**Data preparation**

dbo.prognosesPervasco: gekopieerd van 4food.pervasco.productie (jaarschema)

aanpassingen gemaakt:

versioninfo column verwijdert voegt niks toe.

2016=> niet meenemen te weinig data

dbo.prognosesNapoleon: gekopieerd van 4food.napoleon.productie (jaarschema)

aanpassingen gemaakt:

versioninfo column verwijdert voegt niks toe.

2007=> niet meenemen te weinig data

dbo.opbNapoleon**:** gekopieerd van4food.napoleon.productie (opb)

aanpassingen gemaakt:

Dropped colloms:

| **Kolom** | **Reden voor verwijderen** |
| --- | --- |
| Lastpackagecompletion | Alle waarden NULL |
| Hidehistory | Alle waarden 0 |
| Supplierremark | Alle waarden NULL |
| basedOnOrderRowId | Alle waarden NULL |
| externalpicking | Alle waarden NULL |
| turnovergroupid | Alle waarden NULL |
| createdOn | Alle waarden NULL |
| createdBy | Alle waarden NULL |
| vatgroupid | Bijna alleen waarde 3, niet relevant voor scope |
| originCountryID | Alle waarden NULL |
| Tuperpallet | Alle waarden NULL |
| Palletdatacode | Alle waarden NULL |
| Venturecostcenterid | Alle waarden NULL |
| pickupFromSupplierid | Alle waarden NULL |
| loadingLocationID | Alle waarden NULL |
| username | Alle waarden NULL |
| userid | Alle waarden NULL |
| externalDelivery | Alleen waarde 0 |
| rowNumber | Niet relevant voor scope |
| actualTuWeight | Alle waarden NULL |
| deliveryConditionID | Te veel NULL, niet relevant voor scope |
| planProdQuantity | Alle waarden NULL |
| planProdDate | Alle waarden NULL |
| itemType | Alleen waarde 4, niet relevant voor scope |
| planningComment | Alleen NULL of leeg |
| planState | Alleen waarden 0 |
| wagework | Alleen waarden 0 |
| unloadingReference | Alle waarden NULL |
| loadingReference | Alle waarden NULL |
| fee | Alleen waarden 0 |
| loadingWarehouseID | Alle waarden NULL |
| qualityCheck | Alle waarden NULL |
| invoiceNumber | Alle waarden NULL |
| cmrNumber | Alle waarden NULL |
| externalReference | Alle waarden NULL |
| departureDate | Alle waarden NULL |
| salesContractID | Alle waarden NULL |
| rowStatus | Alleen waarden 0 |
| classificationId | Alle waarden NULL |
| originalQuantity | Relevante info, maar intern quantity gebruikt, niet meegenomen |
| previousState | Niet relevant voor scope |
| mainOrderRow | Alleen NULL en 0 waarden |
| orderChangeDate | >35% ontbrekend, niet relevant voor scope (betere tijd-measurement beschikbaar) |
| Eenheid | Helemaal leeg |
| Korting | Alleen waarde 0 |
| sort\_art | Alleen waarde 0 |
| volgnr | Alleen waarde 0 |
| backorder | Alleen waarde 0 |
| onderneming | Alleen waarde ‘001’, niet relevant voor scope |
| betalingconditie | Belangrijke info aanwezig, buiten scope opdracht |
| versioninfo | Belangrijke info aanwezig, buiten scope opdracht |
| tht | Buiten scope opdracht |
| inpaklijst | Alleen waarde 0 |
| handplanning | Alleen waarde 0 |
| opmerking | Helemaal leeg |
| palletaantal | Alleen NULL en 0 |
| Gereed | Belangrijke info aanwezig, buiten scope opdracht |
| Nettokg | Alleen NULL en 0 |
| Levertijdweek | Alleen NULL en 0 |
| Caramels | Alleen waarde 0 |

**dbo.orderinfoNapoleon:** gekopieerd van 4food.napoleon.productie (orderinfo)

aanpassingen gemaakt:

