# BAX-421 Homework 3

Mehul Rangwala

Fall 2023

Assignment Max. Score: 100 points

Due Date: December 10, 2023 11:59 PM (Both Sections)

## Instructions

1. Please submit your `*.sql` script files and a Word document containing the screenshots of your result sets.
2. Leaving questions unanswered will earn a zero.
3. You need to write the queries for the following databases:
    - PRESTIGE CARS database
    - COLONIAL database
    - ENTERTAINMENT AGENCY database
    - ACCOUNTS PAYABLE database
4. The five queries at the end are SQL Server based. Please install SQL Server and the SQL Server Management Console before writing these queries.
5. The last two queries are based on the topic of recursive CTEs, a topic we will cover in the last class session. So, you can hold off on those until we cover them in the class.
6. The scripts to install the databases are available on Canvas.
7. Please reuse the ER diagrams from your previous homework assignment.
8. **This assignment has been curated to develop individual competency in this very important part of preparation for your interviews. PLEASE DO NOT INDULGE IN COPYING EACH OTHERS' CODE. Any suspicion of copying will be reported.**

# PRESTIGE CARS DATABASE

## Question 1

Suppose the sales director wants to see the list of cars (make and model) bought on July 25, 2015. Write a SQL query to create this list.

## Question 2

The sales director now wants to see a list of all the cars (make and model) bought between July 15, 2018 and August 31, 2018. Write a SQL query to create this list.

## Question 3

The finance director is keen to ensure that cars do not stay on the firm's books too long —it ties up expensive capital. So, the finance director wants a list of the makes and models and the number of days that each vehicle remained, unsold, on the lot until they were bought by a customer. Create this list. The director wants to see this list in such a way that the cars which remained on the lot the most longer appears on the top.

## Question 4

Suppose the CFO wants to know the average daily purchase spend on cars over a six-month period. Create this list. Choose the period from July 1, 2015 through December 31, 2015.

## Question 5

As Prestige Cars has been selling cars for several years, the finance director wants to isolate the records for a specific year. In particular, the director wants to see a list of cars (make and model) sold in the year 2015. Create this list.

## Question 6

Now that the director has the sales list for 2015 from the previous part, he wants to compare the list of makes and models sold in both 2015 and in 2016. Create this list in an ordered fashion.

## Question 7

The CEO is convinced that some months are better for sales than others. She has asked for the sales for July 2015 to check out her hunch. List the vehicles (makes and models) sold during July 2015.

## Question 8

The CEO was disappointed about the sales in July 2015 from the previous question. Now, she wants to see sales for the entire third quarter of 2015. Generate this list.

## Question 9

The sales director wants to do an analysis of sales on a particular day of the week. So, you are asked to create a list of the vehicles sold on Fridays in the year 2016. Create this list.

## Question 10

The sales director was pleased with your list of vehicles sold on Fridays from the previous question. He now wants to take a look at the sales for the 26th week of 2017. Create such a list.

## Question 11

The HR manager needs to see how sales vary across days of the week. He has explained that he needs to forecast staff requirements for busy days. He wants to see, overall, which were the weekdays where Prestige Cars made the most sales in 2015. Create this list.

## Question 12

The HR manager liked the information you gave him in the previous question. However, the list in the previous question was a little too cryptic for him. He has requested that you display the weekday name instead of the weekday number. Regenerate the list from the previous question with this new request.

## Question 13

The sales manager has had another of her ideas. You can tell by her smile as she walks over to you in the cafeteria while you were having lunch. Her idea, fortunately, is unlikely to spoil your meal. What she wants is the total and average sales for each day of the year since Prestige Cars started trading. Create this list for her.

## Question 14

The CEO wants you to give her the total and average sales for each day of the month for all sales, ever. Create this list for the CEO.

## Question 15

Just as you are about to leave for home, the CEO flags you down on your way out of the office and insists that she needs the number of vehicles sold per month in 2018. Create this list showing the month number, month name, and the number of vehicles sold per month.

## Question 16

The HR manager has emailed another request. He needs to calculate the final bonus of a salesperson who is leaving the company and consequently needs to see the accumulated sales made by this staff member for a 75 day period up to July 25, 2015. The salesperson sold Jaguars for Prestige Cars.

## Question 17

The CEO has a single list of all customers. However, there is a problem with the list. She doesn't like the address split into many columns. She seeks your help. Recreate the customer list for the CEO with the address neatly formatted into one column with a dash (-) between the address and the `PostCode`. Since the list is for the CEO, you want the list to be as polished as possible. For instance, avoid `NULL`s in concatenated output.

