

MSIS 5663 – Advanced Data Wrangling – Spring 2024

Assignment 5

(5 points – Points indicated for each question)

Submit your assignment through Canvas by **Monday, March 18 – 11:59 pm**

- **Type your answers** and submit it to Canvas as a Microsoft Word or pdf document.
- *Late assignments will be penalized 10% of grade for each hour past the due date and time (the first minute late starts the first hour) and will have a 0 grade after the solution is posted (usually the next day).*
- All questions/ disputes about assignment grades must be resolved within one week after grades for that assignment are posted. After that, the grades will not be changed.

Policies:

This is an **individual** assignment. Problems like these are likely to appear on your exam. Do not consult other students or otherwise plagiarize. Violations **will** be subject to Academic Integrity actions. Violations may also be reported to the Director of your graduate program and may result in loss of assistantship opportunities.

From OSU Academic Integrity Site:

Unauthorized Collaboration: Completing an assignment or examination with other students, turning in work that is identical or very similar to others' work, or receiving help on assignments without permission of the instructor. This may also include excessively relying upon and borrowing the ideas and work of others in a group effort.

Plagiarism: Presenting the written, published, or creative work of another as the student's own work. Whenever the student uses wording, arguments, data, design, etc., belonging to someone else in a paper, report, oral presentation, or other assignment, the student must make this fact explicitly clear by correctly citing the appropriate references or sources. The student must fully indicate the extent to which any part or parts of the project are attributed to others. The student must also provide citations for paraphrased materials. The following are examples of plagiarism:

- Copying another student's assignment, computer program or examination with or without permission from the author.
- Copying another student's computer program and changing only minor items such as logic, variable names, or labels.

Generative AI Tool Use:

Oklahoma State University: "Students are prohibited from using any generative AI tools such as ChatGPT, Bing AI, or Bard when completing course assignments. Use of these tools, or other similar generative AI tools, will not be tolerated and will be considered plagiarism and could result in the student failing the course. Any incident detected will be addressed through the university's academic integrity procedures."

I hereby declare that I have not engaged in Unauthorized Collaboration, Plagiarism, or the use of Generative AI in completing this assignment. I understand that violations **will** result in Academic Integrity action.

Sign or type your name here: Brock Taylor Bennett

Date: March 17, 2024

SPECIAL NOTES AND INSTRUCTIONS:

START EARLY – LONG ASSIGNMENT.

DO NOT PLAGIARIZE OR USE ANYONE ELSE'S COMPUTER.

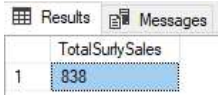
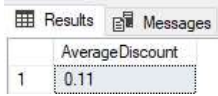

DO NOT DISCUSS THE QUERIES WITH ANYONE ELSE. IT SHOULD BE YOUR WORK. QUESTIONS SUCH AS THIS WILL BE ON THE EXAM.

YOU ARE ALLOWED TO POST QUESTIONS/COMMENTS REGARDING PYTHON ISSUES ONLY ON THE DISCUSSION BOARD. **NO QUESTION/COMMENTS ON THE SQL QUERIES THEMSELVES, INCLUDING THE STORED PROCEDURES, ARE PERMITTED.**

**** YOU MUST HAVE RESOLVED EVERYTHING BEFORE THURSDAY MARCH 14TH MIDNIGHT (96-HOUR WINDOW PRIOR TO DEADLINE BECAUSE OF 2-WEEK SUBMISSION DEADLINE). NO QUESTIONS/CLARIFICATIONS WILL BE PERMITTED AFTER THAT ****.

ACADEMIC INTEGRITY VIOLATIONS WILL RESULT IN A ZERO GRADE AND FURTHER ACTION.