**dropped colloms:**

| **Kolom** | **Reden voor verwijderen** |
| --- | --- |
| clientTotalDiscount | Alle waarden NULL |
| carrierComment | Alle waarden NULL of leeg |
| ventureCostCenterID | Alle waarden NULL |
| contactpersonId | Alle waarden NULL |
| deliveryComment | Alle waarden NULL |
| characteristicId | Alle waarden NULL |
| ventureAddressId | Alle waarden NULL |
| invoiceAddressId | Alle waarden NULL |
| latestDeliveryDate | Alle waarden NULL |
| earliestDeliveryDate | Alle waarden NULL |
| importID | Alle waarden NULL of leeg |
| planningComment | Alle waarden NULL of leeg |
| deliveryAddressId | Alle waarden NULL |
| clientPlanGroupId | Alle waarden NULL |
| loadDateAndTime | >35% ontbrekend, niet relevant voor scope (betere tijd-measurement beschikbaar) |
| planningDate | orderDate wordt al meegenomen als tijd-measurement |
| deliveryDate | orderDate wordt al meegenomen als tijd-measurement |
| sourceid | Vooral NULL, overige data buiten scope opdracht |
| sourcetype | Buiten scope opdracht |
| transportHandling | Buiten scope opdracht |
| versioninfo | Buiten scope opdracht |
| removed | Buiten scope opdracht |
| orderModifiactionDate | Buiten scope opdracht |
| invoiceComment | Bijna alle waarden NULL, overige buiten scope opdracht |
| orderNumberYear | Hier staan het aantal orders per jaar, niet bijgehouden, buiten scope opdracht |
| orderNumberClient | Veel data aanwezig, buiten scope opdracht |
| orderNumber | Buiten scope opdracht |

2007=> niet meenemen te weinig data

Je gaat select moeten doen zodat hij breed inzetbaar is en je kan pervasco niet met realisaties van napoleon vergelijken verder kijken naar status en ordersoort

**Orderstatus 4 uitsluiten**

**Odertype meenemen: 2,3,4 en 18**

**Ordertype 18 zijn retourzendingen dus die omzetten naar negatieve waardes(quantity).**

**Jaren met dat van minder dan 6 maanden uitsluiten.**

**Nul waarden bij Quantity en PriceKG en TuWeight er uit halen**

**Select Quary met de goede parameters(orderstatus, orderType, volledige jaren):**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(orderDate) AS OrderYear,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

)

INSERT INTO dbo.Napoleon\_All (OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

opb.quantity AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.OrderYear = YEAR(oi.orderDate)

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18);

**Alles toegepast van bovenstaande notities:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(orderDate) AS OrderYear,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

)

INSERT INTO dbo.Napoleon\_All (OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.OrderYear = YEAR(oi.orderDate)

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG <> 0

AND opb.tuWeight <> 0;

**Outliers er uit (en de rest nog steeds):**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All2', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All2;

CREATE TABLE dbo.Napoleon\_All2

(

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

-- Alleen jaren met minstens 6 maanden data

SELECT

YEAR(orderDate) AS OrderYear,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

),

Quantiles AS

(

-- Percentielen voor Quantity over de hele dataset

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

)

INSERT INTO dbo.Napoleon\_All2 (OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

-- Retouren omzet naar negatief indien positief

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.OrderYear = YEAR(oi.orderDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

-- Outlierfilter alleen voor Quantity, negatieve values behouden

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity));

**Werkt ook voor andere database:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.test', 'U') IS NOT NULL

DROP TABLE dbo.test;

CREATE TABLE dbo.test

(

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

-- Alleen jaren met minstens 6 maanden data

SELECT

YEAR(orderDate) AS OrderYear,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.aquamossel-triton.prod].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

),

Quantiles AS

(

-- Percentielen voor Quantity over de hele dataset

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.aquamossel-triton.prod].[dbo].[opb] opb

)

INSERT INTO dbo.test (OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

-- Retouren omzet naar negatief indien positief

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.aquamossel-triton.prod].[dbo].[orderinfo] oi

INNER JOIN [4food.aquamossel-triton.prod].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.OrderYear = YEAR(oi.orderDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

-- Outlierfilter alleen voor Quantity, negatieve values behouden

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity));

**Alle nul waardes in Quantity en PriceKG uitgesloten:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

-- Alleen jaren met minstens 6 maanden data

SELECT

YEAR(orderDate) AS OrderYear,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

),

Quantiles AS

(

-- Percentielen voor Quantity over de hele dataset

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

)

INSERT INTO dbo.Napoleon\_All (OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

-- Retouren omzet naar negatief indien positief

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.OrderYear = YEAR(oi.orderDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND (opb.priceKG) > 0.001

