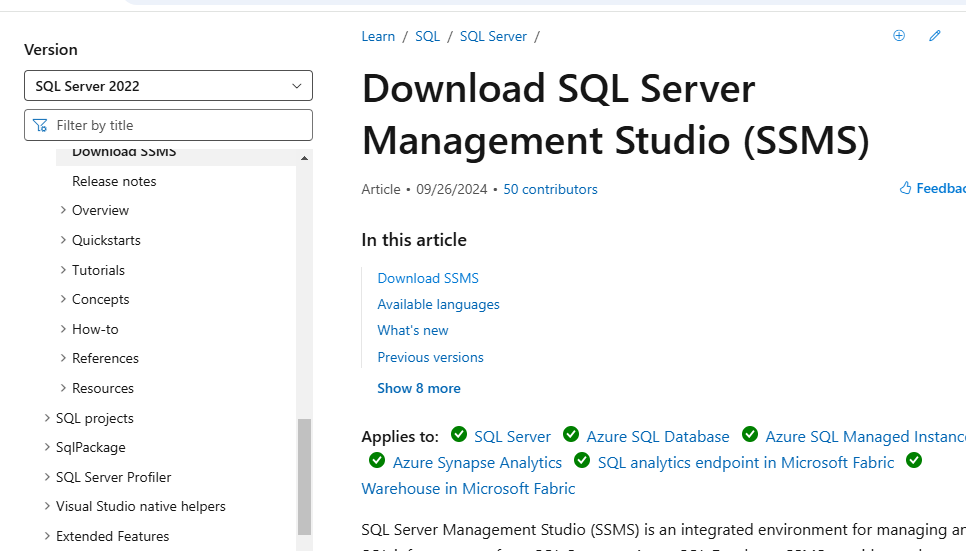
**SQL Server**

* Installation
* Got to the site to download [SQL Server](https://www.microsoft.com/en-in/sql-server/sql-server-downloads). Download developer edition.
* Open the downloaded exe extension and setup SQL. Select *BASIC* installation type.
* Need to restart the system to complete the setup.
* Go to the site to download [SSMS](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16&redirectedfrom=MSDN#download-ssms) setup.
* Click on download SSMS.



* Open the downloaded SSMS exe extension and complete SQL setup.
* Open the SQL app and set up the connection.

A screenshot of a computer

Description automatically generated

**Database Design:**

**(**[**https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5**](https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5) **)**

* Database terms:
  + Tables: List of rows and columns.
  + Record: Each row is called a record.
  + Field: Each column is called a field.
* What is a good database?
  + Principles of database design process:
    - Duplicate information (also called redundant data) is bad, because it wastes space and increases the likelihood of errors and inconsistencies.
    - The correctness and completeness of information is important. If the database contains incorrect information, any reports that pull information from the database will also contain incorrect information. As a result, any decisions made that are based on those reports will then be misinformed.

**The Design Process:** The design process consists of following steps:

* **Determine the purpose of the database.**
  + Identify the purpose of the database – its purpose, how you expect to use it, and who will use it.
* **Find and organize the information required**.
  + Identify the items (information) that need to be stored in the table.
  + Each item will be column of the table.
  + A key point to remember is that you should break each piece of information into its smallest useful parts.
  + Try to get the information from the table as required, so that you can add an extra column if any information is pending.
* **Divide the information into tables.**
  + To divide the information into tables, choose the major entities, or subjects.
  + When you design your database, always try to record each fact just once.
  + If there is any data that is repeatedly stored in the table, the create a separate table and store it only once to avoid duplication of data.
  + If we want to change the data, we can just change it at one place instead of changing many records and avoid chance of missing to change at some records.
  + Once you have chosen the subject that is represented by a table, columns in that table should store facts only about the subject.
* **Turn information into columns.**
  + To determine the columns in a table, decide what information you need to track about the subject recorded in the table.
  + Tips for determining the columns:
    - Don’t include calculated data.
    - Store information in its smallest logical parts.
* **Specify primary key.** 
  + The primary key is a column that is used to uniquely identify each row.
  + Each table should include a column or set of columns that uniquely identifies each row stored in the table.
  + Can’t have duplicate values in the primary key column.
  + Primary key can’t be null.
  + Always choose a primary key whose value will not change.
    - Using a primary key that will not change reduces the chance that the primary key might become out of sync with other tables that reference it.
  + A column set to the AutoNumber data type often makes a good primary key.
  + When a primary key employs more than one column, it is also called a ***composite key***.
* **Setup the table relationships.**
  + One-to-many relationship: (e.g.: products (many) and suppliers(one))
    - To represent a one-to-many relationship in your database design, take the primary key on the "one" side of the relationship and add it as an additional column or columns to the table on the "many" side of the relationship.
    - That column become foreign key in “many” side of the relationship table.
    - A foreign key is another table’s primary key.
  + Many-to-many relationship: (e.g.: products (many) and orders (many))
    - To detect many-to-many relationships between your tables, it is important to consider both sides of the relationship.
    - Create a third table, often called a ***junction table***, that breaks down the many-to-many relationship into two one-to-many relationships.
    - Insert the primary key from each of the two tables into the third table. As a result, the third table records each occurrence or instance of the relationship.
  + One-to-one relationship: (e.g.: products table and products supplement table with some empty fields)
  + When a one-to-one or one-to-many relationship exists, the tables involved need to share a common column or columns. When a many-to-many relationship exists, a third table is needed to represent the relationship.
* Refine the design.
* Apply the normalization rules.
  + Use these rules to see if the tables are structured correctly.
  + The process of applying the rules to the database design is called normalizing the database, or just normalization.
  + First normal form:
    - First normal form states that at every row and column intersection in the table there, exists a single value, and never a list of values.
  + Second normal form:
    - Second normal form requires that each non-key column be fully dependent on the entire primary key, not on just part of the key.
    - This rule applies when you have a primary key that consists of more than one column.
    - E.g. table having two primary keys as product id and order id. This table have a column with product name which is dependent only on product id and not on order id. The product name is not dependent on the entire primary key. So, the product name column has to be placed in products table.
  + Third normal form:
    - Third normal form requires that not only every non-key column be dependent on the entire primary key, but that non-key columns be independent of each other.