## Question 18

The sales manager now wants a list of all the different make and model combinations that have ever been sold with the total sale price for each combination. However, this time, she wants the make and model output as a single column. She knows that this is an easy request for you, so she decides to hover near your desk. Write a query to create this list.

## Question 19

The marketing director thinks that some text are too long. She wants you to show the make names as acronyms using the first three letters of each make in a catalog of products. Create a list. For your list, create a single column showing the model name with the acronym for the make name in the parentheses.

## Question 20

The finance director wants you to show only the three characters at the right of the invoice number. Write a query to display this list.

## Question 21

In the Prestige Cars IT system, the fourth and fifth characters of the invoice number indicate the country where the vehicles were shipped. Knowing this, the sales director wants to extract only these characters from the invoice number field in order to analyze destination countries. Create such a list.

## Question 22

The sales director has requested a list of sales where the invoice was paid in Euros. Display this list.

## Question 23

The sales director now wants to see all the cars shipped to France but made in Italy.

## Question 24

The sales director wants a "quick list" of all vehicles sold and the destination country. Generate such a list.

## Question 25

The sales director wants you to make some reports that you send directly from MySQL to appear looking slightly easier to read. She wants you to display the `cost` from the `stock` table with a thousands separator and two decimals. Create a report with this column in the requested format.

## Question 26

The sales director is rushing to a meeting with the CEO. In a rush she requests you to create a report showing the make, model, and the sale price. The sale price in the report should include thousands separators, two decimals, and a British pound symbol. Create this report.

## Question 27

The CEO wants the sale price that you displayed in the previous question (with thousands separators and two decimals) to be now in German style —that is with a period as the thousands separator and a comma as the decimal. Create such a list.

## Question 28

Suppose the CEO requests a report showing the invoice number and the sale date, but the sale date needs to be in a specific format —first the day, then the abbreviation for the month, and finally the year in four figures —in this occurrence. Create this report.

## Question 29

Suppose the CEO requests the report in the previous question showing the invoice number and the sale date, but this time the sale date needs to be in the ISO format. Create this report.

## Question 30

The CEO, who is very happy with your work, now requests the report in the previous question to show the invoice number and the time at which the sale was made. The time needs to be in `hh:mm:ss` format showing AM or PM after it. Create this report.

## Question 31

Keeping track of costs is an essential part of any business. Suppose that the finance director of Prestige Cars Ltd. wants a report that flags any car ever bought where the parts cost was greater than the cost of repairs. In your report, the finance director wants you to flag such costs with an alert. Write a query to generate such a report.

## Question 32

The sales director wants some customer feedback. She knows that the sales database has comments from clients in it. But she does not need —or want —to display all the text in the comments. All she wants is to display the first 25 characters and then use ellipses to indicate that the text has been shortened. Write an SQL query to display the comments in this format.

## Question 33

This time, the sales director wants you to look at the profit on each car sold and flag any sale where the profit figure is less than 10 percent of the purchase cost —while at the same time the repair cost is at least twice the parts cost! Flag such records with a cost alert such as “Warning!”. Other sales need to be flagged as “OK”. Write a SQL query to generate this report.

## Question 34

The sales director looks at your previous result and becomes overjoyed thinking that SQL might be able to do much more! In addition to the cost alert displayed in the previous question, she wants to flag the costs as “Acceptable” if the net margin is greater than 10 percent, but less than 50 percent of the sale price. Otherwise, flag the cost as “OK”. Write an SQL query.

## Question 35

The finance director needs to manage exchange rate risk. So, he wants you to add each client’s currency area to the printout. Unfortunately, the database doesn’t have a field that holds the currency area. The currency areas that the director wants are “Eurozone” for countries in Europe, “Pound Sterling” for the United Kingdom, “Dollar” for the United States, and “Other” for all other regions. Write an SQL query to generate a report showing the country name and the corresponding currency region.

## Question 36

The finance director is overjoyed that you solved his previous conundrum and were able to add currency areas to the output. So, now he wants to take this one step further and has asked you for a report that counts the makes of the car according to the geographical zone where they were built. Divide the countries in which the cars were built into three regions: European, American, and British. Create such a report using SQL.

