

Gestión de Datos

Gestión de Flotas

2C 2021

Grupo Nro. 38 - SIN_NOMBRE

- SANCHEZ, TOMAS AGUSTIN	- 1660433
- ALVAREZ, LEANDRO	- 1468870
- CAPUTO, FEDERICO	- 1498289
- CACCIA CAMPANER, EMILIO NICOLAS	- 1669916

Contenido

Contenido	1
Modelo y Entidades	3
Migracion	4
Ordenes de Trabajo	4
Materiales	5
Viajes y Paquetes	6
Otras entidades	8
Modelo de BI	11
Correcciones	11
Dimensiones	13
Tiempo	13
Recorrido	13
Rango Edad	13
Choferes y Mecánicos	13
Camion	13
Modelo Camion	13
Marca Camion	13
Dimension Taller	13
Tipo Tarea	14
Hechos	14
Viajes	14
Ordenes de Trabajo	14
Vistas	15
Máximo Tiempo Fuera de Servicio	15
Costo Total Por Mantenimiento de Camion	16
Facturación por Recorrido	17
Costo por Choferes	19
Desvio de Tareas	20
Top 5 Tareas por Camión	21
Top 10 Materiales por Taller	24
Apendice	27
Diagrama Entidad Relación - BI V2	27
Diagrama Entidad Relación - BI V1	28
Diagrama Entidad Relación Transaccional - V3	28
Diagrama Entidad Relación Transaccional - V2	30
Diagrama Entidad Relación Transaccional - V1	31

Modelo y Entidades

De acuerdo a las correcciones entregadas durante la primera entrega, se modelo toda entidad que posea los componentes similares a ID y Descripción.

Segun revision:

Se deben modelar entidades como Ciudad, marca, modelo, estado, tipo y todas aquellas entidades donde se tenga una Id - descripción o id - cod - descripción deben modelarse.

Luego, el resto de entidades surgen de la especificación del caso de uso, sumado al relevamiento realizado por los alumnos sobre la tabla maestra.

Además se corrigió el modelo entre Paquetes y Viajes, correspondientes a la observación:

Un viaje posee varios paquetes, un paquete puede estar en más de un viaje (lo que va dentro del paquete es indistinto), además de que un paquete tiene su tipo predefinido. Por consiguiente es necesario tener una tabla que relacione los viajes y los paquetes (viaje_x_paquete) en donde se pueda indicar qué cantidad de estos se incluyen. (la cantidad de paquetes no define al viaje ni al paquete)

Modelando entonces, los Tipos de paquetes, junto con los Paquetes por Viajes para atender la relación de muchos a muchos que se da entre Paquetes y Viajes.

Se presenta el diagrama de entidad relación, en el [Apéndice](#), adjunta la versión de la primera entrega, junto a su corrección.

Migracion

Ordenes de Trabajo

En cuanto a la migración de Órdenes de Trabajo, en adelante OTs, y Tareas, aceptamos la sugerencia de la corrección.

Un mecánico podría estar asignado a un taller en vez de que la OT tenga el taller. La OT es ejecutada por el mecánico no por el taller... (seguerencia)

Por lo cual asumimos que una OT, y todas sus tareas son ejecutadas por el mismo mecánico, por lo cual, para el llenado de la tabla, se tomó cada orden según el camión, la fecha, y el mecánico para la identificación unívoca dentro de la tabla Maestra.

En primer lugar, el estado de la orden de trabajo fue muy sencillo de obtener, por lo cual no vamos a explayar en sus detalles

```
]INSERT INTO [SIN_NOMBRE].ORDEN_TRABAJO
SELECT DISTINCT
    M.ORDEN_TRABAJO_FECHA,
    M.CAMION_PATENTE,
    M.MECANICO_NRO_LEGAJO,
    EOT.Id_Estado
FROM [gd_esquema].Maestra M
JOIN [SIN_NOMBRE].ESTADO_ORDEN_TRABAJO EOT ON EOT.Descripcion = M.ORDEN_TRABAJO_ESTADO
WHERE ORDEN_TRABAJO_ESTADO IS NOT NULL
GROUP BY M.ORDEN_TRABAJO_FECHA,
         M.CAMION_PATENTE,
         M.MECANICO_NRO_LEGAJO,
         EOT.Id_Estado
.
```

