**Microsoft SQL**

**Connecting to SQL Server**

**Launch AppsAnywhere**

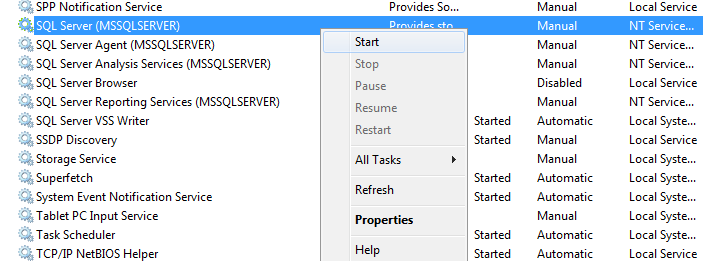
<https://appsanywhere.bcit.ca/>

Select Computing Desktop

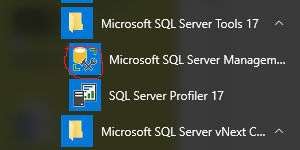
Use Citrix light version

Once the desktop is showing

1)      Open the Windows service by searching “Service”. Check if SQLServer service is running, if not manually start it.



3)      Start  Microsoft SQL Server xxxx  SQL Server Management Studio  
Open SQL Server Management Studio



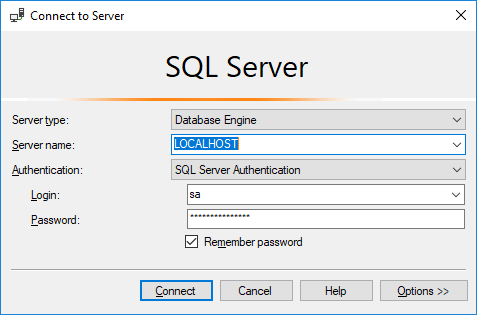
 4)     Change the authentication type to SQL Server Authentication

Credentials

Server Type = database engine

**Username = ‘sa’**

**Password = ‘bcitsql’**



5.

**Create database Sale**

Run the script

USE [Sales]

GO

CREATE TABLE [dbo].[Product]

(

[ProductId] [uniqueidentifier] DEFAULT NEWID() NOT NULL,

[ProductName] [nchar](50) NULL,

[ProductDescription] [nchar](3000) NULL,

[ProductPrice] MONEY NULL

) ON [PRIMARY]

GO

Then, the script to keeps the database a little busy:

USE [Sales]

GO

--WHILE (1=1)

-- BEGIN

TRUNCATE TABLE [Sales].[dbo].[Product]

DECLARE @Record INT

SET @Record=1

WHILE @Record<=10

BEGIN

INSERT INTO [Sales].[dbo].[Product]

([ProductName] ,[ProductDescription],[ProductPrice])

VALUES ('Product ' + STR(@Record), 'Description ' + STR(@Record), @Record\*100/3)

SET @RECORD = @RECORD+1

END

SELECT COUNT(ProductID) as RecordsCreated FROM [Sales].[dbo].[Product]

--END

select \* from product

## Introduction to Transact SQL

To perform SQL queries against a Microsoft SQL database, open Microsoft SQL Server Management Studio  and navigate to the database you are using.  Highlight the database, right click and select New Query or click the New Query button on the left side of the application's toolbar.  In order to execute a query, press F5 or click the Execute button in the toolbar.

Keep in mind many database systems (Microsoft SQL Server, Oracle SQL, MySQL, etc) each have a slightly different syntax for queries.  The language used by MS SQL is called Transact-SQL (or T-SQL for short).  This article is written specifically with MS SQL / T-SQL in mind.

**Getting started: SELECT and FROM**

 To pull information from a database using a SQL query one must first specify the column(s) and tables(s) which contain the desired information. This is done with the SELECT statement.

On this page

* Introduction
* Getting started: SELECT and FROM
* Linking tables: JOIN
* Filtering results: WHERE
* Examples
* Further reading

Desired columns are specified immediately after the SELECT command. Multiple values are separated by commas. Field names may be dragged and dropped from the left Object Explorer. Entering a star (\*) will return all columns. Tables to be queried are specified following the FROM command in the second part of the statement.

Here are some examples:

-- Return all records from the 'Customer' table

SELECT \* FROM Customer

-- Return the 'name' and 'address' columns from the 'Customer' table

SELECT name, address FROM Customer**Linking tables: JOIN**

In many cases, information needs to be pulled from more than one table that need to be joined together on certain conditions. In selection the columns need to be prefixed by the table they belong to and a period (eg: Customer.name and Product.type).