## Question 37

The sales director would like you to create a report that breaks down total sales values into a set of custom bandings by value (Under 5000, 5000-50000, 50001-100000, 100001-200000, Over 200000) and show how many vehicles have been sold in each category. Write an SQL query to create such a report.

## Question 38

The sales director wants to make it clear in which season a vehicle is sold. The seasons are: Winter (Nov - Feb), Spring (Mar, Apr), Summer (May, Jun, Jul, Aug), and Autumn (Sept, Oct). Create a report showing the month number, sale date, and the sale season.

## Question 39

The sales director has asked you to find all the sales for the top five bestselling makes. Write an SQL query to display such a list. Order by sale price descending. Write the query without using any window functions.

## Question 40

Suppose you are asked to show which colors sell the most. In addition, you also want to find the percentage of cars purchased by value for each color of vehicle. Write an SQL query to show this result set. Write the query without using any window functions.

## Question 41

The CEO requests a list of all the vehicle makes and models sold this year but not in the previous year. Write an SQL query to create this list. Write the query without using any window functions.

## Question 42

The sales manager wants to see a list of all vehicles sold in 2017, with the percentage of sales each sale represents for the year as well as the deviation of sales from the average sales figure. Hint: To simplify writing this query, you can use a view named `salesbycountry` included in the database. You can use the view like the source table. Write the query without using any window functions.

## Question 43

Classifying product sales can be essential for an accurate understanding of which products sell best. At least that is what the CEO said when she requested a report showing sales for 2017 ranked in order of importance by make. Write an SQL query to generate this report.

## Question 44

Buyer psychology is a peculiar thing. To better understand Prestige Cars' clients, the sales director has decided that she wants to find the bestselling color for each make sold. Write an SQL query to create this list.

## Question 45

Prestige Cars caters to a wide range of clients, and the sales director does not want to forget about the 80% that are outside the top 20% of customers. She wants you to take a closer look at the second quintile of customers —those making up the second 20% of sales. Her exact request is this "Find the sales details for the top three selling makes in the second 20% of sales." Write an SQL query to create this result set.

## Question 46

The CEO is interested in analyzing key metrics over time. Her latest request is that you obtain the total sales to each date and then display the running total of sales by value for each year. Write an SQL query to fulfill her request.

## Question 47

Sales at the company are increasing and senior management is convinced that effective analytics is a key factor of corporate success. The latest request to arrive in your inbox is for a report that shows both the first order and the last four sales for each customer. Write an SQL query to generate this result set.

## Question 48

The sales manager is on a mission to find out if certain weekdays are better for sales than others. Write a query so that she can analyze sales for each day of the week (but not weekends) in 2017 where there was a sale.

# COLONIAL DATABASE

## Question 1

List the reservation ID, trip ID, and trip date for reservations for a trip in Maine (ME). Write this query one way using JOIN and two ways using subqueries. In total, you will write three queries returning the same result set. One query will use only JOINs and no subqueries. The other two will use subqueries (with or without joins, if applicable.)

## Question 2

Find the trip ID and trip name for each trip whose maximum group size is greater than the maximum group size of every trip that has the type Hiking.

## Question 3

Find the trip ID and trip name for each trip whose maximum group size is greater than the maximum group size of at least one trip that has the type Biking.

# ENTERTAINMENT AGENCY DATABASE

## Question 1

Display all the entertainers who played engagements for customers Berg and Hallmark. For this question, write the query in two different ways - each way will use subqueries (with joins.) So in all you will write two different queries - each returning the same resultset.

## Question 2

What is the average salary of a booking agent?

## Question 3

Display the engagement numbers for all engagements that have a contract price greater than or equal to the overall average contract price.

## Question 4

How many of our entertainers are based in Bellevue?

## Question 5

Display which engagements occur earliest in October 2017.

## Question 6

Display all entertainers and the count of each entertainer's engagements.

## Question 7

List customers who have booked entertainers who play country or country rock. Use subqueries (including JOINS if applicable.)

## Question 8

Rewrite 7 using ONLY JOINs and no subqueries.

## Question 9

Find the entertainers who played engagements for customers Berg or Hallmark. Use subqueries (and JOINs.) There is only one query to write.

## Question 10

Repeat 9. No subqueries but only JOINs.

## Question 11

Display agents who haven't booked an entertainer. Answer in two different ways both ways using subqueries. So, in all, you will write two different queries (each using subqueries) showing the same result set.

## Question 12

Repeat 11 using ONLY JOINs and no subqueries.