De manera similar, las tareas por OT se identifican en la tabla maestra, según la fecha, el mecánico y el camión

```

INSERT INTO [SIN_NOMBRE].TAREA_POR_ORDEN
SELECT DISTINCT
    M.TAREA_FECHA_INICIO_PLANIFICADO,
    M.TAREA_FECHA_INICIO,
    M.TAREA_FECHA_FIN,
    M.TAREA_CODIGO,
    OT.Nro_OT
FROM [gd_esquema].Maestra M
JOIN [SIN_NOMBRE].ORDEN_TRABAJO OT ON
    OT.Fecha_Creacion      = M.ORDEN_TRABAJO_FECHA
    AND OT.Legajo_Mecanico = M.MECANICO_NRO_LEGAJO
    AND OT.Patente_Camion  = M.CAMION_PATENTE

```

Materiales

Armar un maestro de materiales no genero complicación alguna

```

INSERT INTO SIN_NOMBRE.MATERIAL
SELECT DISTINCT [MATERIAL_COD], [MATERIAL_DESCRIPCION], [MATERIAL_PRECIO]
FROM gd_esquema.Maestra
WHERE MATERIAL_COD IS NOT NULL

```

Sin embargo, al ordenar los materiales que se utilizarían por cada tarea nos encontrábamos con que no existían en la tabla Maestra datos modelados, como la cantidad y unidad de materiales a utilizar en cada tarea. Por lo cual consideramos cargarlas con valores NULL.

```

INSERT INTO [SIN_NOMBRE].MATERIAL_POR_TAREA
SELECT DISTINCT
    M.MATERIAL_COD,
    M.TAREA_CODIGO,
    NULL
FROM [gd_esquema].Maestra M
WHERE M.TAREA_CODIGO IS NOT NULL
      AND M.MATERIAL_COD IS NOT NULL
ORDER BY M.TAREA_CODIGO

```

Viajes y Paquetes

Para identificar los viajes, utilizamos la combinación de las fechas tanto inicio como fin, el consumo de combustible, el recorrido y su precio, el chofer y la patente del camión, por lo cual asumimos que existe solo un chofer y camión por viaje.

```
INSERT INTO [SIN_NOMBRE].VIAJE
SELECT DISTINCT
    M.VIAJE_FECHA_INICIO,
    M.VIAJE_FECHA_FIN,
    M.VIAJE_CONSUMO_COMBUSTIBLE,
    M.RECORRIDO_PRECIO,
    R.Codigo,
    M.CHOFER_NRO_LEGAJO,
    M.CAMION_PATENTE
FROM [gd_esquema].Maestra M
JOIN [SIN_NOMBRE].CIUDAD C ON C.Nombre = M.RECORRIDO_CIUADAD_DESTINO
JOIN [SIN_NOMBRE].CIUDAD C2 ON C2.Nombre = M.RECORRIDO_CIUADAD_ORIGEN
JOIN [SIN_NOMBRE].RECORRIDO R ON
    R.Ciudad_Origen = C2.Id_Ciudad
    AND R.Ciudad_Destino = C.Id_Ciudad
    AND R.KM = M.RECORRIDO_KM
WHERE VIAJE_CONSUMO_COMBUSTIBLE IS NOT NULL
```

De igual manera se identifican los paquetes, como no hay datos sobre las dimensiones reales de los paquetes, asumimos que son del máximo de su categoría.

