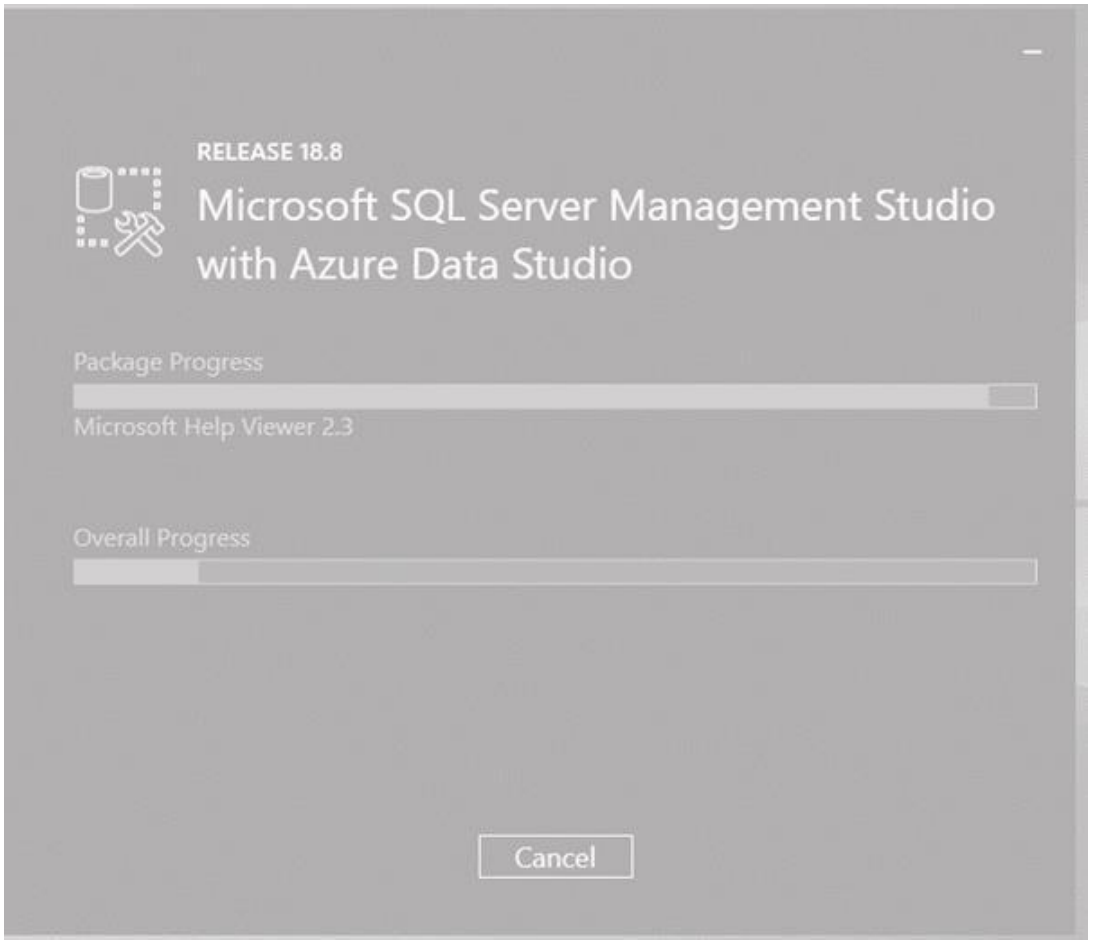
The installation window will be open after giving permission to install. Click the Install button.