## Question 13

Display all customers and the date of the last booking each made. Use subqueries.

## Question 14

List the entertainers who played engagements for customer Berg. Write the query in two different ways using subqueries. So, in all, you will write two different queries (each using subqueries) returning the same result set.

## Question 15

Rewrite the query in Question 14 using only JOINs (and no subqueries.)

## Question 16

Using a subquery, list the engagement number and contract price of all engagements that have a contract price larger than the total amount of all contract prices for the entire month of November 2017.

## Question 17

Using a subquery, list the engagement number and contract price of contracts that occur on the earliest date.

## Question 18

What was the total value of all engagements booked in October 2017?

## Question 19

List customers with no engagement bookings. Use only JOINs and NOT subqueries.

## Question 20

Repeat number 19. Write the query in two different ways returning the same result set. Each way will use a subquery.

## Question 21

For each city where our entertainers live, display how many different musical styles are represented. Display using subtotals and grand totals.

## Question 22

Which agents booked more than $3,000 worth of business in December 2017?

## Question 23

Display the entertainers who have more than two overlapped bookings.

## Question 24

Show each agent's name, the sum of the contract price for the engagements booked, and the agent's total commission for agents whose total commission is more than $1,000.

## Question 25

Display agents who have never booked a Country or Country Rock group.

## Question 26

Display the entertainers who did not have a booking in the 90 days preceding May 1, 2018.

## Question 27

List the entertainers who play the Jazz, Rhythm and Blues, and Salsa styles. Answer this question using two queries - one with subqueries (with or without joins) and another using only JOINs. In sum, two queries returning the same resultset.

## Question 28

List the customers who have booked Carol Peacock Trio, Caroline Coie Cuartet, and Jazz Persuasion. Write this query in three ways - each way uses subqueries of some sort - all returning the same resultset. In all, you will write three different queries, each one returning the same result set.

## Question 29

Display customers and groups where the musical styles of the group match all of the musical styles preferred by the customer.

## Question 30

Display the entertainer groups that play in a jazz style and have more than three members.

## Question 31

Display Customers and their preferred styles, but change 50's, 60's, 70's, and 80's music to 'Oldies'. This query should return 36 rows. Use `CASE...WHEN...THEN`.

## Question 32

Display all the engagements in October 2017 that start between noon and 5 p.m. Note: This database already has fields using the correct datatypes (date and time). Assume the dates and times were stored as strings. Write this query under such an assumption. This query should return 17 rows.

## Question 33

List entertainers and display whether the entertainer was booked (on the job) on Christmas 2017 (December 25th). For this, you have to display three columns – EntertainerID, Entertainer Stage Name, and a new column indicating if the engagement was booked on Christmas or not. The query should return 13 rows. Use `CASE...WHEN...THEN`.

## Question 34

Find customers who like Jazz but not Standards. The query should return 2 rows. Use `CASE...WHEN...THEN`.

## Question 35

For each customer, display the CustomerID, name of the customer (First name and Last name separated by a space), StyleName (style of music each customer prefers), and the total number of preferences for each customer.

## Question 36

For each customer, display the CustomerID, name of the customer (First name and Last name separated by a space), StyleName (style of music each customer prefers), the total number of preferences for each customer, and a running total of the number of styles selected for all the customers. Display the results sorted by Customer Name.

## Question 37

Display the Customer City, Customer, Number of Preferences of Music Styles, and a running total of preferences for each city overall.

## Question 38

Assign a row number for each customer. Display their CustomerID, their combined (First and Last) name, and their state. Return the customers in alphabetical order.

## Question 39

Assign a number for each customer within each city in each state. Display their Customer ID, their combined name (First and Last), their city, and their state. Return the customers in alphabetical order.

## Question 40

Show a list of all engagements. Display the start date for each engagement, the name of the customer, and the entertainer. Number the entertainers overall and number the engagements within each start date.

## Question 41

Rank all the entertainers based on the number of engagements booked for each. Arrange the entertainers into three groups (buckets). Remember to include any entertainers who haven't been booked for any engagements.

## Question 42

Rank all the agents based on the total dollars associated with the engagements that they've booked. Make sure to include any agents that haven't booked any acts.

## Question 43

Display a list of all of the engagements our entertainers are booked for. Display the entertainer's stage name, the customer's (combined) name, and the start date for each engagements, as well as the total number of engagements booked for each entertainer.

