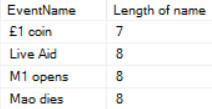
***\*\* To Submit all this Exercise Create a Single word file and copy all your scripts along with screenshot of the OutPut. And Share it on*** [***nehalole1115@gmail.com***](mailto:nehalole1115@gmail.com)

***\*\* to start with your exercise first execute all queries from below link in your WORLD EVENT DATA BASE*** [***https://github.com/PurpleGrad/2203/blob/main/Generate%20list%20of%20countries.sql***](https://github.com/PurpleGrad/2203/blob/main/Generate%20list%20of%20countries.sql)

**String**

1. To do this exercise, you'll need to know that you can use the **LEN**function to work out how many characters there are in a bit of text.

Use this information to create a query listing out **each event** with the **length of its name**, with the **"shortest event" first**:



*The first few events in the list.*

Save this query as **Shortest events**, then close it down.

1. The **tblContinent**table lists out the world's continents:



Add up to 3 columns to **show Lower case, Upper case and length of Name of Continent**

1. If you've reached this point, you're probably ready for a challenge.  See if you can write a query to list out all of the non-boring events:



*The first few of the 28 events you should see.*

*As the above query shows, a boring event is one which doesn't begin and end with the same letter, and which doesn't begin and end with a vowel!*

If you get this working, you could always show that there aren't any ultra-interesting events (ones which begin and end in a vowel).

Save this query as **Boring events**, then close it down.

**Date Calculations:**

1. First create a query showing events which took place in 1978

This should give you something like this:



Amend your query so that it shows the event date neatly divided:

* Show **Months** for every event in separate column
* Show **Date** for every event in separate column

1. Create a query to show the day of the week and also the day number on which each event occurred:



*In no particular order, this is what the output of your query should look like.*

*You can use the****DATENAME****function to get the day of the week, and the****DAY****or the****DATEPART****functions to get the day number.*

Use this to show:

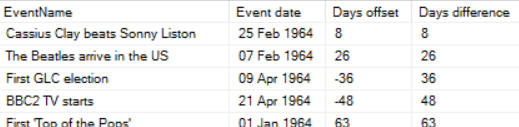
* That mercifully there weren't any events on Friday the 13th;
* That there was one event on Thursday 12th (the day before); and
* That there were two events on Saturday the 14th (the day after).

Save this query as **Friday the 13th**, then close it down.

1. The idea behind this exercise is to see what was happening in the world around the time when you were born (but you can use any reference date).  First create a query to show the number of days which have elapsed for any event since your birthday:



The **ABS**function returns the absolute value of a number (for example,**ABS(42)** and **ABS(-42)** both equal 42).  Use this to list the events in order of closeness to your birthday:



*For the same completely imaginary trainer, these are the events that they would see.*

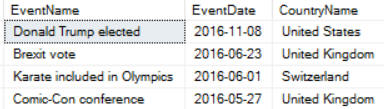
*You need to take the absolute value because the closest event to your birthday may be just before it (in which case the offset days will be negative) or just after it (in which case the offset days will be positive).*

Was yours an auspicious year?  Save this query as **Birth pool**, then close it down.

**Subqueries**

1. Create a query which lists out all of the events in the **tblEvent**table which happened after the last one for country 21 (**International**) took place.  Here's the gist of what you need to do:

You should get these 4 events:

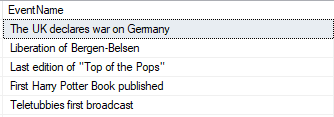


*There are only 4 events which happened after the last international one.*

Save this query as **Post-Int Notes**, then close it down.

1. Write a sub query to filter events to show only those which have an event name of longer than average length. You will need

the **AVG** and **LEN** functions to do this.



*The top 5 results out of 204 events.*

You can run your subquery separately by highlighting it to show that the average **EventName**length is 26 characters:

Sub Query average length

*The advantage of a sub query over typing 26 is that it will stay up to date if the average moves.*

Optionally save this exercise as **More words, more important.sql**, and close it down.

1. Create a subquery to list out all of those events whose:

Country id is not in the list of the last 30 country ids in alphabetical order; and

Category id is not in the list of the last 15 category ids in alphabetical order.

*You'll need to use the words****NOT IN****, as well as****TOP N****...****DESC****.*

Your query should list these 8 events (shown here in chronological order):



*These events occurred in countries and categories coming early in the alphabet.*

Save this query as **Early events**, then close it down.