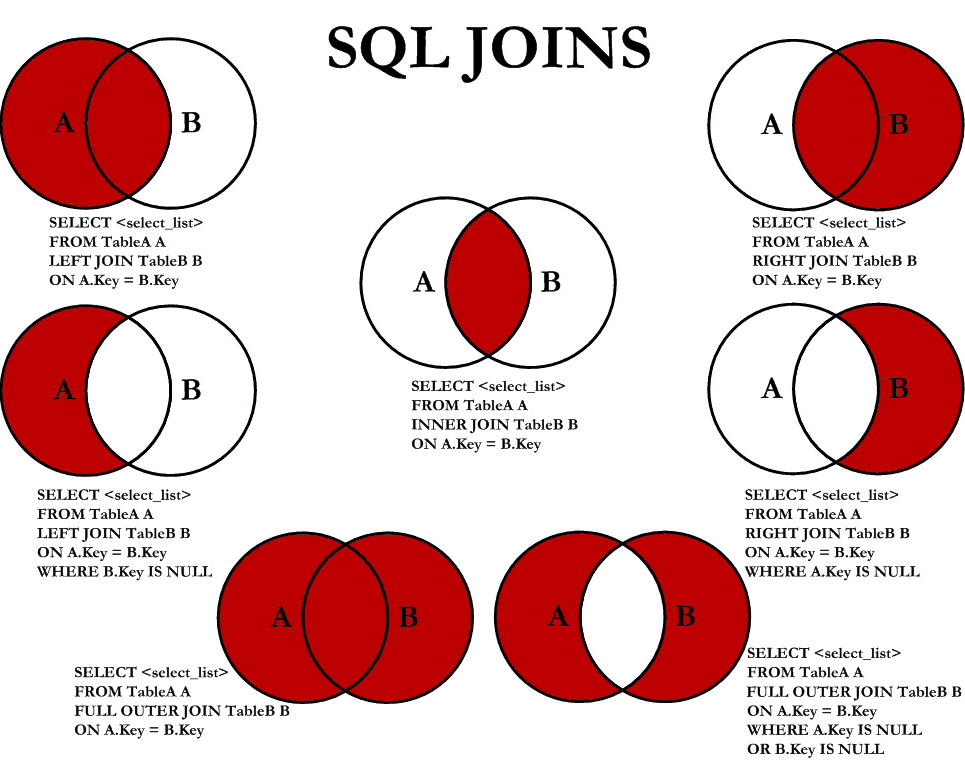
The following example selects all the products for all customers with products:

SELECT Customer.name, Customer.address, Product.type -- see prefix

FROM Customer

INNER JOIN Product ON Customer.id = Product.customerid -- Joins 'Customer' & 'Product' tables

There are various ways to join tables. INNER JOIN is most often used (if you just type JOIN with no prefix, it defaults to INNER JOIN. Specifying the INNER prefix is preferred because it makes the query clearer to read particularly if there are several types of joins being used). The adjacent diagram is a helpful representation of the possible join types:



Filtering results: WHERE

The WHERE clause enables you to narrow your search by specifying additional parameters. For example, you could narrow your query to return:

* Customers with the Address in Burnaby
* Customers with no address
* All Active products

The syntax for the WHERE clause is:

WHERE [column\_name] [operator] 'some\_value'

Some common operators are listed below. If more than one condition is required, the AND & OR operators may be used as well. These operators may be combined in parenthesis to form complex expressions.

=              Equal  
>              Greater than  
>              Less than  
>=           Greater than or equal  
<=           Less than or equal  
<>           Not equal to  
AND        True if both expressions are true  
OR           True if either expression is true   
IN            True if the operand is equal to one of a list of expressions: (‘one’, ‘two’, ‘three’)  
NOT IN    Similar to above, but true if the operand is not equal

There are some more advanced operators which may be useful as well such as LIKE or IN. With LIKE you can use the '%' wildcard character to search for patterns. With IN you can search for columns which may have multiple values (similar to several OR statements).

SELECT Customer.name FROM Customer

    INNER JOIN Product ON Customer.Id = Product.CustomerId

WHERE

    Customer.Address like ‘%Burn%’

    AND Product.Status  = 'active'

Examples

-- Returns the first active customer from the Langley

SELECT TOP 1 Name FROM Customer

WHERE Address='Langley' AND Status='active'

ORDER BY Name asc

-- Return all records in the 'Customer' table

SELECT \* FROM Customer

-- Return all records in the 'Customer' table, with Birth Date and SIN displayed first

SELECT BirthDate, sin, \* FROM Customer

-- Return first 50 records in the 'Customer' table

SELECT TOP 50 \* FROM Customer

-- Return all records in the 'Customer' table where the name is numeric

SELECT \* FROM Customer WHERE name not like ‘%[^0-9]%’

-- Return the # of records in the ‘Customer’ table where the title=Mr

SELECT COUNT(\*) FROM Customer WHERE title = 'Mr'

-- Same as above, but with a column heading specified

SELECT COUNT(\*) AS 'Total' FROM Customer WHERE title = 'Mr'

--  Return the total number of Customers in each city

SELECT City, COUNT(\*) AS 'Total'

FROM Customer

WHERE Status='active'

GROUP BY City

ORDER BY City

-- Example with SUM

SELECT Name, COUNT(Name) AS 'Count', SUM(Amount) AS 'Total'

FROM Product

WHERE Amount > 0 AND InvoiceDate > '2016-01-01'

GROUP BY Name

INSTALL MSSQL and SSMS

Youtube

https://www.youtube.com/watch?v=QsXWszvjMBM

===========================================================================

Navigate to

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Choose SQL Server 2019 Developer -> Download now

Graphical user interface, text, application

Description automatically generated

Choose Basic

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Complete the installation

Graphical user interface, text, application

Description automatically generated

Server=localhost;Database=master;Trusted\_Connection=True;

Click Download SSMS

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?redirectedfrom=MSDN&view=sql-server-ver16>

Graphical user interface, text, application, email

Description automatically generated

Save the download