A su vez, como una vez cargados en la tabla de paquetes, nos resulta imposible identificarlos unívocamente cual paquete corresponde a cada viaje, optamos por una estrategia de tabla temporal en memoria, y a medida que se carga un paquete, también cargar la tabla de paquetes por viaje.

```

SELECT DISTINCT
    TP.Codigo,
    M.PAQUETE_PESO_MAX,
    M.PAQUETE_ALTO_MAX,
    M.PAQUETE_ANCHO_MAX,
    M.PAQUETE_LARGO_MAX,
    M.PAQUETE_CANTIDAD,
    V.Id AS [Nro de Viaje]
FROM [gd_esquema].Maestra M
JOIN [SIN_NOMBRE].TIPO_PAQUETE TP ON TP.Descripcion = M.PAQUETE_DESCRIPCION
JOIN [SIN_NOMBRE].CIUDAD C ON C.Nombre = M.RECORRIDO_CIUADAD_DESTINO
JOIN [SIN_NOMBRE].CIUDAD C2 ON C2.Nombre = M.RECORRIDO_CIUADAD_ORIGEN
JOIN [SIN_NOMBRE].RECORRIDO R ON
    R.Ciudad_Origen = C2.Id_Ciudad
    AND R.Ciudad_Destino = C.Id_Ciudad
    AND R.KM = M.RECORRIDO_KM
JOIN [SIN_NOMBRE].VIAJE V ON
    V.Cod_Recorrido = R.Codigo
    AND V.Patente_Camion = M.CAMION_PATENTE
    AND V.Legajo_Chofer = M.CHOFER_NRO_LEGajo
    AND V.Fecha_Fin = M.VIAJE_FECHA_FIN
    AND V.Fecha_Inicio = M.VIAJE_FECHA_INICIO
ORDER BY V.Id
SET @RowsToProcess = @@ROWCOUNT
SET @CurrentRow = 0
WHILE @CurrentRow < @RowsToProcess
BEGIN
    -- Paso al proximo indice
    SET @CurrentRow = @CurrentRow + 1
    -- Primero guardo los Paquetes, para instanciar un Paquete.ID
    INSERT INTO [SIN_NOMBRE].PAQUETE
    SELECT
        PKG.Codigo,
        PKG.Peso,
        PKG.Alto,
        PKG.Ancho,
        PKG.Largo
    FROM @LT_TEMP_PKG PKG
    WHERE PKG.RowID = @CurrentRow
    -- El ID va a ser el mismo que el nro de fila, al ser la carga.
    INSERT INTO [SIN_NOMBRE].PAQUETE_POR_VIAJE
    SELECT
        PKG.Viaje,
        PKG.RowID,
        PKG.Cantidad
    FROM @LT_TEMP_PKG PKG
    WHERE PKG.RowID = @CurrentRow
END

```

Otras entidades

El resto de las entidades no poseen la suficiente complejidad para entrar en detalle sobre cada una, sin embargo a modo ilustrativo, se adjuntan las cargas realizadas

Algunos modelos poseen el mismo nombre, por lo cual es necesario realizar la búsqueda con igualdad de velocidades y capacidades.

```
INSERT INTO [SIN_NOMBRE].CAMION
SELECT DISTINCT
M.CAMION_PATENTE,
M.CAMION_NRO_CHASIS,
M.CAMION_NRO_MOTOR,
M.CAMION_FECHA_ALTA,
MC.Marca_Id,
MC.Modelo_Id
FROM [gd_esquema].Maestra M
JOIN [SIN_NOMBRE].MODELO_CAMION MC ON MC.Descripcion = M.MODELO_CAMION
AND MC.Velocidad_Max = M.MODELO_VELOCIDAD_MAX
AND MC.Capacidad_Tanque = M.MODELO_CAPACIDAD_TANQUE
AND MC.Capacidad_Carga = M.MODELO_CAPACIDAD_CARGA
ORDER BY CAMION_FECHA_ALTA
```



```

INSERT INTO SIN_NOMBRE.TIPO_PAQUETE
SELECT DISTINCT [PAQUETE_DESCRIPCION],
                [PAQUETE_PESO_MAX],
                [PAQUETE_ALTO_MAX],
                [PAQUETE_ANCHO_MAX],
                [PAQUETE_LARGO_MAX],
                [PAQUETE_PRECIO]
FROM gd_esquema.Maestra
WHERE PAQUETE_CANTIDAD IS NOT NULL

```

```

INSERT INTO SIN_NOMBRE.MARCA_CAMION
SELECT DISTINCT MARCA_CAMION_MARCA
FROM gd_esquema.Maestra
WHERE MARCA_CAMION_MARCA IS NOT NULL

```

```

INSERT INTO SIN_NOMBRE.MODELO_CAMION
SELECT DISTINCT MC.Id,
                MODELO_CAMION,
                MODELO_VELOCIDAD_MAX,
                MODELO_CAPACIDAD_TANQUE,
                MODELO_CAPACIDAD_CARGA
FROM gd_esquema.Maestra
JOIN SIN_NOMBRE.MARCA_CAMION MC ON MARCA_CAMION_MARCA = MC.Descripcion
WHERE MODELO_CAMION IS NOT NULL
ORDER BY 2

```

Correcciones

De acuerdo con nuestra corrección de la entrega anterior:

La entrega de Modelo de datos y migración se encuentra aprobada, salvoooooooooooooooooo que deben corregir para la entrega de BI como relacionan las OT con las tareas_x_OT, debiese haber muchos menos OT, ya que las mismas pueden contener varias tareas_x_OT. Uds han armado casi una relación 1 a 1

Corregimos la base de datos, donde ahora cada Tarea X Ot tiene al mecánico quedando el nuevo diagrama (ver apéndice V3)