-- Outlierfilter alleen voor Quantity, negatieve values behouden

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity));

**Omzetten naar per maand om prognoses te vergelijken (nog niet af denk ik):**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All2', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All2;

CREATE TABLE dbo.Napoleon\_All2

(

Jaar INT,

Maand INT,

OrderId INT,

OrderType INT,

OrderStatus INT,

ClientId NVARCHAR(50),

OrderDate DATETIME,

Artikelnummer NVARCHAR(50),

Quantity DECIMAL(18,3),

PriceKG DECIMAL(18,3),

TuWeight DECIMAL(18,3),

Artnaam NVARCHAR(255)

);

;WITH YearMonthCount AS

(

-- Alleen jaren met minstens 6 maanden data

SELECT

YEAR(orderDate) AS Jaar,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

),

Quantiles AS

(

-- Percentielen voor Quantity over de hele dataset (globaal)

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

)

INSERT INTO dbo.Napoleon\_All2 (Jaar, Maand, OrderId, OrderType, OrderStatus, ClientId, OrderDate, Artikelnummer, Quantity, PriceKG, TuWeight, Artnaam)

SELECT

YEAR(oi.orderDate) AS Jaar,

MONTH(oi.orderDate) AS Maand,

oi.id AS OrderId,

oi.orderType AS OrderType,

oi.orderStatus AS OrderStatus,

oi.clientId AS ClientId,

oi.orderDate AS OrderDate,

opb.artikelnummer AS Artikelnummer,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight,

opb.artnaam AS Artnaam

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.orderDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

-- Outlierfilter: negatieve hoeveelheden altijd meenemen

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity));

**Totaal per product per maand voor de vergelijking met prognose:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All2', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All2;

CREATE TABLE dbo.Napoleon\_All2

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

GemiddeldePrijs DECIMAL(18,3),

GemiddeldTuWeight DECIMAL(18,3),

);

;WITH YearMonthCount AS

(

SELECT

YEAR(orderDate) AS Jaar,

COUNT(DISTINCT MONTH(orderDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(orderDate)

HAVING COUNT(DISTINCT MONTH(orderDate)) >= 6

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.orderDate) AS Jaar,

MONTH(oi.orderDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.orderDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

)

INSERT INTO dbo.Napoleon\_All2 (Jaar, Maand, Artikelnummer, Artnaam, TotaalQuantity, GemiddeldePrijs, GemiddeldTuWeight)

SELECT

Jaar,

Maand,

Artikelnummer,

Artnaam,

SUM(Quantity) AS TotaalQuantity,

AVG(PriceKG) AS GemiddeldePrijs,

AVG(TuWeight) AS GemiddeldTuWeight

FROM Data\_CTE

GROUP BY Jaar, Maand, Artikelnummer, Artnaam

ORDER BY Jaar, Maand, Artikelnummer;

**Nieuwe versie met datum aanpassing:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_Realisaties', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_Realisaties;

CREATE TABLE dbo.Napoleon\_Realisaties

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

GemiddeldePrijs DECIMAL(18,3),

GemiddeldTuWeight DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

-- Alleen jaren meenemen die in minstens 10 maanden data hebben

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

-- Percentielen voor Quantity over de hele dataset (globaal)

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

)

INSERT INTO dbo.Napoleon\_Realisaties (Jaar, Maand, Artikelnummer, Artnaam, TotaalQuantity, GemiddeldePrijs, GemiddeldTuWeight)

SELECT

Jaar,

Maand,

Artikelnummer,

Artnaam,

SUM(Quantity) AS TotaalQuantity,

AVG(PriceKG) AS GemiddeldePrijs,

AVG(TuWeight) AS GemiddeldTuWeight

FROM Data\_CTE

GROUP BY Jaar, Maand, Artikelnummer, Artnaam

ORDER BY Jaar, Maand, Artikelnummer;

**Prognoses per product toegevoegd bij table:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

GemiddeldePrijs DECIMAL(18,3),

GemiddeldTuWeight DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity,

opb.priceKG AS PriceKG,

opb.tuWeight AS TuWeight

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

jaar,

maand,

artikel,

SUM(aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema]

GROUP BY jaar, maand, artikel

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

TotaalQuantity,

GemiddeldePrijs,

GemiddeldTuWeight,

PrognoseWaarde

)

