SQL

Click + on Databases, choose (eg) Northwind

Seattle Cycle Hire – reloading –

DVD on LOD – select Seattle\_Cycle\_Hire

Unzip the zip file and save

ON SSML Databases, R click select Import Data Tier Application

Click all the next buttons and the DBase will appear

+ to show all the different tables

R Click, new Query to create a new ‘report’

To view the structure of the DBase – Open the DBase and R click – select new DBase diagram

Select all the tables and then ‘Add’ and then ‘Close’ when the process has stopped. This can then be saved in the program to be able to view the relationships (keys) between all the tables

The order of writing queries is important

|  |
| --- |
| SELECT |
| FROM |
| JOIN |
| (ON) |
| WHERE |
| GROUP BY |
| HAVING |
| ORDER BY |

We also have functions (Count, sum, floor, ceiling, month, year, left, right, average) and other things which need to go in the correct order. Examples to follow with the relevant queries in pink.

select avg (Freight) from Orders where EmployeeID=6

select sum (Quantity) as 'Total Quantity' from [Order Details]

Always type SELECT \* FROM Table Name first as this shows all the columns. Then you can select the ones you need

It is possible to add / multiply etc any columns to create a new one and rename it

Eg Units in stock + units on order as ‘Future Stock’

Tip - Put each query on a new line. When listing columns to be selected put the , (separator) at the start of the next line so it is easier to see in case of an error

Tip – Put any comments or notes after – so they appear in green and do not affect the query. If there is a lot you can use /\* at the start and \*/ at the end.

The symbol <> or != mean Not equal to

Dates are written as default – YYYYMMDD. They can be changed using the Convert function (eg 101 which amends to DD/MM/YYYY)

A WHERE example – Or Where

Where CategoryID=7 Unitprice <=40

Or CategoryID=8 and Unitprice>=35

And UnitPrice>30

Or - and o.ShipCountry like '%u%' - which returns all countries where there is a letter U in the name

Null will be returned where a field has no info so can be used to tidy a DBase using the query Where ‘column’ = NULL

Limiting results – use the query Select Top x where x is the top number of responses required or Select Top 10 percent. Can also add with ties to get first equal etc

**JOINS**

SELECT

e.EmployeeID -- the e. denotes that we have selected from table e and column EmployeeID

,e.LastName

,e.FirstName

,o.Freight -- the o. denotes we need the infor from the table orders

,o.ShipCountry

,e.FirstName+' '+e.LastName as 'Full Name' -- creates a new column with the whole name rather than two fields. Note the need for two + signs and the space between the apostrophes

from Employees as e -- this is where I have told SQL that the table Employees is now to be known as o (makes for less typing)

join Orders as o --we have to tell SQL which tables to join and the tables must have a matching key (primary or foreign) ORDERS

on e.EmployeeID=o.EmployeeID -- and Employees with EmployeeID as the key

where o.Freight >=100 -- tells SQL some Parameters to narrow down the responses

and o.Freight <= 800 -- as above to show a maximum and a minimum

and o.ShipCountry like '%u%' -- this shows all countries with a letter U in the name. 'u%' would show those with U at the start, '%u' - at the end

ORDER BY o.Freight desc

Inner join (default) shows data in the subset of the two columns. Outer join shows the data in all the of the column listed first

Some tables have a self join

Eg when creating a table which shows an employee’s manager when both come from the same table

select

e.FirstName+' '+e.LastName as 'Full Name' -- creates new column to replace First and Last Names

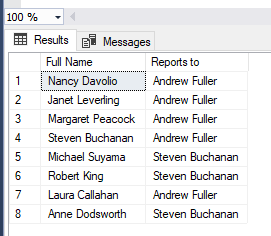
, m.FirstName+' '+m.LastName AS 'Reports to' -- same, creating another new coumn but in a new table 'm'