Modelo de BI

Correcciones

DER BI - El diseño del modelo BI no es correcto:

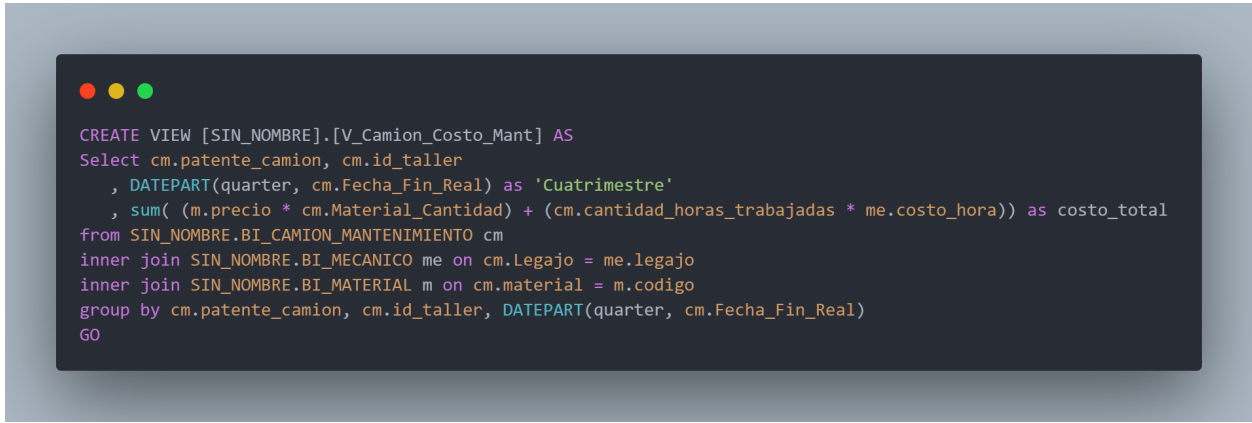
No determina claramente cuáles son las tablas de hechos y dimensiones. corregir para la entrega de BI

Si bien el diagrama estaba mal diseñado, agregamos el prefijo DIM y HECHO al diagrama, además de cambiar el color y su disposición en semejanza a lo visto en clases.

En las supuestas tablas de hechos no hay valores pre-calculados producto de las agrupaciones (costo de viajes, facturación, tiempo fuera servicio, etc.) lo cual facilita la resolución de las consultas a la hora de responder a una necesidad de negocio. (las que se plantean en el enunciado)

Entendimos como facilita la creación de las vistas, pasamos de obtener los cálculos en la vista, a realizar un tratamiento previo:

Anteriormente, realizamos cálculos de más véase:

A screenshot of a code editor with a dark background and light-colored text. The code is a SQL query to create a view named [V_Camion_Costo_Mant]. It uses a SELECT statement with columns cm.patente_camion, cm.id_taller, DATEPART(quarter, cm.Fecha_Fin_Real) as 'Cuatrimestre', and a calculated column sum((m.precio * cm.Material_Cantidad) + (cm.cantidad_horas_trabajadas * me.costos_hora)) as costo_total. The FROM clause is SIN_NOMBRE.BI_CAMION_MANTENIMIENTO cm. There are two inner joins: SIN_NOMBRE.BI_MECANICO me on cm.Legajo = me.legajo and SIN_NOMBRE.BI_MATERIAL m on cm.material = m.codigo. The query ends with a GROUP BY clause and a GO command.

```
CREATE VIEW [SIN_NOMBRE].[V_Camion_Costo_Mant] AS
Select cm.patente_camion, cm.id_taller
      , DATEPART(quarter, cm.Fecha_Fin_Real) as 'Cuatrimestre'
      , sum( (m.precio * cm.Material_Cantidad) + (cm.cantidad_horas_trabajadas * me.costos_hora)) as costo_total
from SIN_NOMBRE.BI_CAMION_MANTENIMIENTO cm
inner join SIN_NOMBRE.BI_MECANICO me on cm.Legajo = me.legajo
inner join SIN_NOMBRE.BI_MATERIAL m on cm.material = m.codigo
group by cm.patente_camion, cm.id_taller, DATEPART(quarter, cm.Fecha_Fin_Real)
GO
```