Step 2: (3 points) Formulate the queries (SQL and Extended SQL) and make sure they produce the output shown:

No	Requirement	Output
1	<p>(0.25 points) <u>Slice query</u>: Find the Total Quantity of the product with brand name "Surly".</p>	 <pre>--1 Slice Query: Find the Total Quantity of the product with Brand Name "Surly" SELECT SUM(FactSales.quantity) AS Total_Quantity FROM FactSales JOIN DimProducts ON FactSales.product_id = DimProducts.Product_id WHERE DimProducts.brand_name = 'Surly';</pre>
2	<p>(0.25 points) <u>Dice query</u>: Find the Average Discount for product category 'Mountain Bikes' for Customers living in state 'NY'. Note: Use the SQL ROUND() function to round the average out to 2 decimal places.</p>	 <pre>--2 Dice Query: Find the Average Discount for product category 'Mountain Bikes' for Customers living in state 'NY' SELECT ROUND(AVG(FactSales.discount), 2) AS AverageDiscount FROM FactSales JOIN DimProducts ON FactSales.product_id = DimProducts.Product_id JOIN DimCustomers ON FactSales.customer_id = DimCustomers.Customer_id WHERE DimProducts.category_name = 'Mountain Bikes' AND DimCustomers.state = 'NY';</pre>
3	<p>(0.5 points) <u>Rollup Query</u>: Find the Total Undiscounted Sales by Product Category. Notes:</p> <ol style="list-style-type: none"> 1. Undiscounted Sales = List_Price * Quantity for each product in the category 2. Use COALESCE() function to get "All Categories" label for the grand total 3. Use FORMAT() function to display in dollars. 4. Use the GROUP BY ROLLUP() function. 5. Use ORDER BY to order by category name in descending order. 	 <pre>--3 Rollup Query: Find the Total Undiscounted Sales by Product Category. SELECT COALESCE(Category_Name, 'All Categories') AS Category, FORMAT(SUM(List_Price * Quantity), 'C', 'en-US') AS Total_Undiscounted_Sales FROM FactSales FS JOIN DimProducts DP ON FS.product_id = DP.Product_id GROUP BY ROLLUP(Category_Name) ORDER BY Category DESC;</pre>

4

(0.5 points) Drill-Down Query: Display the Total Undiscounted Sales by Product Category for each year (Drill-down by Year).

Notes:

1. Undiscounted Sales = List_Price * Quantity for each product in the category
2. Use FORMAT() function to display in dollars.
3. Use ORDER BY to order by category name and year within each category, in descending order.

Results		Messages	
	Category	Model_Year	UnDiscounted_Sales
1	All Categories	NULL	\$7,438,010.06
2	Children Bicycles	2018	\$17,489.33
3	Children Bicycles	2017	\$153,754.49
4	Children Bicycles	2016	\$120,385.71
5	Children Bicycles	NULL	\$291,629.53
6	Comfort Bicycles	2018	\$19,699.74
7	Comfort Bicycles	2017	\$143,992.98
8	Comfort Bicycles	2016	\$221,995.95
9	Comfort Bicycles	NULL	\$385,688.67
10	Cruisers Bicycles	2018	\$111,328.69
11	Cruisers Bicycles	2017	\$312,325.36
12	Cruisers Bicycles	2016	\$542,801.81
13	Cruisers Bicycles	NULL	\$966,455.86
14	Cyclocross Bicycles	2018	\$30,095.92
15	Cyclocross Bicycles	2017	\$227,499.35
16	Cyclocross Bicycles	2016	\$465,514.53
17	Cyclocross Bicycles	NULL	\$723,109.80
18	Electric Bikes	2018	\$172,399.53
19	Electric Bikes	2017	\$218,679.37
20	Electric Bikes	2016	\$425,998.58
21	Electric Bikes	NULL	\$817,077.48
22	Mountain Bikes	2018	\$142,395.31
23	Mountain Bikes	2017	\$962,954.12
24	Mountain Bikes	2016	\$1,674,605.64
25	Mountain Bikes	NULL	\$2,779,955.07
26	Road Bikes	2018	\$267,965.17
27	Road Bikes	2017	\$1,206,128.48
28	Road Bikes	NULL	\$1,474,093.65

--4 Drill Down Query: Display the Total Undiscounted Sales by Product Category for each year(Drill-Down by Year)

-- Calculate undiscounted sales by product category for each year

```
SELECT COALESCE(dp.Category_name, 'All Categories') AS
Category_name, dp.model_year, FORMAT(SUM(fs.List_Price *
fs.Quantity), 'C', 'en-US') AS Total_Undiscounted_Sales
FROM FactSales fs
JOIN DimProducts dp ON fs.product_id = dp.product_id
JOIN DimDate dd ON fs.Order_Date_Key = dd.Date_Key
GROUP BY dp.Category_name, dp.model_year
WITH ROLLUP
ORDER BY
CASE WHEN dp.Category_name IS NULL THEN ''
ELSE dp.Category_name
END ASC, dp.model_year DESC;
```

5 (0.5 points) Cube Query: Find the Undiscounted Total Sales by Product Category for every combination product category and brand name.