from

employees as e -- already in existence

join employees as m -- the 'new' table

on e.reportsTo=m.employeeID -- the link

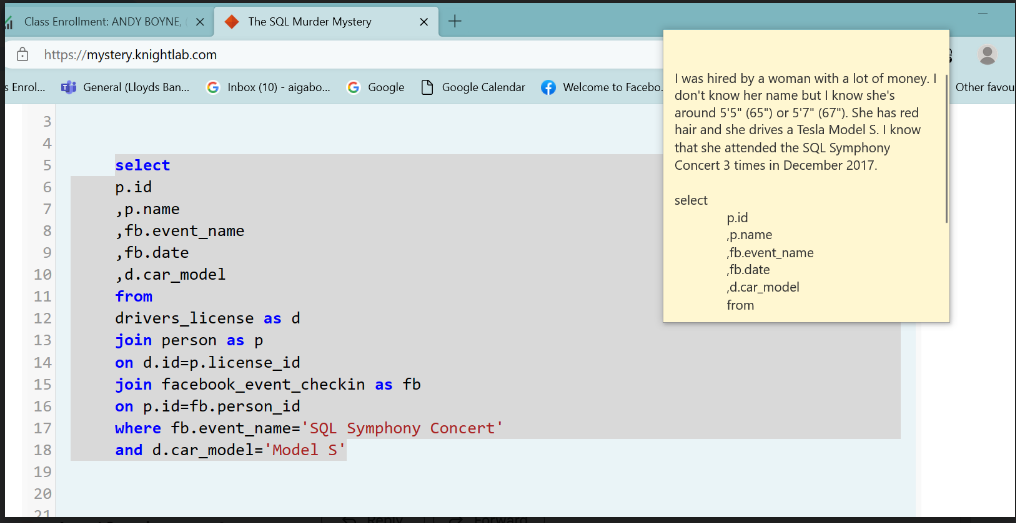


Union joins rows not columns

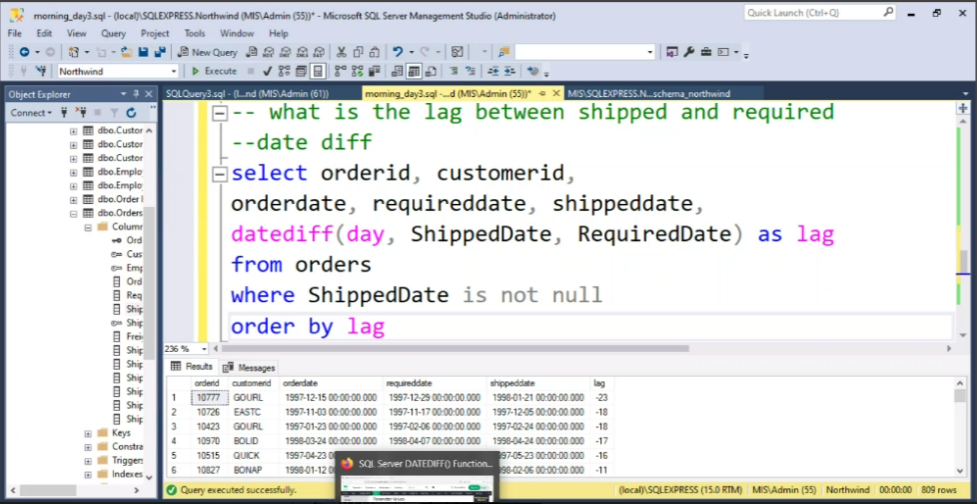
Tip –

* First thing to do is to find Primary or Foreign keys which are common in both tables to use as the joins
* This can be done by creating a schema diagram to see the keys and the links
* What kind of join do we need?
* Left Outer / Right Outer, Inner or Full join, self join, Union
* Decide which is the main table (usually the one with most data) and make it the (Left) first one

Eg below



This is an example of some functions in use



This one shows how to calculate the difference between two dates and return the answer in days – a bit like the Doc Ordering SP where the difference is between today and the date created.

Other functions – see W3Schools website (<https://www.w3schools.com/sql/default.asp>)

Example of a multi question table –

Questions –

1. From which station (ID and name) were most cycle hire trips taken overall?
2. How many trips were taken from this station?
3. How many unique bikes were borrowed from this station?
4. What was the average trip duration (rounded to the nearest number of minutes) from this station?

select

--s.station\_id --not required on the select section

t.from\_station\_name -- do not need to select columns and then do the count, sum or avg functions. Enough to write the function

,count(t.trip\_id) as 'Total Trips'

,count (distinct t.bikeid) as 'Bikes taken from station'

,round(avg(t.tripduration) /60, 0) as 'Average Duration in whole minutes'

from Seattle\_cycles\_station as s

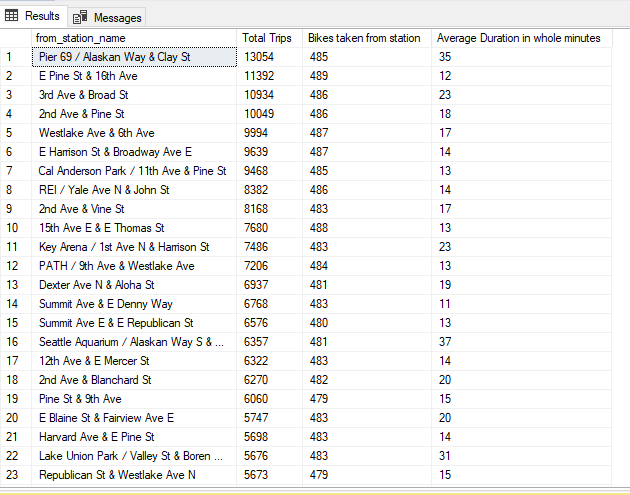
join Seattle\_cycles\_trip as t

on s.station\_id=t.from\_station\_id

group by t.from\_station\_id, t.from\_station\_name – don’t need a where... because we can see the most popular station in the table

order by 'Total Trips' desc

This gives the table –



Answers –

1. From which station (ID and name) were most cycle hire trips taken overall? Pier 69…
2. How many trips were taken from this station? 13,054
3. How many unique bikes were borrowed from this station? 485
4. What was the average trip duration (rounded to the nearest number of minutes) from this station? 35