INFO 430: Database Design and Management

Assignment 4: Working with Window Functions, Date Functions, and String Functions

Due Date: Thursday, May 23 (by end of Day)

Learning Objectives

- Demonstrate ability to write SQL queries that leverage Window functions to obtain desired results.
- Demonstrate ability to use Date and String functions to format output to a desired format to present results effectively.

Background

The database for this assignment is based on the data provided by Mobi Bike Share (mobibikes.ca), a public bike share system located in Vancouver, BC, Canada. Users sign up for membership and may choose from one of the various membership types. The original dataset consists of the following columns:

Column name	Desciption
depart_time	Departing time of the bike renter
return_time	Time the bike was returned
bike_id	ID of the bike rented
Bike type	Type of the bike (either electric bike or non-electric bike)
Departure station	Departure station name
Return station	Return station name
Membership type	Membership type
distance_covered	Distance travelled during the bike rental (in meters)
duration	Time the bike rental lasted (in seconds)
depart_temp	Temperature at departure of the renter (in degrees Celsius)
return_temp	Temperature at return of the renter (in degrees Celsius)
stopover_duration	Time spent in stopovers (in seconds)
num_stopovers	Total number of stopovers

From the above dataset, I created the following four tables for the relational database:

```
tblBikeType(bike_type_id TINYINT PRIMARY KEY IDENTITY(1,1),
                                        bike_type VARCHAR(50)
                                        )

tblStation(station_id INTEGER PRIMARY KEY IDENTITY(1,1),
                            station_name VARCHAR(100)
                            )

tblMembershipType(
                    membership_type_id INTEGER PRIMARY KEY IDENTITY(1,1),
                    membership_type VARCHAR(100)
                    )
```

```
tblBikeRentals
CREATE TABLE tblBikeRentals(
                                    rent_id INTEGER PRIMARY KEY IDENTITY(1,1),
                                    depart_time DATETIME,
                                    return_time DATETIME,
                                    bike_id INTEGER,
                                    bike_type_id TINYINT,
                                    depart_station_id INTEGER,
                                    return_station_id INTEGER,
                                    membership_type_id INTEGER,
                                    distance_covered INTEGER,
                                    duration INTEGER,
                                    depart_temp INTEGER,
                                    return_temp INTEGER,
                                    stopover_duration INTEGER,
                                    num_stopovers INTEGER
                                    )
```

The database name is ***BikeShare_db***. Please use this database to complete this assignment. Please use the same Server login credentials as before.

1. Write a SQL query to return the top 3 longest trips in hours from each starting station. Your query should list starting station name, rent_id, trip duration (in hours), and rank. Note: trip duration is recorded in seconds, so you need to convert that to hours in your query, rounded to 2 decimal places. Also, station names are recorded with the original station IDs are starting characters in the name (e.g., 0230 Alexander & Railway). Your query should return the station name with the leading numbers stripped (e.g., Alexander & Railway). Hint: use string functions to strip leading numbers from the station name. The sample output below shows how your query results should be formatted (note: this is not the complete set of rows in the result).

	Station Name	rent_id	Duration (hrs)	Rank
1	10th & Cambie	69488	46.84	1
2	10th & Cambie	54943	26.38	2
3	10th & Cambie	94497	15.39	3
4	Burrard Station (Melville & Dunsmuir)	38425	5.15	1
5	Burrard Station (Melville & Dunsmuir)	61339	3.27	2
6	Burrard Station (Melville & Dunsmuir)	17145	3.24	3

2. Write a SQL query to return the day of the month and the change of the number of rides from the previous day where the following conditions are met:
    - The bike renter returned the bike to the same station where they rented the bike.
    - The change in the number of rides from the previous day is greater than 0.

The excerpt below shows sample output (note: there should be 15 rows in the output).

Hint: you need to use Date function to extract day of month from the appropriate column.

	Day of Month	Change in numRides from Prev Day
1	3	147
2	5	245
3	7	28
4	9	434
5	12	382
6	13	257
7	16	87

3. Write a query to return for each membership type, the top 2 departing stations (ranked by number of rentals), a count of the number of rentals, bike type used, and rank. Please use dense rank approach to compute the rankings. As in question 1, your query output should display station nation with leading numbers stripped. Sample output is shown below (note: not a complete set of rows in the result). (Hint: You may also need to use ROW_NUMBER() Window function to ensure that the results are presented with desired rank ordering. To obtain the desired output, you will need to sort the final result in ascending order of membership type and row number). Sample output is shown below (note: does not include all the rows in the result).

	membership_type	Depart Station	numRides	bike_type	Rank
1	24 Hour	Stanley Park - Third Beach Parking Lot	21	Non Electric Bike	1
2	24 Hour	Stanley Park - Information Booth	21	Non Electric Bike	1
3	24 Hour	Stanley Park - Totem Poles	19	Non Electric Bike	2
4	30 Day Pass	Hornby & Drake	101	Non Electric Bike	1
5	30 Day Pass	Stanley Park - Third Beach Parking Lot	93	Non Electric Bike	2
6	365 Corporate Plus	Richards & Davie	101	Non Electric Bike	1
7	365 Corporate Plus	Hornby & Drake	82	Non Electric Bike	2

4. Write a query to return the following columns: station name, date, and difference in the number of rentals (i.e., [number of rentals where station served a return station] – [number of rentals where the station served as a depart station]) where the following conditions are met:
<u>Limit output to dates where the difference in the number of rentals is the highest for each station and the difference is at least 25.</u>
This query helps to determine dates when a given station served more as a return station than as a departing station. Hint: You need to Date functions and string functions to present the output in the desired format. Sample output is shown below (sample rows only).

	Station Name	Date	Number of rentals diff
1	Davie & Beach	Saturday, April 13	122
2	Olympic Village Station	Tuesday, April 23	48
3	Vancouver Art Gallery	Thursday, April 18	30
4	Bute & Robson	Friday, April 19	34
5	Yaletown-Roundhouse Station	Tuesday, April 30	43
6	Stanley Park - Second Beach North	Sunday, April 14	88
7	Rogers Tower	Saturday, April 20	67
8	Pender & Burrard	Tuesday, April 9	25
9	Cardero & Bayshore	Friday, April 5	44
10	Beach & Seymour	Sunday, April 14	33

What to Submit

Upload your SQL script file to Canvas.