VISTAS MEDORO FABRICA

--1vista inicial: ConCuboDesde2020

CREATE OR ALTER VIEW ConCuboDesde2020 AS

WITH DatosParseados AS (

SELECT \*,

TRY\_CAST(Inicio AS DATETIME) AS InicioDT,

TRY\_CAST(Fin AS DATETIME) AS FinDT

FROM ConCubo

WHERE

TRY\_CAST(Inicio AS DATETIME) >= '2020-01-01' AND

ISNUMERIC(SUBSTRING(ID, PATINDEX('%[0-9]%', ID), LEN(ID))) = 1

),

HorasCalculadas AS (

SELECT \*,

DATEDIFF(SECOND, InicioDT, FinDT) / 3600.0 AS Total\_Horas

FROM DatosParseados

)

SELECT

ID,

-- Clave limpia

TRY\_CAST(SUBSTRING(ID, PATINDEX('%[0-9]%', ID), LEN(ID)) AS INT) AS ID\_Limpio,

Renglon,

Estado,

-- Fechas corregidas (resta de 2 días)

DATEADD(DAY, -2, InicioDT) AS Inicio\_Corregido,

DATEADD(DAY, -2, FinDT) AS Fin\_Corregido,

-- Fechas legibles como texto

CONVERT(VARCHAR(16), DATEADD(DAY, -2, InicioDT), 120) AS Inicio\_Legible\_Texto,

CONVERT(VARCHAR(16), DATEADD(DAY, -2, FinDT), 120) AS Fin\_Legible\_Texto,

-- Fecha agrupada (solo día)

CONVERT(DATE, DATEADD(DAY, -2, InicioDT)) AS Fecha,

-- Duración total y por tipo de estado

Total\_Horas,

CASE WHEN Estado = 'Producción' THEN Total\_Horas ELSE 0 END AS Horas\_Produccion,

CASE WHEN Estado = 'Preparación' THEN Total\_Horas ELSE 0 END AS Horas\_Preparacion,

CASE WHEN Estado = 'Maquina Parada' THEN Total\_Horas ELSE 0 END AS Horas\_Parada,

CASE WHEN Estado = 'Mantenimiento' THEN Total\_Horas ELSE 0 END AS Horas\_Mantenimiento,

-- Producción buena y mala

TRY\_CAST(CantidadBuenosProducida AS FLOAT) AS CantidadBuenosProducida,

TRY\_CAST(CantidadMalosProducida AS FLOAT) AS CantidadMalosProducida,

-- Nuevas columnas solicitadas

Turno,

Maquinista,

Operario,

codproducto,

motivo

FROM HorasCalculadas;

--2vista segunda: ConCuboDesde2020Secuencias

CREATE OR ALTER VIEW VistaConCuboDesde2020Secuencias AS

WITH Base AS (

SELECT \*,

DATEDIFF(SECOND, Inicio\_Corregido, Fin\_Corregido) / 3600.0 AS Duracion\_Horas

FROM ConCuboDesde2020

)

SELECT

ID,

ID\_Limpio,

Renglon,

Estado,

Inicio\_Corregido,

Fin\_Corregido,

Inicio\_Legible\_Texto,

Fin\_Legible\_Texto,

CONVERT(DATE, Inicio\_Corregido) AS Fecha,

Duracion\_Horas AS Total\_Horas,

CASE WHEN Estado = 'Producción' THEN Duracion\_Horas ELSE 0 END AS Horas\_Produccion,

CASE WHEN Estado = 'Preparación' THEN Duracion\_Horas ELSE 0 END AS Horas\_Preparacion,

CASE WHEN Estado = 'Maquina Parada' THEN Duracion\_Horas ELSE 0 END AS Horas\_Parada,

CASE WHEN Estado = 'Mantenimiento' THEN Duracion\_Horas ELSE 0 END AS Horas\_Mantenimiento,

CantidadBuenosProducida,

CantidadMalosProducida,

-- ✅ Columnas nuevas pedidas

Turno,

Maquinista,

Operario,

codproducto,

Motivo

FROM Base;

--3vista: ConCuboDesde2020SecuenciasFlag

CREATE OR ALTER VIEW ConCuboDesde2020SecuenciasFlag AS

-- Primer CTE: agrega un número de secuencia por ID\_Limpio y máquina (Renglon)

WITH Base AS (

SELECT \*,

ROW\_NUMBER() OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido ASC

) AS Nro\_Secuencia

FROM VistaConCuboDesde2020Secuencias

),

-- Segundo CTE: detecta inicio de bloque de preparación

PrepFlag AS (

SELECT \*,

CASE

WHEN Estado = 'Preparación' AND (

LAG(Estado) OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido

) IS DISTINCT FROM 'Preparación'

