Homework Assignment 04

This homework introduces some of the basic concepts of advanced SQL. The primary objective of this assignment is to gain an understanding and mastery of these concepts:

1. How to pivot rows to columns
2. How to generate a total row by cloning rows through a CARTESIAN JOIN.

*(**ATTENTION:** Use the MySQL Server 8 installed on **your computer** for this homework, as the log history **must be your own**. Also, do not use analytic/window/rollup functions for this assignment*)

Jan 31, 2023

1. Set the MySQL Server to log query history

Confirm that the MySQL Server's log history is turned on. For that, execute the queries below.

```
SHOW VARIABLES WHERE variable_name = 'log_output';
```

Variable_name	Value
log_output	TABLE

```
SHOW VARIABLES WHERE variable_name = 'general_log';
```

Variable_name	Value
general_log	ON

If the query results differ from above, run the queries below to set the log history on.

```
SET GLOBAL log_output = 'TABLE';
SET GLOBAL general_log = 'ON';
```

2. Query Example: Pivot

Analyze and understand the following query, which demonstrates how to pivot rows into columns.

```
  WITH t_data AS (
    SELECT 'cust01' AS custid, '01' Mon, 1000 amount UNION ALL
    SELECT 'cust01' AS custid, '02' Mon, 3000 amount UNION ALL
    SELECT 'cust01' AS custid, '03' Mon, 4000 amount UNION ALL
    SELECT 'cust01' AS custid, '04' Mon, 5000 amount UNION ALL
    SELECT 'cust02' AS custid, '01' Mon, 1100 amount UNION ALL
    SELECT 'cust02' AS custid, '02' Mon, 3010 amount UNION ALL
    SELECT 'cust02' AS custid, '03' Mon, 4001 amount UNION ALL
    SELECT 'cust02' AS custid, '04' Mon, 5400 amount
  )
SELECT custid
     , MAX((CASE WHEN MON='01' THEN amount ELSE NULL END)) AS Jan
     , MAX((CASE WHEN MON='02' THEN amount ELSE NULL END)) AS Feb
     , MAX((CASE WHEN MON='03' THEN amount ELSE NULL END)) AS Mar
     , MAX((CASE WHEN MON='04' THEN amount ELSE NULL END)) AS Apr
  FROM t_data
 GROUP BY custid;
```

t_data

custid	Mon	amount
cust01	01	1000
cust01	02	3000
cust01	03	4000
cust01	04	5000
cust02	01	1100
cust02	02	3010
cust02	03	4001
cust02	04	5400

Pivot

Pivotted t_data

custid	Jan	Feb	Mar	Apr
cust01	1000	3000	4000	5000
cust02	1100	3010	4001	5400

3. Query Example: Total

Analyze and understand the following query, which demonstrates how to clone rows through cartesian joins to generate the total row at the end.

```
  WITH t_data AS (
       SELECT 'cust01' AS custid, 1000 AS JAN, 3000 AS FEB
                                , 4000 AS MAR, 5000 AS APR
        UNION ALL
       SELECT 'cust02' AS custid, 1100 AS JAN, 3010 AS FEB
                                , 4001 AS MAR, 5400 AS APR
      ),

      /* a dummy set for a cartesian join */
      t_clone AS (
       SELECT 1 AS seq UNION ALL
       SELECT 2 AS seq
      )
/* Cartesian join (that is, no join condition) */
SELECT IF(b.seq=1, a.custid, 'TOTAL') AS custid
     , SUM(a.JAN) AS JAN, SUM(a.FEB) AS FEB
     , SUM(a.MAR) AS MAR, SUM(a.APR) AS APR
  FROM t_data a, t_clone b
 GROUP BY IF(b.seq=1, a.custid, 'TOTAL')
;
```

t_data

custid	JAN	FEB	MAR	APR
cust01	1000	3000	4000	5000
cust02	1100	3010	4001	5400

t_data with totals

custid	JAN	FEB	MAR	APR
cust02	1100	3010	4001	5400
cust01	1000	3000	4000	5000
TOTAL	2100	6010	8001	10400

4. Construct a Query: Pivot & Total

[Deliverables A] Construct **one SQL statement** with **WITH** clause (or inline views) that aggregates the order quantity(`purchaseorderdetail.OrderQty`) by month(`purchaseorderheader.OrderDate`) and category(`productcategory.name`). The query result must pivot the order month. There must be a month total row. Here is the expected result of the query:

Category	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Accessories	30250	28275	31700	33000	39600	51150	32200	46750	44000	15950	17600	30800
Clothing	NULL	300	NULL	20300	18300	1230	24000	NULL	NULL	300	NULL	17200
Components	45340	53530	50900	55240	67460	60310	72840	68560	58660	43570	21570	33670
Total	75590	82105	82600	108540	125360	112690	129040	115310	102660	59820	39170	81670

Source Data:

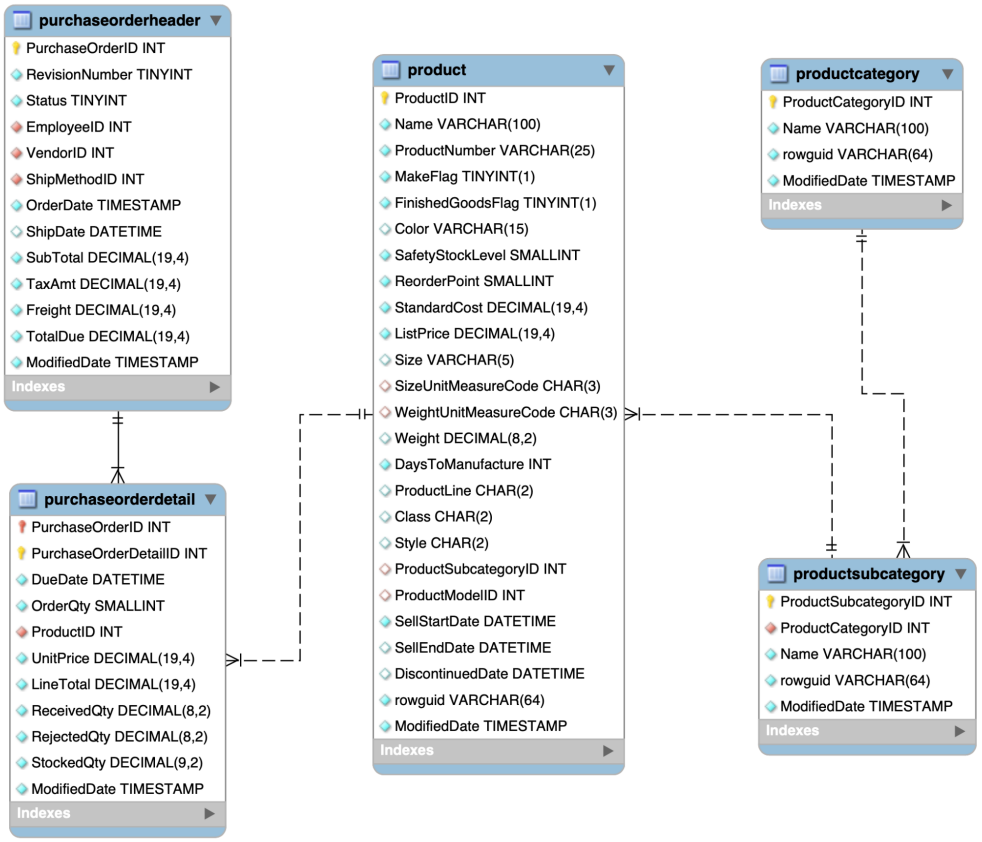
Database

- DBMS: MySQL Server 8
- AdventureWorks
- Data Dictionary: https://www.sqldatadictionary.com/AdventureWorks2014/)

Tables

- Purchaseorderdetail
- Purchaseorderheader
- Product
- Productsubcategory
- Productcategory

ERD



Hint: Suggested steps:

***[Step 1] Join** the five tables to extract the columns below. The OrderMonth can be parsed out of the OrderDate using a SUBSTR function:*

OrderMonth	Category	Subcategory	OrderQty
02	Components	Brakes	550
05	Components	Brakes	550
09	Components	Brakes	550
05	Components	Brakes	550
06	Components	Brakes	550
09	Components	Brakes	550
09	Components	Brakes	550
09	Components	Brakes	550
09	Components	Brakes	550
10	Components	Brakes	550
10	Components	Brakes	550
...			

***[Step 2] Aggregate** the result by Category and OrderMonth.*

***[Step 3] Pivot** the result.*

***[Step 4]** Create a **total** row.*

5. [Extra Credit] Subcategory & Subtotal

[Deliverables B] Extend query #4 and construct **one SQL statement** with **WITH** clause (or inline views) to have subcategories and their subtotals. The query result must be as below. .