Step 7

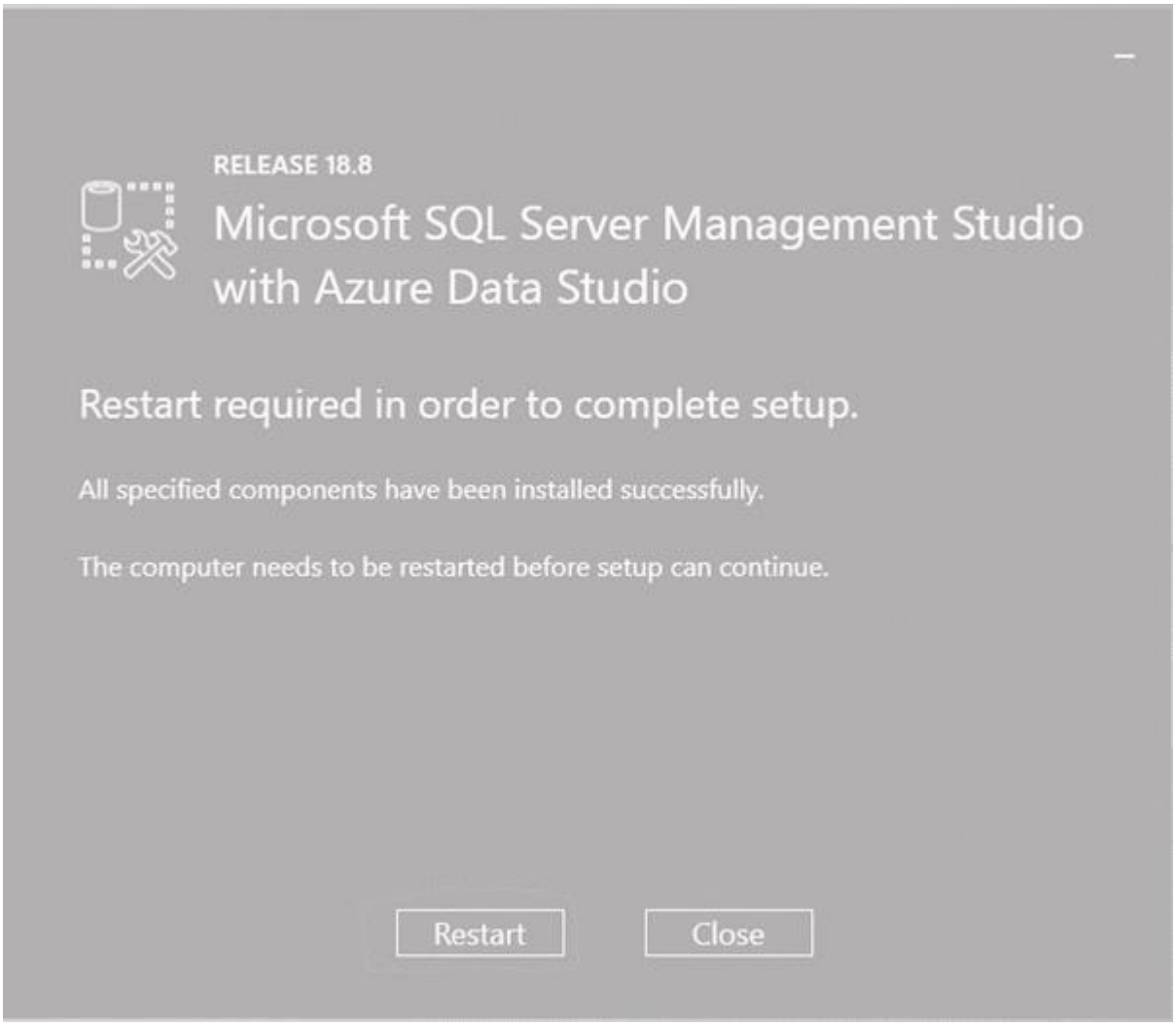
After loading packages progress bar will be shown. One is Package Progress and Overall Progress. Wait for few minutes while the installer sets up the software.