## Question 44

Display a list of all of the Entertainers and their members. Number each member within a group.

# ACCOUNTS PAYABLE DATABASE

## Question 1

Display the count of unpaid invoices and the total due.

## Question 2

Display the invoice details for each vendor in CA. Use JOINs and not subqueries.

## Question 3

Repeat question 2 using subqueries.

## Question 4

List vendor information for all vendors without invoices. Use JOINs and no subqueries.

## Question 5

Repeat 4 using two different subqueries returning the same result set. So, in all, you will write two different queries here - each using a different subquery.

## Question 6

List invoice information for invoices with a balance due less than average. Use subqueries.

## Question 7

List vendor name, invoice number, and invoice total for invoices larger than the largest invoice for vendor 34. Write two different subqueries yielding the same result set. So in all you should have two separate queries each returning the same result set.

## Question 8

List vendor name, invoice number, and invoice total for invoices smaller than the largest invoice for vendor 115. Use subquery in two different ways. So in all there will be two different subqueries generating the same result set.

## Question 9

Get the most recent invoice for each vendor. The query should return vendor name and the latest invoice date in the result set. Use a subquery without JOINs.

## Question 10

Repeat question 9. Use JOINs and no subqueries.

## Question 11

Get each invoice amount that is higher than the vendor's average invoice amount.

## Question 12

Get the largest invoice total for the top vendor in each state.

## Question 13

Display the GL account description, the number of line items are in each GL account, and the line item amount for accounts which have more than 1 line item.

## Question 14

What is the total amount invoiced for each GL account number. Display the grand total as well.

## Question 15

Which vendors are being paid from more than one account?

## Question 16

What are the last payment date and the total amount due for each vendor with each terms ID. Show the subtotal and grand total at each terms ID level.

## Question 17

Display the invoice totals from the invoices column and display all the invoice totals with a $ sign.

## Question 18

Write a query to convert invoice date to a date in a character format and invoice total in integer format. Both conversions should be performed in the same query. Please note, then integers have no decimals.

## Question 19

In the Invoices table, pad the single-digit and double-digit invoice numbers with one or two zeros before the invoice numbers. For example, the invoice number 1 should be displayed as 001, invoice number 20 should be displayed as 020, etc.

## Question 20

Write a query to return the invoice_total column with one decimal digit and the invoice_total column with no decimal digits.

## Question 21

Create a new table named Date_Sample using the script given below. Download this script from Canvas and run it in your MySQL Workbench on the ap database. Running this will create the Date_Sample table in your ap database.

```
USE ap;
CREATE TABLE date_sample
(
  date_id       INT       NOT NULL,
  start_date    DATETIME
);

INSERT INTO date_sample VALUES
(1, ’1986-03-01 00:00:00’),
(2, ’2006-02-28 00:00:00’),
(3, ’2010-10-31 00:00:00’),
(4, ’2018-02-28 10:00:00’),
(5, ’2019-02-28 13:58:32’),
(6, ’2019-03-01 09:02:25’);
```

Display the start_date column, a new date column - call it Format_1 which displays date in this format: Mar/01/86, a new date column - call it Format_2 which displays 3/1/86 where the month and days are returned as integers with no leading zeros, and a third date column - call it Format_3 which displays only hours and minutes on a 12-hour clock with an am/pm indicator, for example, 12:00 PM.

## Question 22

Write a query that returns the following columns from the Vendors table:

- The vendor_name column
- The vendor_name column in all capital letters
- The vendor_phone column
- A column that displays the last four digits of each phone number

When you get that working right, add the columns that follow to the result set. This can be more difficult for some students as these columns require use of nested functions.

- The vendor_phone column with the parts of the number separated by dots as in 111.111.1111
- A column that displays the second word in each vendor name if there is one and blanks if there isn’t

## Question 23

Write a query that returns these columns from the Invoices table:

- The invoice_number column
- The invoice_date column
- The invoice_date column plus 30 days
- The payment_date column
- A column named days_to_pay that shows the number of days between the invoice date and the payment date
- The number of the invoice date’s month
- The four-digit year of the invoice date

When you have this working, add a WHERE clause that retrieves just the invoices for the month of May based on the invoice date, and not the number of the invoice month.