Notes:

1. Undiscounted Sales = List_Price * Quantity for each product in the category.
2. Use COALESCE() function to get "All Categories" label for the grand total of categories.
3. Use COALESCE() function to get "All Brands" label for the grand total of brands.
4. Use FORMAT() function to display in dollars.
5. Use the GROUP BY CUBE() function.
6. Use ORDER BY to order by category name and brand name within each category, in descending order.

Results		Messages	
	Category	Brand	UnDiscounted_Sales
1	All Categories	Trek	\$4,369,534.36
2	All Categories	Surly	\$973,402.30
3	All Categories	Sun Bicycles	\$346,782.30
4	All Categories	Strider	\$2,249.87
5	All Categories	Ritchey	\$87,748.83
6	All Categories	Pure Cycles	\$159,160.00
7	All Categories	Heller	\$178,118.75
8	All Categories	Haro	\$183,397.03
9	All Categories	Electra	\$1,137,616.62
10	All Categories	All Brands	\$7,438,010.06
11	Children Bicycles	Trek	\$45,637.94
12	Children Bicycles	Sun Bicycles	\$2,639.76
13	Children Bicycles	Strider	\$2,249.87
14	Children Bicycles	Haro	\$30,398.76
15	Children Bicycles	Electra	\$210,703.20
16	Children Bicycles	All Brands	\$291,629.53
17	Comfort Bicycles	Sun Bicycles	\$125,373.36
18	Comfort Bicycles	Electra	\$260,315.31
19	Comfort Bicycles	All Brands	\$385,688.67
20	Cruisers Bicycles	Sun Bicycles	\$155,097.70
21	Cruisers Bicycles	Pure Cycles	\$159,160.00
22	Cruisers Bicycles	Electra	\$652,198.16
23	Cruisers Bicycles	All Brands	\$966,455.86
24	Cyclocross Bicycles	Trek	\$251,399.27
25	Cyclocross Bicycles	Surly	\$471,710.53
26	Cyclocross Bicycles	All Brands	\$723,109.80
27	Electric Bikes	Trek	\$758,997.81
28	Electric Bikes	Sun Bicycles	\$43,679.72
29	Electric Bikes	Electra	\$14,399.95
30	Electric Bikes	All Brands	\$817,077.48
31	Mountain Bikes	Trek	\$1,905,123.16
32	Mountain Bikes	Surly	\$435,974.30

-- 5 Cube Query: Find the Undiscounted Total Sales by Product Category for every combination

--product Category for every combination product category and brand name.

```
SELECT COALESCE(dp.Category_name, 'All Categories') AS
Category_name, COALESCE(dp.brand_name, 'All Brands') AS
Brand_name, FORMAT(SUM(fs.List_Price * fs.Quantity), 'C', 'en-
US') AS Total_Undiscounted_Sales
```

```
FROM FactSales fs
```

```
JOIN DimProducts dp ON fs.product_id = dp.product_id
```

```
GROUP BY CUBE(dp.Category_name, dp.brand_name)
```

```
ORDER BY CASE WHEN dp.Category_name IS NULL THEN 'AAA'
```