RELEASE 18.8
Microsoft SQL Server Management Studio
with Azure Data Studio
Loading packages. Please wait...
Cancel


RELEASE 18.8
Microsoft SQL Server Management Studio
with Azure Data Studio
Package Progress
Microsoft Help Viewer 2.3
Overall Progress
Cancel

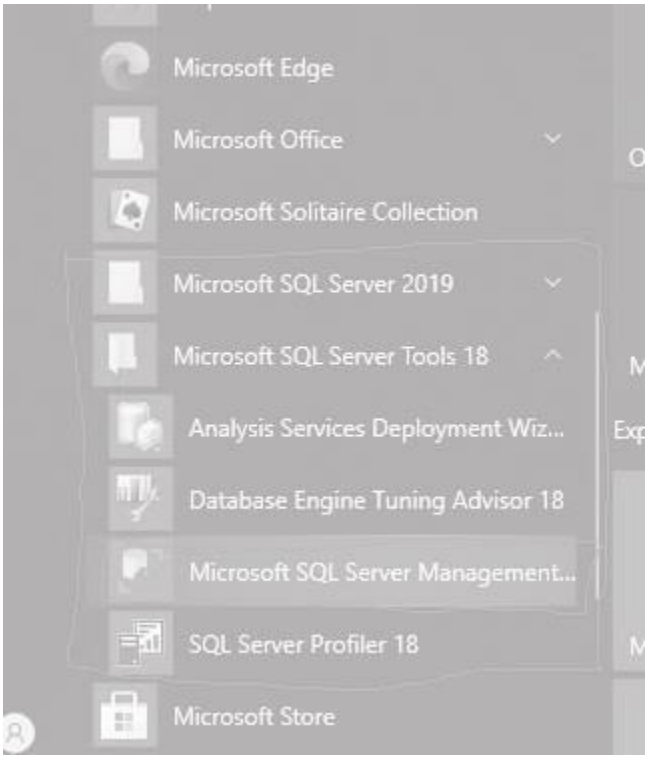
Step 8

Installation completed. After completing the installation restart your computer for a complete setup.



Step 9

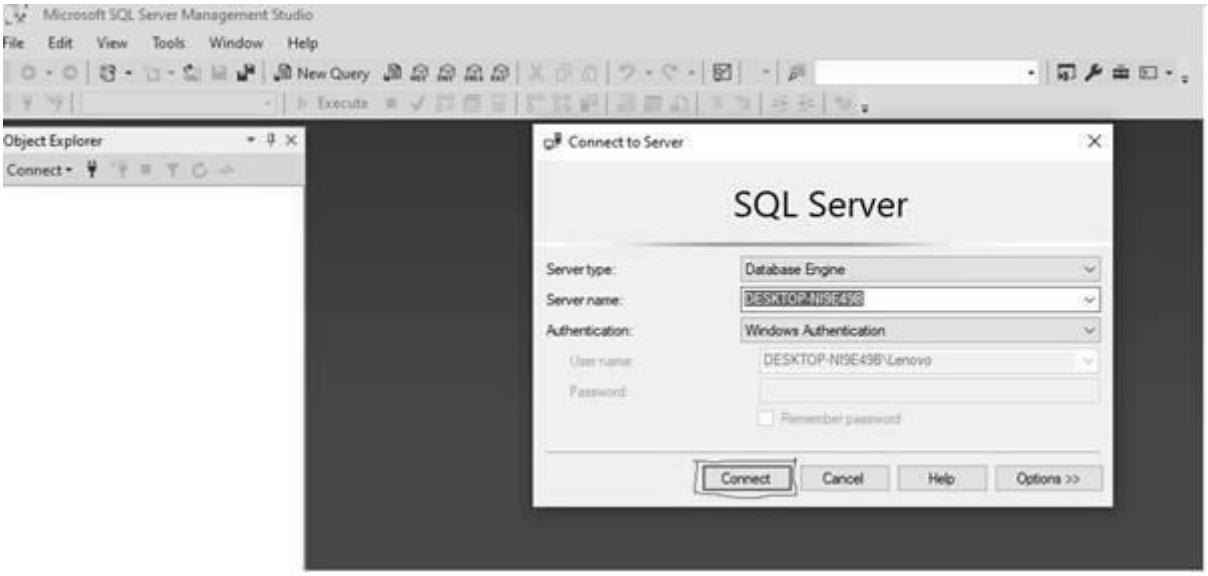
Go to all programs in your systems, and we can see two folders one is Microsoft SQL Server and another one is Microsoft SQL Server Tool 2018. Under Microsoft SQL Server Tools 18 you can see the Microsoft SQL Server Management Studio 18.