## Question 24

Create a new table named string_sample using the script given below. Download this script from Canvas and run it in your MySQL Workbench on the ap database. Running this will create the string_sample table in your ap database.

```
USE ap;
CREATE TABLE string_sample
(
  emp_id        VARCHAR(3),
  emp_name      VARCHAR(25)
);

INSERT INTO string_sample VALUES
('1', 'Lizbeth Darien'),
('2', 'Darnell O''Sullivan'),
('17', 'Lance Pinos-Potter'),
('20', 'Jean Paul Renard'),
('3', 'Alisha von Strump');
```

Write a query that returns these columns from the string_sample table you created with the above script:

- The emp_name column
- A column that displays each employee's first name
- A column that displays each employee's last name

Use regular expression functions to get the first and last name. If a name contains three parts, everything after the first part should be considered part of the last name. Be sure to provide for last names with hyphens and apostrophes. You can refer to references online to learn about the regular expressions in MySQL. It is required that you use regular expressions and no other way. Any other way will not earn you credit (partial or full) even if your results are right.

## Question 25

Write a query to display the vendor id, balance due, total balance due for all vendors in the Invoices table, and the total balance due for each vendor in the Invoices table. The total balance due for each vendor should contain a cumulative total by balance due. This query should return 11 rows.

## Question 26

Modify the query in the question above so it includes a column that calculates the average balance due for each vendor in the Invoices table.

## Question 27

Write a query to calculate a moving average of the sum of invoice totals. Display the month of the invoice date, sum of the invoice totals, and the four-month moving average of the invoice totals sorted by invoice month.

# SQL SERVER WINDOW FUNCTIONS QUERY 1

Consider a table with two columns. The first, ID, is an incrementing integer, the primary key of the table. The second column contains data values. Some data values may be NULL. Here is the script that creates the temp table with data values. The script is also available under the Files on Canvas.

To write this query, you can create a temp table instead of a permanent table in SQL Server. Here is the script you can use to generate the temp table. I have provided the script in a *.sql file because copying from this document and pasting did not retain the formatting. You can download the script file available on Canvas which contains this code. Open it in SQL Server Management Studio by clicking on File → Open → File. Run the script first and then begin writing your query underneath it.

```
-- Create the table
CREATE TABLE #TableValues(ID INT, Data INT);

-- Populate the table
INSERT INTO #TableValues(ID, Data)
VALUES(1,100),(2,100),(3,NULL),
(4,NULL),(5,600),(6,NULL),
(7,500),(8,1000),(9,1300),
(10,1200),(11,NULL);

-- Display the results
SELECT * FROM #TableValues;
```

SQL Server drops a temporary table automatically when you close the connection that created it. So you don't have to worry about deleting it. Every time you run this query, you will have to first create the temporary table.

The exercise is to use window functions to replace each NULL value with the previous non-NULL value.

# SQL SERVER WINDOW FUNCTIONS QUERY 2

The script below generates a table containing data for customer subscriptions to an online magazine. The data are randomly generated so everyone will have different data in the temporary table. Use this data and window functions to calculate the number of active subscriptions at the end of each month. There may be cancellation dates past the latest subscription date, and you can ignore those rows. The script to generate this table is also available under Files on Canvas.

```
-- Create the temp table
CREATE TABLE #Registrations(ID INT NOT NULL IDENTITY PRIMARY KEY,
DateJoined DATE NOT NULL, DateLeft DATE NULL);

-- Variables
DECLARE @Rows INT = 10000,
@Years INT = 5,
@StartDate DATE = '2011-01-01'

-- Insert 10,000 rows with five years of possible dates
INSERT INTO #Registrations (DateJoined)
SELECT TOP(@Rows) DATEADD(DAY,CAST(RAND(CHECKSUM(NEWID())) * @Years *
365 as INT) ,@StartDate)
FROM sys.objects a
CROSS JOIN sys.objects b
CROSS JOIN sys.objects c;

-- Give cancellation dates to 75% of the subscribers
UPDATE TOP(75) PERCENT #Registrations
SET DateLeft = DATEADD(DAY,CAST(RAND(CHECKSUM(NEWID())) * @Years * 365
as INT),DateJoined)
```

Your final result set should include four columns: MonthEnding containing the end date of the month, Number Subscribed, Number Unsubscribed, and Active Subscriptions.