		<p>ELSE dp.Category_name END ASC, COALESCE(dp.brand_name, 'All Brands') DESC;</p> <p>Partial output shown.</p>																																	
6	<p>(0.5 points) <u>Top N Query:</u> Find the top 10 cities by Total Discounted Sales to Customers in that city.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. Discounted Sales = (List_Price – Discount) * Quantity for each product in the category. 2. Use FORMAT() function to display Discounted Sales in dollars. 3. Use ORDER BY to order by Discounted Sales, in descending order. Use the formula for the sum, not the formatted version of the sum in the ORDER BY statement. 	<div> <div> <div>Results</div> <div>Messages</div> </div> <table> <thead> <tr> <th></th><th>City</th><th>DiscountedSales</th></tr> </thead> <tbody> <tr><td>1</td><td>Mount Vernon</td><td>\$105,563.33</td></tr> <tr><td>2</td><td>Ballston Spa</td><td>\$98,619.75</td></tr> <tr><td>3</td><td>Howard Beach</td><td>\$95,328.99</td></tr> <tr><td>4</td><td>Canyon Country</td><td>\$79,378.56</td></tr> <tr><td>5</td><td>Smithtown</td><td>\$77,295.60</td></tr> <tr><td>6</td><td>Harlingen</td><td>\$76,869.87</td></tr> <tr><td>7</td><td>Webster</td><td>\$71,361.33</td></tr> <tr><td>8</td><td>San Angelo</td><td>\$70,308.05</td></tr> <tr><td>9</td><td>Astoria</td><td>\$68,860.76</td></tr> <tr><td>10</td><td>Troy</td><td>\$67,607.49</td></tr> </tbody> </table> <p>-- 6 Top N Query: Find the top 10 cities by Total Discounted Sales to Customers in that city -- Top N Query: Find the top 10 cities by Total Discounted Sales SELECT TOP 10 dc.City, FORMAT(SUM((fs.list_price - (fs.list_price * fs.discount)) * fs.quantity), 'C', 'en-US') AS Total_Discounted_Sales FROM FactSales fs JOIN DimCustomers dc ON fs.customer_id = dc.Customer_id GROUP BY dc.City ORDER BY SUM((fs.list_price - (fs.list_price * fs.discount)) * fs.quantity) DESC;</p> </div>		City	DiscountedSales	1	Mount Vernon	\$105,563.33	2	Ballston Spa	\$98,619.75	3	Howard Beach	\$95,328.99	4	Canyon Country	\$79,378.56	5	Smithtown	\$77,295.60	6	Harlingen	\$76,869.87	7	Webster	\$71,361.33	8	San Angelo	\$70,308.05	9	Astoria	\$68,860.76	10	Troy	\$67,607.49
	City	DiscountedSales																																	
1	Mount Vernon	\$105,563.33																																	
2	Ballston Spa	\$98,619.75																																	
3	Howard Beach	\$95,328.99																																	
4	Canyon Country	\$79,378.56																																	
5	Smithtown	\$77,295.60																																	
6	Harlingen	\$76,869.87																																	
7	Webster	\$71,361.33																																	
8	San Angelo	\$70,308.05																																	
9	Astoria	\$68,860.76																																	
10	Troy	\$67,607.49																																	
7	<p>(0.5 points) <u>Comparing dimensional database with normalized database:</u> Use the MyStore normalized database to produce a similar** result as the Roll-up query to find the Total Undiscounted Sales by Product Category (3 above).</p> <ol style="list-style-type: none"> 1. Undiscounted Sales = List_Price * Quantity for each product in the category 2. Use FORMAT() function to display in dollars. 3. Use the UNION ALL clause to combine the sales for each category with the 'All Categories' grand total (see overhead 51 in Lecture 6) 4. Use ORDER BY to order by category name in descending order. 	<table> <thead> <tr> <th></th><th>category_name</th><th>UnDiscounted_Sales</th></tr> </thead> <tbody> <tr><td>1</td><td>Road Bikes</td><td>\$38,699.92</td></tr> <tr><td>2</td><td>Mountain Bikes</td><td>\$6,207.96</td></tr> <tr><td>3</td><td>Electric Bikes</td><td>\$4,679.97</td></tr> <tr><td>4</td><td>Cyclocross Bicycles</td><td>\$3,499.99</td></tr> <tr><td>5</td><td>Cruisers Bicycles</td><td>\$10,842.85</td></tr> <tr><td>6</td><td>Comfort Bicycles</td><td>\$2,383.95</td></tr> <tr><td>7</td><td>Children Bicycles</td><td>\$4,069.88</td></tr> <tr><td>8</td><td>All categories</td><td>\$70,384.52</td></tr> </tbody> </table> <p>**The reason for difference with the numbers in question 3 is because MyStore database has fewer records.</p>		category_name	UnDiscounted_Sales	1	Road Bikes	\$38,699.92	2	Mountain Bikes	\$6,207.96	3	Electric Bikes	\$4,679.97	4	Cyclocross Bicycles	\$3,499.99	5	Cruisers Bicycles	\$10,842.85	6	Comfort Bicycles	\$2,383.95	7	Children Bicycles	\$4,069.88	8	All categories	\$70,384.52						
	category_name	UnDiscounted_Sales																																	
1	Road Bikes	\$38,699.92																																	
2	Mountain Bikes	\$6,207.96																																	
3	Electric Bikes	\$4,679.97																																	
4	Cyclocross Bicycles	\$3,499.99																																	
5	Cruisers Bicycles	\$10,842.85																																	
6	Comfort Bicycles	\$2,383.95																																	
7	Children Bicycles	\$4,069.88																																	
8	All categories	\$70,384.52																																	