SELECT

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam,

SUM(d.Quantity) AS TotaalQuantity,

AVG(d.PriceKG) AS GemiddeldePrijs,

AVG(d.TuWeight) AS GemiddeldTuWeight,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

LEFT JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam

ORDER BY

d.Jaar,

d.Maand,

d.Artikelnummer;

**test database met prognoses en prijs en gewicht verwijderd uit table**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.test', 'U') IS NOT NULL

DROP TABLE dbo.test;

CREATE TABLE dbo.test

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.veldt.prod].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.veldt.prod].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.veldt.prod].[dbo].[orderinfo] oi

INNER JOIN [4food.veldt.prod].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

jaar,

maand,

artikel,

SUM(aantalhe) AS PrognoseWaarde

FROM [4food.veldt.prod].[dbo].[jaarschema]

GROUP BY jaar, maand, artikel

)

INSERT INTO dbo.test

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam,

SUM(d.Quantity) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

LEFT JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam

ORDER BY

d.Jaar,

d.Maand,

d.Artikelnummer;

**Napoleon database met prognoses en prijs en gewicht verwijderd uit table**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

jaar,

maand,

artikel,

SUM(aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema]

GROUP BY jaar, maand, artikel

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam,

SUM(d.Quantity) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

LEFT JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam

ORDER BY

d.Jaar,

d.Maand,

d.Artikelnummer;

**Quary op productsoort op te halen:**

select s.id, g.\*

From specificaties s

inner join specificationProductGroup spg ON spg.specificationId = s.internalId

inner join groepen g on g.id = spg.groupID

**Napoleon database met productsoorten erbij:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

GroepNaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

g.groepnaam AS GroepNaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = opb.artikelnummer

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

jaar,

maand,

artikel,

SUM(aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema]

GROUP BY jaar, maand, artikel

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

GroepNaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam,

d.GroepNaam,

SUM(d.Quantity) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

LEFT JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

d.Jaar,

d.Maand,

d.Artikelnummer,

d.Artnaam,

d.GroepNaam

ORDER BY

d.Jaar,

d.Maand,

d.Artikelnummer;

**V1 van final Quary:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

GroepNaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

g.groepnaam AS GroepNaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = opb.artikelnummer

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

jaar,

maand,

artikel,

SUM(aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema]

GROUP BY jaar, maand, artikel

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

GroepNaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

COALESCE(d.Jaar, j.jaar) AS Jaar,

COALESCE(d.Maand, j.maand) AS Maand,

COALESCE(d.Artikelnummer, j.artikel) AS Artikelnummer,

COALESCE(d.Artnaam, '-') AS Artnaam,

COALESCE(d.GroepNaam, '-') AS GroepNaam,

SUM(ISNULL(d.Quantity, 0)) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

FULL OUTER JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

COALESCE(d.Jaar, j.jaar),

COALESCE(d.Maand, j.maand),

COALESCE(d.Artikelnummer, j.artikel),

COALESCE(d.Artnaam, '-'),

COALESCE(d.GroepNaam, '-')

ORDER BY

Jaar, Maand, Artikelnummer;

**V2 van final quary:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

Artnaam NVARCHAR(255),

GroepNaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

opb.artnaam AS Artnaam,

g.groepnaam AS GroepNaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = opb.artikelnummer

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

j.jaar,

j.maand,

j.artikel,

g.groepnaam,

SUM(j.aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema] j

INNER JOIN YearMonthCount ym

ON ym.Jaar = j.jaar

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = j.artikel

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

GROUP BY j.jaar, j.maand, j.artikel, g.groepnaam

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

Artnaam,

GroepNaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

COALESCE(d.Jaar, j.jaar) AS Jaar,

COALESCE(d.Maand, j.maand) AS Maand,

COALESCE(d.Artikelnummer, j.artikel) AS Artikelnummer,

d.Artnaam AS Artnaam, -- alleen uit realisatie

COALESCE(d.GroepNaam, j.groepnaam, '-') AS GroepNaam, -- eventueel uit prognose

SUM(ISNULL(d.Quantity, 0)) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

FULL OUTER JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

COALESCE(d.Jaar, j.jaar),

COALESCE(d.Maand, j.maand),

COALESCE(d.Artikelnummer, j.artikel),

d.Artnaam,

COALESCE(d.GroepNaam, j.groepnaam, '-')

ORDER BY

Jaar, Maand, Artikelnummer;

