-- TASK 4 **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;**

--Task 4: Outer and Self Joins

-- simple join of two tables where the product code match the row is returned.

SELECT SalesAmount,ProductType FROM uop.dbo.FactSales$ inner join uop.dbo.DimProduct$ on uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode where PromotionID IS NULL;

-- would tell you the sales amount of every employee

SELECT SalesAmount FROM uop.dbo.FactSales$ right outer join uop.dbo.DimEmployee$ on uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID Where Quantity <2;

--

SELECT \* FROM uop.dbo.FactSales$ right outer join uop.dbo.DimEmployee$ on uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID Where Quantity <2;

--

SELECT uop.dbo.FactSales$.SalesAmount,

uop.dbo.FactSales$.Quantity,

uop.dbo.DimEmployee$.EmployeeFirstName,

uop.dbo.DimEmployee$.EmployeeType

FROM uop.dbo.FactSales$

RIGHT JOIN uop.dbo.DimEmployee$

ON uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID

WHERE Quantity < 2;

--

SELECT COUNT(\*) FROM UOP.DBO.DimEmployee$

WHERE EmployeeID NOT IN (SELECT DISTINCT EmployeeID FROM UOP.DBO.FactSales$);

--

SELECT COUNT(\*) FROM UOP.DBO.FactSales$

WHERE EmployeeID NOT IN (SELECT DISTINCT EmployeeID FROM UOP.DBO.DimEmployee$);

--

SELECT \* FROM uop.dbo.FactSales$ left join uop.dbo.DimEmployee$ on uop.dbo.FactSales$.EmployeeID = uop.dbo.DimEmployee$.EmployeeID Where Quantity <2;

--Join of more than 1 tables

SELECT \* FROM uop.dbo.FactSales$ inner join uop.dbo.DimProduct$ on uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode inner join uop.dbo.DimCategory$ on uop.dbo.DimCategory$.CategoryID = uop.dbo.DimProduct$.CategoryID where PromotionID IS NULL;

--

SELECT ProductCode, COUNT(\*)

FROM uop.dbo.DimProduct$

GROUP BY ProductCode

HAVING COUNT(\*) > 1;

--

SELECT CategoryID, COUNT(\*)

FROM uop.dbo.DimCategory$

GROUP BY CategoryID

HAVING COUNT(\*) > 1;

--Create queries to:

--1. List the date, category range, product name and employee first name and last name of every sale occurred in december in descending date order.

SELECT

FS.FullDateTime AS SaleDate,

DC.CategoryRange AS CategoryName,

DP.ProductName,

E.EmployeeFirstName,

E.EmployeeLastName

FROM uop.dbo.FactSales$ FS

INNER JOIN uop.dbo.DimProduct$ DP

ON FS.ProductCode = DP.ProductCode

INNER JOIN uop.dbo.DimCategory$ DC

ON DP.CategoryID = DC.CategoryID

INNER JOIN uop.dbo.DimEmployee$ E

ON FS.EmployeeID = E.EmployeeID

WHERE MONTH(FS.FullDateTime) = 12

ORDER BY FS.FullDateTime DESC;

--2. List all sale, transaction type, product, and waste records that have sales and waste recorded on the same days. (hint: try different joints)

--[This query helps identify products that were both sold and wasted on the same day.

--Useful for analyzing product efficiency, reducing waste, or understanding demand patterns.]

Select SalesAmount, TransactionType, ProductName, uop.dbo.FactWaste$.Amount, uop.dbo.FactWaste$.ActionTaken, uop.dbo.FactWaste$.Quantity

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

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

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

where Convert(date, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate

--USING LEFT JOIN [EVEN IF NO WASTE EXITS]

SELECT SalesAmount,

TransactionType,

ProductName,

uop.dbo.FactWaste$.Amount,

uop.dbo.FactWaste$.ActionTaken,

uop.dbo.FactWaste$.Quantity

FROM uop.dbo.FactSales$

LEFT JOIN uop.dbo.DimProduct$

ON uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode

LEFT JOIN uop.dbo.DimTransactionType$

ON uop.dbo.DimTransactionType$.TransactionNo = uop.dbo.FactSales$.TransactionNo

LEFT JOIN uop.dbo.FactWaste$

ON uop.dbo.DimProduct$.ProductCode = uop.dbo.FactWaste$.ProductCode

AND CONVERT(DATE, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate;

--USING RIGHT JOIN [ EVEN IF NO SALES EXIT]

SELECT SalesAmount,

TransactionType,

ProductName,

uop.dbo.FactWaste$.Amount,

uop.dbo.FactWaste$.ActionTaken,

uop.dbo.FactWaste$.Quantity

FROM uop.dbo.FactSales$

RIGHT JOIN uop.dbo.DimProduct$

ON uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode

RIGHT JOIN uop.dbo.DimTransactionType$

ON uop.dbo.DimTransactionType$.TransactionNo = uop.dbo.FactSales$.TransactionNo

RIGHT JOIN uop.dbo.FactWaste$

ON uop.dbo.DimProduct$.ProductCode = uop.dbo.FactWaste$.ProductCode

AND CONVERT(DATE, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate;

--Using FULL OUTER JOIN to Include All Sales and Waste (Even If No Matches Exist)

SELECT SalesAmount,

TransactionType,

ProductName,

uop.dbo.FactWaste$.Amount,

uop.dbo.FactWaste$.ActionTaken,

uop.dbo.FactWaste$.Quantity

FROM uop.dbo.FactSales$

FULL OUTER JOIN uop.dbo.DimProduct$

ON uop.dbo.FactSales$.ProductCode = uop.dbo.DimProduct$.ProductCode

FULL OUTER JOIN uop.dbo.DimTransactionType$

ON uop.dbo.DimTransactionType$.TransactionNo = uop.dbo.FactSales$.TransactionNo

FULL OUTER JOIN uop.dbo.FactWaste$

ON uop.dbo.DimProduct$.ProductCode = uop.dbo.FactWaste$.ProductCode

AND CONVERT(DATE, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate;

--Using CROSS JOIN to Pair Every Sale With Every Waste Record (Not Recommended- UNNECESSARY DUPLICATIONS]

/\*This creates a cartesian product, leading to a huge number of rows.

Not useful unless you need every possible sales-waste combination for further filtering.

Avoid using CROSS JOIN unless necessary!\*/

--3. List all sale, transaction type, product, and waste records that have at least sales recorded on the common sale-waste days. (hint: try different joints)

Select SalesAmount, TransactionType, ProductName, uop.dbo.FactWaste$.Amount, uop.dbo.FactWaste$.ActionTaken, uop.dbo.FactWaste$.Quantity

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

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

left join uop.dbo.FactWaste$ on uop.dbo.DimProduct$.ProductCode = uop.dbo.FactWaste$.ProductCode

and Convert(date, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate

-- for the above query If the last three rows of your result contain all NULL values for waste data, this means that those sales transactions did not have a corresponding waste record. This is expected behavior when using a LEFT JOIN.

--4. List all sale, transaction type, product, and waste records that have at least wastes recorded on the common sale-waste days. (hint: try different joints)

Select SalesAmount, TransactionType, ProductName, uop.dbo.FactWaste$.Amount, uop.dbo.FactWaste$.ActionTaken, uop.dbo.FactWaste$.Quantity

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

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

right join uop.dbo.FactWaste$ on uop.dbo.DimProduct$.ProductCode = uop.dbo.FactWaste$.ProductCode

and Convert(date, uop.dbo.FactSales$.FullDateTime) = uop.dbo.FactWaste$.FullDate