--Task 11 /\*Task 11: **Group By & Having**

--Task 11 **ROUGH WORK[DOUBLE CHECK FOR ERRORS]** - **work [ REMOVE UOP.DBO. AND THE $ SIGN AND THEN EXECUTE THE QUERIES.]**

for example the query

**SELECT \* FROM uop.dbo.FactSales$;**

Should be typed as

**SELECT \* FROM FactSales; \*/**

/\*Try the following queries and have a look at the results. What are the differences between them? Do they do what you expect?HAVING specifies a search condition for a group or an aggregate and can be used only with the SELECT statement. HAVING is typically used with a GROUP BY clause. When GROUP BY is not used, there is an implicit single, aggregated group.\*/

**SELECT Count(uop.dbo.FactSales$.TransactionNo) As [Number of Transactions], Month(FullDateTime) As MONTH from uop.dbo.DimTransactionType$**

**inner join uop.dbo.FactSales$ on uop.dbo.FactSales$.TransactionNo = uop.dbo.DimTransactionType$.TransactionNo**

**where year(FullDateTime) = 2019 group by Month(FullDateTime);**

-- next query does not work as promotion id is not null does not occur in my data.

**SELECT EmployeeLastName, EmployeeFirstName, Count(SalesAmount) from uop.dbo.FactSales$ inner join uop.dbo.DimEmployee$ on uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID WHERE PromotionID IS NULL GROUP BY EmployeeLastName, EmployeeFirstName Having Count(SalesAmount) > 43 ORDER by Count(SalesAmount) desc;**

-- next query does not work as promotion id is not null does not occur in my data.

**SELECT EmployeeLastName, EmployeeFirstName, COUNT(\*) AS SaleCount**

**FROM uop.dbo.FactSales$**

**INNER JOIN uop.dbo.DimEmployee$ ON uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID**

**WHERE PromotionID IS NOT NULL**

**GROUP BY EmployeeLastName, EmployeeFirstName**

**HAVING COUNT(\*) > 43**

**ORDER BY COUNT(\*) DESC;**

SELECT COUNT(\*)

FROM uop.dbo.FactSales$

WHERE PromotionID IS NOT NULL;

SELECT DISTINCT ActionTaken FROM uop.dbo.FactWaste$;

--1. List the amount of waste per month that is gifted.

SELECT

FORMAT(FullDate, 'yyyy-MM') AS WasteMonth,

SUM(Amount) AS TotalGiftedWaste

FROM uop.dbo.FactWaste$

WHERE ActionTaken = 'Gift'

GROUP BY FORMAT(FullDate, 'yyyy-MM')

ORDER BY WasteMonth;

--2. List how many different waste actions occurred per month in 2020.

SELECT

FORMAT(FullDate, 'yyyy-MM') AS WasteMonth,

COUNT(DISTINCT ActionTaken) AS UniqueWasteActions

FROM uop.dbo.FactWaste$

WHERE YEAR(FullDate) = 2020

GROUP BY FORMAT(FullDate, 'yyyy-MM')

ORDER BY WasteMonth;

--counter check for the above query 2

SELECT DISTINCT ActionTaken from uop.dbo.FactWaste$;

select \* from uop.dbo.FactWaste$ where ActionTaken = 'test';

--3. What is the minimum and maximum amount of waste for January 2019 per different actions.

SELECT

ActionTaken,

MIN(Amount) AS MinWasteAmount,

MAX(Amount) AS MaxWasteAmount

FROM uop.dbo.FactWaste$

WHERE YEAR(FullDate) = 2019 AND MONTH(FullDate) = 1

GROUP BY ActionTaken

ORDER BY ActionTaken;

--4. What is the amount of waste and the quantity of waste per month when the quantity of waste is at least 30

SELECT

FORMAT(FullDate, 'yyyy-MM') AS WasteMonth,

SUM(Amount) AS TotalWasteAmount,

SUM(Quantity) AS TotalWasteQuantity

FROM uop.dbo.FactWaste$

WHERE Quantity >= 30

GROUP BY FORMAT(FullDate, 'yyyy-MM')

ORDER BY WasteMonth;

-- eXTRA ONE

Select sum(SalesAmount) as 'SumSales', avg(SalesAmount) as 'avgSales', MIN(Quantity) as 'minQ', max(Quantity) as 'maxQ', uop.dbo.FactSales$.ProductCode, uop.dbo.DimProduct$.ProductName

From uop.dbo.FactSales$

inner join uop.dbo.DimProduct$ on uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode

Group by uop.dbo.FactSales$.ProductCode, uop.dbo.DimProduct$.ProductName

order by ProductCode