**Artnaam er uit gehaald quary afgerond:**

USE [4food.aidan];

GO

IF OBJECT\_ID('dbo.Napoleon\_All', 'U') IS NOT NULL

DROP TABLE dbo.Napoleon\_All;

CREATE TABLE dbo.Napoleon\_All

(

Jaar INT,

Maand INT,

Artikelnummer NVARCHAR(50),

GroepNaam NVARCHAR(255),

TotaalQuantity DECIMAL(18,3),

PrognoseWaarde DECIMAL(18,3)

);

;WITH YearMonthCount AS

(

SELECT

YEAR(deliveryDate) AS Jaar,

COUNT(DISTINCT MONTH(deliveryDate)) AS MonthsWithData

FROM [4food.napoleon.productie].[dbo].[orderinfo]

GROUP BY YEAR(deliveryDate)

HAVING COUNT(DISTINCT MONTH(deliveryDate)) >= 10

),

Quantiles AS

(

SELECT

PERCENTILE\_CONT(0.01) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q1\_quantity,

PERCENTILE\_CONT(0.99) WITHIN GROUP (ORDER BY opb.quantity) OVER () AS Q99\_quantity

FROM [4food.napoleon.productie].[dbo].[opb] opb

),

Data\_CTE AS

(

SELECT

YEAR(oi.deliveryDate) AS Jaar,

MONTH(oi.deliveryDate) AS Maand,

opb.artikelnummer AS Artikelnummer,

g.groepnaam AS GroepNaam,

CASE

WHEN oi.orderType = 18 AND opb.quantity > 0 THEN -opb.quantity

ELSE opb.quantity

END AS Quantity

FROM [4food.napoleon.productie].[dbo].[orderinfo] oi

INNER JOIN [4food.napoleon.productie].[dbo].[opb] opb

ON oi.id = opb.orderInfoId

INNER JOIN YearMonthCount ym

ON ym.Jaar = YEAR(oi.deliveryDate)

CROSS JOIN (SELECT DISTINCT Q1\_quantity, Q99\_quantity FROM Quantiles) q

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = opb.artikelnummer

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

WHERE oi.orderStatus <> 4

AND oi.orderType IN (2,3,4,18)

AND opb.quantity <> 0

AND opb.priceKG > 0.001

AND (opb.quantity < 0 OR (opb.quantity BETWEEN q.Q1\_quantity AND q.Q99\_quantity))

),

Jaarschema\_CTE AS

(

SELECT

j.jaar,

j.maand,

j.artikel,

g.groepnaam,

SUM(j.aantalhe) AS PrognoseWaarde

FROM [4food.napoleon.productie].[dbo].[jaarschema] j

INNER JOIN YearMonthCount ym

ON ym.Jaar = j.jaar

LEFT JOIN [4food.napoleon.productie].[dbo].[specificaties] s

ON s.id = j.artikel

LEFT JOIN [4food.napoleon.productie].[dbo].[specificationProductGroup] spg

ON spg.specificationId = s.internalId

LEFT JOIN [4food.napoleon.productie].[dbo].[groepen] g

ON g.id = spg.groupID

GROUP BY j.jaar, j.maand, j.artikel, g.groepnaam

)

INSERT INTO dbo.Napoleon\_All

(

Jaar,

Maand,

Artikelnummer,

GroepNaam,

TotaalQuantity,

PrognoseWaarde

)

SELECT

COALESCE(d.Jaar, j.jaar) AS Jaar,

COALESCE(d.Maand, j.maand) AS Maand,

COALESCE(d.Artikelnummer, j.artikel) AS Artikelnummer,

COALESCE(d.GroepNaam, j.groepnaam, '-') AS GroepNaam,

SUM(ISNULL(d.Quantity, 0)) AS TotaalQuantity,

SUM(ISNULL(j.PrognoseWaarde, 0)) AS PrognoseWaarde

FROM Data\_CTE d

FULL OUTER JOIN Jaarschema\_CTE j

ON d.Jaar = j.jaar

AND d.Maand = j.maand

AND d.Artikelnummer = j.artikel

GROUP BY

COALESCE(d.Jaar, j.jaar),

COALESCE(d.Maand, j.maand),

COALESCE(d.Artikelnummer, j.artikel),

COALESCE(d.GroepNaam, j.groepnaam, '-')

ORDER BY

Jaar, Maand, Artikelnummer;