Category	SubCategory	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEV
Accessories	Bike Racks	NULL	NULL	NULL	NULL	NULL	NULL	250	NULL	NULL	NULL	NULL	NULL
Accessories	Bike Stands	NULL	NULL	NULL	NULL	NULL	NULL	400	NULL	NULL	NULL	NULL	NULL
Accessories	Bottles and Cages	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	3000
Accessories	Cleaners	NULL	NULL	NULL	NULL	NULL	NULL	150	NULL	NULL	NULL	NULL	NULL
Accessories	Fenders	NULL	NULL	NULL	NULL	NULL	NULL	150	NULL	NULL	NULL	NULL	NULL
Accessories	Helmets	NULL	NULL	350	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Accessories	Hydration Packs	NULL	NULL	NULL	NULL	NULL	NULL	1000	NULL	NULL	NULL	NULL	NULL
Accessories	Lights	NULL	475	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Accessories	Locks	NULL	75	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Accessories	Panniers	NULL	225	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Accessories	Pumps	NULL	550	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Accessories	Tires and Tubes	30250	26950	31350	33000	39600	51150	30250	46750	44000	15950	17600	27800
Accessories	[Subtotal]	30250	28275	31700	33000	39600	51150	32200	46750	44000	15950	17600	30800
Clothing	Bib-Shorts	NULL	NULL	NULL	3050	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	Caps	NULL	NULL	NULL	NULL	NULL	630	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	Gloves	NULL	NULL	NULL	NULL	4800	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	Jerseys	NULL	NULL	NULL	NULL	NULL	600	24000	NULL	NULL	NULL	NULL	1200
Clothing	Shorts	NULL	300	NULL	15000	6000	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	Socks	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	300	NULL	16000
Clothing	Tights	NULL	NULL	NULL	2250	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	Vests	NULL	NULL	NULL	NULL	7500	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Clothing	[Subtotal]	NULL	300	NULL	20300	18300	1230	24000	NULL	NULL	300	NULL	17200
Components	Brakes	4400	4400	4400	4400	6600	6600	4400	6600	6600	2200	2200	2200
Components	Chains	240	180	300	240	360	360	240	360	360	120	120	120
Components	Pedals	20900	24200	24200	26400	27500	28600	31350	31350	29150	20350	9350	17050
Components	Saddles	19800	24750	22000	24200	33000	24750	36850	30250	22550	20900	9900	14300
Components	[Subtotal]	45340	53530	50900	55240	67460	60310	72840	68560	58660	43570	21570	33670
[All Categories]	[Grandtotal]	75590	82105	82600	108540	125360	112690	129040	115310	102660	59820	39170	81670

6. Submission Detail

Submit the deliverables A, C, and B (if applicable) as separate files. Do **NOT** zip them.

1. **Deliverables A**
    - Your SQL query in a text file
        - filename: [YourLastName]_[YourFirstName]_HW03**A**_SQL.txt
    - Documentation in a PDF:
        - Internal documentation: Explain your query
        - The screenshot of the execution result
        - filename: [YourLastName]_[YourFirstName]_HW03**A**_DOC.pdf

2. **Deliverables B (Extra Credit, Optional)**
    - Your SQL query in a text file
        - filename: [YourLastName]_[YourFirstName]_HW03**B**_SQL.txt
    - Documentation in a PDF:
        - Internal documentation: Explain your query
        - The screenshot of the execution result
        - Filename: [YourLastName]_[YourFirstName]_HW03**B**_DOC.pdf

3. **Deliverable C:** Run the query below to list your query history. Export the result in CSV format. ***(ATTENTION:** Failure to provide the log history in your submission will lead to zero points for the effort section of the grading)*
        - Filename: **[YourLastName]_[YourFirstName]_HW03_log.csv**

```
/* Replace YYYY-MM-DD with the date you started working on this
assignment */
SELECT '[YourANumber]' AS hawk_id, event_time, substr(argument,1,1000)
AS argument
  FROM mysql.general_log
 WHERE command_type = 'Query'
   AND event_time >= 'YYYY-MM-DD 00:00:00.000000'
   AND argument NOT LIKE 'SELECT 1%'
   AND argument NOT LIKE 'SHOW%'
ORDER BY event_time DESC
```

7. Rubric

- Points possible: 200
- Efforts: 50%
- Grading of submitted answers: 40%
- Overall quality of the deliverables: 10%
- Extra credit: 10%

STOP.