Database Management System

Querying Data with Transact-SQL

Homework #1

Домашнє повинно бути у форматі .sql, в коментарях вказуєте номер завдання.

Посилання папки, куди ви повинні додати свої роботи: https://drive.google.com/drive/folders/1ILjqALKAbYjSLIeMYnHkKuCKc5arA0jN?usp=sharing

Підписуйте роботи у форматі: прізвище_ім'я_email

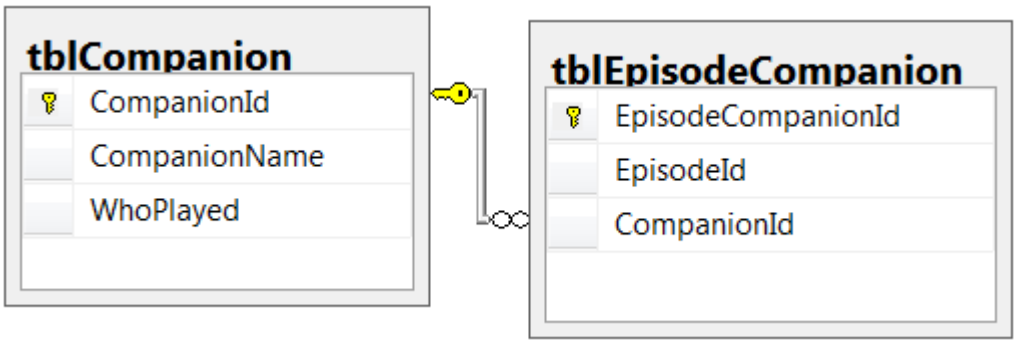
Термін виконнання – до 20.12.2020

1) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

If you haven't already done so, run the script in the above folder to generate the Doctor Who database.

The database contains details of companions, and the episodes each appeared in:



Each companion can appear in one or more episodes.

Use this to list the names of the companions who haven't featured in any episodes. You should find there's one of them, but we won't spoil the surprise by saying who it is!

Create a query based on the companions table, with an outer join to the episode companion table.

2) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

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:

```
--SELECT
--list of fields
--FROM
--table of events
--WHERE
--Event date > (
--SELECT MAX(Event date)
--FROM table of events
--WHERE country id = 21
--)
-- ORDER BY
--Event date (descending)
```

You should get these 4 events:

EventName	EventDate	CountryName
Donald Trump elected	2016-11-08	United States
Brexit vote	2016-06-23	United Kingdom
Karate included in Olympics	2016-06-01	Switzerland
Comic-Con conference	2016-05-27	United Kingdom

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

3) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

Write a query which lists out countries which have more than 8 events, using a correlated subquery rather than **HAVING**.

That is: list the names of countries from the countries table where for each country the number of events matching the country's id is more than 8.

If you list the countries in alphabetical order, you should get:

CountryName
France
International
Space
United Kingdom
United States

Although your query should be easy to read, one disadvantage is that you can't see how many events there were for each country.

4) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

The aim of this exercise is to show the number of events whose descriptions contain the words **this** and/or **that**:

IfThis	IfThat	Number of events
0	0	408
1	0	30
0	1	18
1	1	3

*Only 3 events have the holy grail: both **THIS** and **THAT**.*

*To do this you can find all events whose **EventDetails** column contains the word **this** or **that** respectively, using the **LIKE** keyword.*

In a single query solution you would have to use a **CASE** expression to determine the value of **IfThis** and **IfThat**, then group by the same **CASE** expression. This is messy, and makes it hard to make subsequent changes to your expression (as you have to do it in two places). Read on!

Create a query to solve this problem in two passes:

Pass	What to do
1	Create a CTE (called **ThisAndThat**?) to determine the values of the **IfThis** and **IfThat** flags for each event.
2	Use this CTE to get the required results, as shown at the start of this exercise.

As an aide-memoire, here is the syntax for a CTE:

```
-- create CTE
WITH CteName AS (
SELECT ...
)
-- then immediately use it
SELECT ... FROM CteName
```

If you get this working and still have spare time, try changing or extending your query to show the 3 events whose details contain both **this** and **that**:

EventName	EventDetails
The UK declares war on Germany	Following Germany's invasion of Poland, the UK's...
The Economist launched	The bible of absolute truth on earth was launched...
Karate included in Olympics	On this date he International Olympic Committee's...

Other than the text they contain, there's no obvious link between these 3 events.

5) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

The aim of this exercise is to list out all the continents which have:

- At least 3 countries; but also
- At most 10 events.

It's worth noting that there are many ways to solve this in SQL, but as is so often the case CTEs seem to give the most intuitive approach.

To do this, all that you need to do is to create the following CTEs:

CTE	What it should contain
ManyCountries	A list of continents having at least 3 countries
FewEvents	A list of continents in which no more than 10 events occurred

You can then join them together to show that there is only one country in both camps. We won't spoil the surprise, but it has 4 countries and 9 events.

6) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

The aim of this exercise is to count the number of events by a column called **Era** which you'll calculate, without including this calculation twice:

```
SELECT
    CASE
        WHEN year(e.EventDate) < 1900 THEN
            '19th century and earlier'
        WHEN year(e.EventDate) < 2000 THEN
            '20th century'
        ELSE '21st century'
    END AS Era,
    e.EventID
FROM
    tblEvent AS e
GROUP BY
    CASE
        WHEN year(e.EventDate) < 1900 THEN
            '19th century and earlier'
        WHEN year(e.EventDate) < 2000 THEN
            '20th century'
        ELSE '21st century'
    END
```

Here's what NOT to do - we want to avoid repeating the same calculation twice.

To do this, you can do the calculation in two bites, using a CTE to hold the intermediate stage. First create a query to show the era for each event:

Era	EventID
20th century	1
20th century	2
20th century	3
20th century	4
20th century	5
20th century	6
20th century	7
20th century	8
21st century	9

You could peek at the expression at the start of this exercise to get help on how to calculate the era!

Now store this as a CTE, and write a query using it which shows the number of events per era:

Era	Number of events
19th century and earlier	17
20th century	390
21st century	52

What the entire query should show when you run it.

7) Exercise

!!! Before you can do this exercise, you'll need to download and unzip this file

Create a query to show for each episode the series number, year and episode id:

EpisodeYear	SeriesNumber	EpisodeId
2005	1	1
2005	1	2
2005	1	3
2005	1	4

*You can use the **YEAR** function to derive the year for each episode's date.*

Now store this in a CTE, and pivot it to show the number of episodes by year and series number for the first 5 series:

EpisodeYear	1	2	3	4	5
2005	13	1	0	0	0
2006	0	13	1	0	0
2007	0	0	13	1	0
2008	0	0	0	14	0
2009	0	0	0	3	0
2010	0	0	0	1	13
2011	0	0	0	0	0
2012	0	0	0	0	0
2013	0	0	0	0	0
2014	0	0	0	0	0

You can get limit this to the first 5 series by only including these in the column headers.