Step 3: (1 point) Use Python to run SQL queries and print out report with titles:

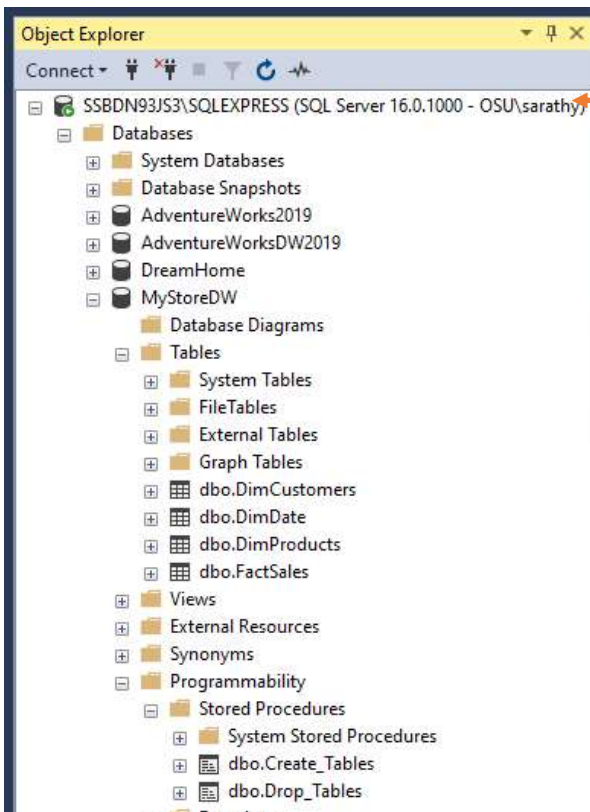
Your assignment submission will be the output of the python file **Assignment6_QueryReports.py**. In the code shown:

Code	Notes
<pre>import pandas as pd from tabulate import tabulate import pyodbc import os, sys #connection_string = "Driver={ODBC Driver 17 for SQL Server};Server=SSBDN93JS3\\SQLEXPRESS;Database=MyStoreDW;Trusted_Connection=yes;" connection_string = "Driver={ODBC Driver 17 for SQL Server};Server=stwssbsql01.ad.okstate.edu;Database=MyStoreDW;Trusted_Connection=yes;" cnxn = pyodbc.connect(connection_string) # cursor = cnxn.cursor() # def queryResult(title, query, outFile): cursor.execute(my_query) col_names = [i[0] for i in cursor.description] result = pd.DataFrame.from_records(cursor, columns=col_names) with open(outFile, 'a') as f: print("\n\n", title, file=f) print("\n\n", query, file=f)</pre>	<ol style="list-style-type: none">1. Choose an appropriate <i>pyodbc</i> connection_string as before.2. Choose a title for your query in title = "".3. Put the query between the pair of triple-quotes("""). IT IS IMPORTANT TO HAVE A SPACE in front of the each of the lines of the query from the second line (In the code: in front of "from", "where" and "GROUP BY ROLLUP")4. Run the program.

```
print("\n\n", tabulate(result, headers = col_names, tablefmt="grid",
showindex="always"), file=f)
#
title = "Undiscounted Sales By Year"
my_query = """select D.[Year] AS Year, SUM(S.[list_price]*S.[quantity]) AS Sales_By_Year
from FactSales S, DimDate D
where S.[Order_Date_Key] = D.[Date_Key]
GROUP BY ROLLUP ( D.[Year]);"""
queryResult(title, my_query, 'out.txt')
```

- Do this for each of the 7 queries. The file *out.txt* is in "append" mode. The result will be the *out.txt* file that contains the title, SQL query and the result formatted as a table.

Step4: Submit (1) your two python files (that will show your specific connection strings) that you used and (2) the out.txt file. Your **Assignment6_QueryReports.py** file should contain the last SQL query in it. (3) Also submit an image showing the name of your database software as below.



Must show your DBMS name.