) THEN 1

ELSE 0

END AS FlagPreparacion

FROM Base

),

-- Tercer CTE: genera la secuencia acumulativa de bloques de preparación

PrepSecuencia AS (

SELECT \*,

SUM(FlagPreparacion) OVER (

PARTITION BY ID\_Limpio, Renglon

ORDER BY Inicio\_Corregido

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

) AS SecuenciaPreparacion

FROM PrepFlag

)

-- Resultado final

SELECT

ID,

ID\_Limpio,

Renglon,

Estado,

Inicio\_Corregido,

Fin\_Corregido,

Inicio\_Legible\_Texto,

Fin\_Legible\_Texto,

Fecha,

Total\_Horas,

Horas\_Produccion,

Horas\_Preparacion,

Horas\_Parada,

Horas\_Mantenimiento,

CantidadBuenosProducida,

CantidadMalosProducida,

Turno,

Maquinista,

Operario,

CodProducto,

Motivo,

Nro\_Secuencia,

FlagPreparacion,

SecuenciaPreparacion

FROM PrepSecuencia;

--4vista final: ConCuboFinal

CREATE OR ALTER VIEW ConCuboFinal AS

SELECT

s.ID,

s.ID\_Limpio,

s.Renglon,

s.Estado,

s.Inicio\_Corregido,

s.Fin\_Corregido,

s.Inicio\_Legible\_Texto,

s.Fin\_Legible\_Texto,

s.Fecha,

-- ✅ Las 5 columnas adicionales reales

s.Turno,

s.Maquinista,

s.Operario,

s.CodProducto,

s.Motivo,

s.Total\_Horas,

s.Horas\_Produccion,

s.Horas\_Preparacion,

s.Horas\_Parada,

s.Horas\_Mantenimiento,

s.CantidadBuenosProducida,

s.CantidadMalosProducida,

s.Nro\_Secuencia,

s.FlagPreparacion,

s.SecuenciaPreparacion,

-- ✅ Solo esta columna viene del JOIN

VU.saccod1

FROM ConCuboDesde2020SecuenciasFlag s

LEFT JOIN TablaVinculadaUNION VU

ON ISNUMERIC(VU.OP) = 1

AND TRY\_CAST(VU.OP AS INT) = s.ID\_Limpio

WHERE YEAR(s.Inicio\_Corregido) >= 2020;

MEDIDAS

(decimal number,2)

**Medidas.%TiempoPreparacion** =

DIVIDE(

    SUM(ConCuboFinal[Horas\_Preparacion]),

    SUM(ConCuboFinal[Horas\_Produccion])

) \* 100

(general, auto)

**Medidas.Porcentaje\_Malos** =

DIVIDE(

    SUM(ConCuboFinal[CantidadMalosProducida]),

    SUM(ConCuboFinal[CantidadBuenosProducida]) +

    SUM(ConCuboFinal[CantidadMalosProducida])

)

(text)

**RatioBMTexto** =

VAR Buenos = SUM(ConCuboFinal[CantidadBuenosProducida])

VAR Malos = SUM(ConCuboFinal[CantidadMalosProducida])

RETURN

    SWITCH(TRUE(),

        Malos = 0 && Buenos > 0, "s/malos",

        Malos = 0 && Buenos = 0, "-",  // Nada producido

        FORMAT(DIVIDE(Buenos, Malos), "0.00")  // Resultado numérico como texto

    )

(text)

**SemaforoPreparación** =

VAR porcentaje = [Medidas.%TiempoPreparacion]

RETURN

SWITCH(

    TRUE(),

    porcentaje < 50, "🟢",

    porcentaje <= 100, "🟡",

    porcentaje > 100, "🔴",

    "❓"

)

(general)

**Unidades\_por\_Hora\_Prod** =

VAR TotalHorasProd = SUM(ConCuboFinal[Horas\_Produccion])

RETURN

    IF(

        TotalHorasProd > 0,

        ROUND(

            DIVIDE(

                SUM(ConCuboFinal[CantidadBuenosProducida]),

                TotalHorasProd

            ),

            2

        ),

        BLANK()

    )

A screenshot of a computer

AI-generated content may be incorrect.

TRANSFORM DATA – SOURCE \* Y CAMBIAR SERVER SQL, ELEGIR CONCUBOFINAL Y SE RECUPERA TODO, INCLUYENDO MEDIDAS.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.