Step 10

Double-click on SQL Server Management Studio 18 and it will open looks like the below screenshot. The first time opening it will take a few minutes.

After opening SQL Server Management Studio 2018, we can see that it looks like the below screenshot.



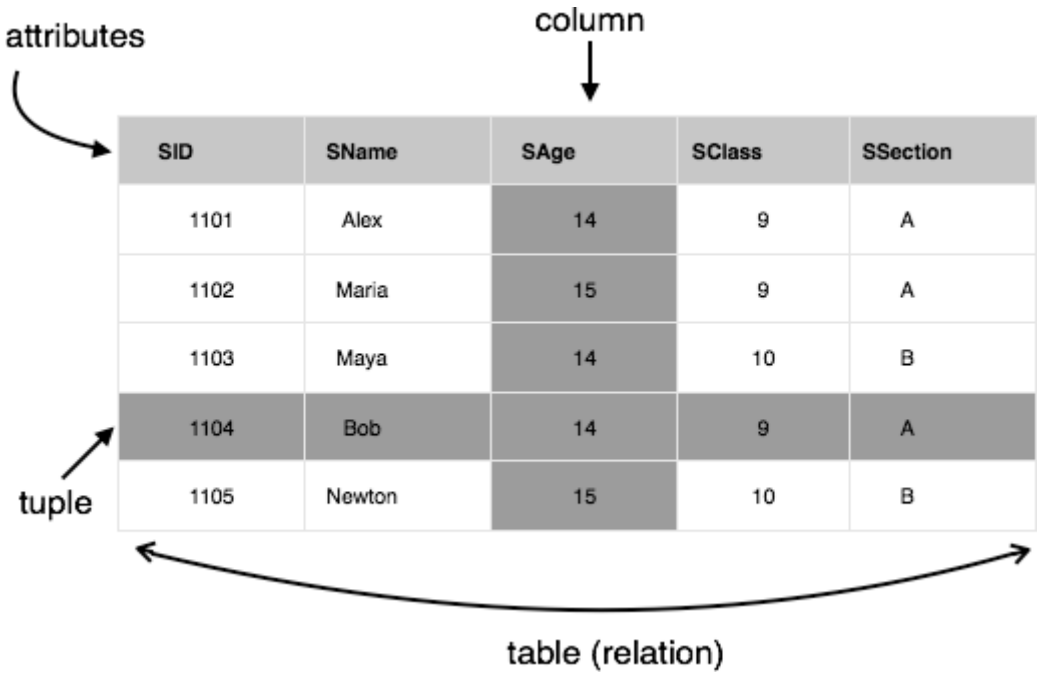
Now we can connect to the server and use SQL Server Management Studio.

Relational model

Relational DBMS is the most widely used DBMS model because it is one of the easiest. This model is based on normalizing data in the rows and columns of the tables. Relational model stored in fixed structures and manipulated using SQL.