## SQL SERVER WINDOW FUNCTIONS QUERY 3

In this exercise you have a table showing clock in and clock out times for employees. The script below generates a temporary table containing the clock in and clock out data for each employee. The script to generate this table is also available under Files on Canvas.

```
-- Create the table
DROP TABLE IF EXISTS #TimeCards;

CREATE TABLE #TimeCards(
TimeStampID INT NOT NULL IDENTITY PRIMARY KEY,
EmployeeID INT NOT NULL,
ClockDateTime DATETIME2(0) NOT NULL,
EventType VARCHAR(5) NOT NULL);

-- Populate the table
INSERT INTO #TimeCards(EmployeeID,
ClockDateTime, EventType)
VALUES
(1,'2023-01-02 08:00','ENTER'),
(2,'2023-01-02 08:03','ENTER'),
(2,'2023-01-02 12:00','EXIT'),
(2,'2023-01-02 12:34','ENTER'),
(3,'2023-01-02 16:30','ENTER'),
(2,'2023-01-02 16:00','EXIT'),
(1,'2023-01-02 16:07','EXIT'),
(3,'2023-01-03 01:00','EXIT'),
(2,'2023-01-03 08:10','ENTER'),
(1,'2023-01-03 08:15','ENTER'),
(2,'2023-01-03 12:17','EXIT'),
(3,'2023-01-03 16:00','ENTER'),
(1,'2023-01-03 15:59','EXIT'),
(3,'2023-01-04 01:00','EXIT');
```

Write a query which uses window functions to calculate how long each employee worked on each shift. The time worked on each shift in your result set should include hours:minutes:seconds. For example, if an employee worked for 7 hours 15 minutes, then it should display `07:15:00`. The final result set should contain three columns: `EmployeeID`, `WorkDate`, and `TimeWorked`.

## SQL SERVER RECURSIVE QUERY 4

Take a look at the `FolderHierarchy` table. Each folder has an ID, a name, and an ID of a parent folder, which is the folder from which we can access the given folder.

Show four columns: `ID`, `Name`, `ParentId`, and `Path`. The last column should contain the path to the folder, starting with '/' and followed by all folder names separated by '/'. At the end of the path, there should be the name of the given folder and a slash ('/'). An example for folder B, which is in the root folder A, would be: /A/B/.

The script below generates a temporary table containing the folder hierarchy details. The script to generate this table is also available under Files on Canvas.

```
-- Create the table
DROP TABLE IF EXISTS #FolderHierarchy;
GO

-- Create the table
CREATE TABLE #FolderHierarchy
(
ID INTEGER PRIMARY KEY,
Name VARCHAR(100),
ParentID INTEGER
);
GO

-- Populate the table
INSERT INTO #FolderHierarchy VALUES
(1, 'my_folder', NULL),
(2,'my_documents', 1),
(3, 'events', 2),
(4, 'meetings', 3),
(5, 'conferences', 3),
(6, 'travel', 3),
(7, 'integration', 3),
(8, 'out_of_town', 4),
(9, 'abroad', 8),
(10, 'in_town', 4);
GO
```

# SQL SERVER RECURSIVE QUERY 5

The table `Destination` contains the names of four cities. You want to travel among the four cities starting from `Warsaw`. We have another table named `Ticket` which lists all possible flight connections and the cost.

Your task is to find the path which will be **cheapest** in terms of the **total tickets' cost**. List all paths starting from `Warsaw` that go through all four cities. Order the paths in **descending** order by `TotalCost`.

Your result set should contain the following columns: `Path` which contains city names separated by `->`, `LastId` which contains the ID of the last city, `TotalCost` which contains the total cost of the tickets, `NumPlacesVisited` which contains the number of places visited; it should equal **4**.

The script below generates the temporary tables. This script is also available under Files on Canvas.

```
DROP TABLE IF EXISTS #Destination;
GO

-- Create the table
CREATE TABLE #Destination
(
ID INTEGER PRIMARY KEY,
Name VARCHAR(100)
);
GO

-- Populate the table
INSERT INTO #Destination VALUES
(1, 'Warsaw'),
(2,'Berlin'),
(3, 'Bucharest'),
(4, 'Prague');
GO

DROP TABLE IF EXISTS #Ticket;
GO

-- Create the table
CREATE TABLE #Ticket
(
CityFrom INTEGER,
CityTo INTEGER,
Cost INTEGER
);
GO
```

```
-- Populate the table
INSERT INTO #Ticket VALUES
(1, 2, 350),
(1, 3, 80),
(1, 4, 220),
(2, 3, 410),
(2, 4, 230),
(3, 2, 160),
(3, 4, 110),
(4, 2, 140),
(4, 3, 75);
GO
```