A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

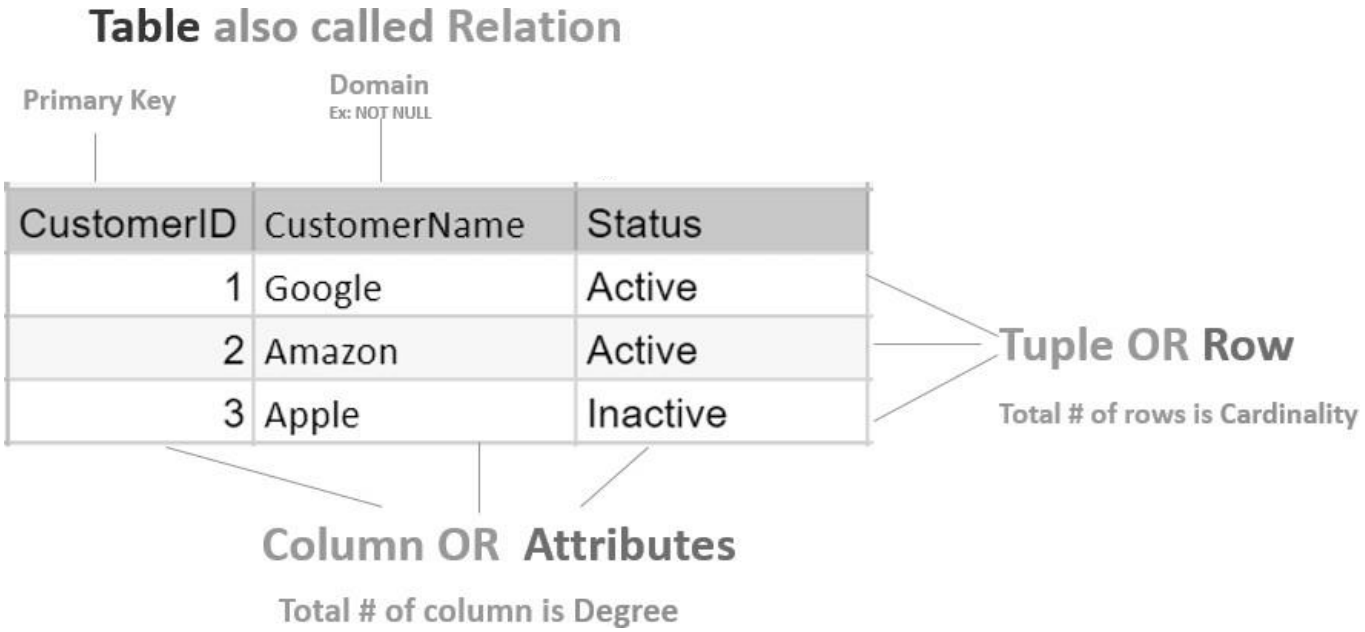


The main highlights of this model are −

- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

Relational Model Concepts

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, NAME,etc.
2. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
3. **Tuple** – It is nothing but a single row of a table, which contains a single record.
4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.
5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.
6. **Cardinality:** Total number of rows present in the Table.
7. **Column:** The column represents the set of values for a specific attribute.
8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain



Relational Integrity constraints

Relational Integrity constraints is referred to conditions which must be present for a valid relation. These integrity constraints are derived from the rules in the mini-world that the database represents.

There are many types of integrity constraints. Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain constraints
2. Key constraints
3. Referential integrity constraints

Domain Constraints

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

Example:

```
Create DOMAIN CustomerName
CHECK (value not NULL)
```

The example shown demonstrates creating a domain constraint such that CustomerName is not NULL

Key constraints

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

Example:

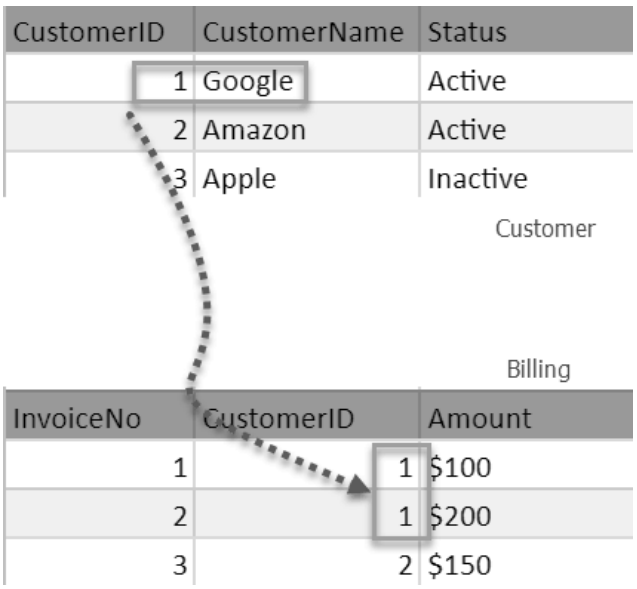
In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Referential integrity constraints

Referential integrity constraints is based on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

Example:



In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So, we know CustomerName=Google has billing amount $300