Donde ahora se sumaliza al cargar la tabla de hechos, fragmento de un SP para la carga de hechos de órdenes de trabajo.

```
DECLARE @c_h_ot CURSOR
SET @c_h_ot = CURSOR FOR
SELECT
    T.Id
    ,TxO.Cod_Tarea
    ,CAST(OT.Patente_Camion as NVARCHAR(15)) AS [Patente]
    ,C.Modelo_Id
    ,C.Marca_Id
    ,TL.Id
    ,SUM(TT.Costo) AS [Costo]
    ,SUM(ABS(DATEDIFF(DAY, TxO.Fecha_Inicio_Real, TxO.Fecha_Fin_Real) - TT.Tiempo_Estimado)) AS [Desvío]
    ,SUM(DATEDIFF(DAY, OT.Fecha_Creacion, TxO.Fecha_Fin_Real)) AS [Dias_Sin_Servicio]
FROM [SIN_NOMBRE].ORDEN_TRABAJO OT
JOIN [SIN_NOMBRE].CAMION C ON C.Patente = OT.Patente_Camion
JOIN [SIN_NOMBRE].BI_TIEMPO T ON T.Anio = YEAR(OT.Fecha_Creacion)
AND DATEPART(QUARTER,OT.Fecha_Creacion) = T.Cuatrimestre
JOIN [SIN_NOMBRE].TAREA_POR_ORDEN TxO ON TxO.Nro_OT = OT.Nro_OT
JOIN [SIN_NOMBRE].BI_TIPO_TAREA TT ON TT.Codigo = TxO.Cod_Tarea
LEFT JOIN [SIN_NOMBRE].MECANICO M ON M.Legajo = TxO.Mecanico
LEFT JOIN [SIN_NOMBRE].TALLER TL ON TL.Id = M.Id_taller
WHERE TxO.Cod_Tarea IN (1,2,3,4,5,6)
GROUP BY T.Id, TxO.Cod_Tarea, TL.Id, OT.Patente_Camion, C.Modelo_Id, C.Marca_Id
ORDER BY 1, 3
```

Se recomienda que repasen las teoría a fin de afinar algunos conceptos, porque básicamente lo que entregado es un pasaje de un modelo relacional a uno BI sin ningún tratamiento previo.

Sumado al punto aclarado anteriormente, los datos son tratados para confeccionar las tablas de hechos. Además añadimos la dimensión del tiempo

```
INSERT INTO [SIN_NOMBRE].[BI_HECHO_VIAJE]
SELECT
    T.Id AS [Tiempo]
    ,CAST(V.Patente_Camion as NVARCHAR(15)) AS [Patente]
    ,V.Cod_Recorrido AS [Recorrido]
    ,SUM(V.Precio_Final + [SIN_NOMBRE].FN_VALOR_VIAJE(V.Id)) AS [Total_Facturado]
    -- 1 LT Combustible = $100 + Asumimos que trabaja todos los días 8 horas.
    ,SUM(V.Consumo_Combustible * 100 + C.Costo_Hora * DATEDIFF(DAY, V.Fecha_Inicio, V.Fecha_Fin) * 8) AS [Costo_Total]
FROM [SIN_NOMBRE].VIAJE V
JOIN [SIN_NOMBRE].BI_TIEMPO T ON T.Anio = YEAR(V.Fecha_Fin)
AND DATEPART(QUARTER,V.Fecha_Fin) = T.Cuatrimestre
JOIN [SIN_NOMBRE].CHOFER C ON C.Legajo = V.Legajo_Chofier
GROUP BY T.Id, V.Patente_Camion, V.Cod_Recorrido
ORDER BY V.Patente_Camion, T.Id, Recorrido
GO
```

Dimensiones

Tiempo

Según la corrección, añadimos esta dimensión siendo su carga a través de un stored procedure.

Recorrido

Dimensiona los recorridos con sus respectivas ciudades y costos.

Rango Edad

Esta era una dimensión faltante al igual que el tiempo,

Choferes y Mecánicos

Ahora se valen de la dimensión rango edad

Camion

No sufre ningún cambio

Modelo Camion

No sufre ningún cambio

Marca Camion

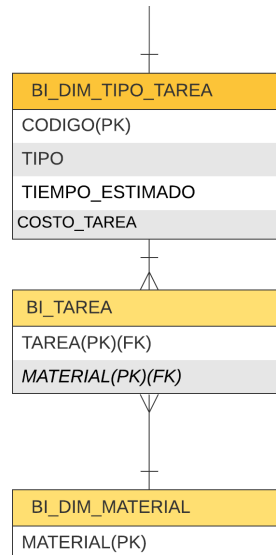
No sufre ningún cambio

Dimension Taller

No sufre ningún cambio

Tipo Tarea

Optamos por aplanar la tabla de tipo de tarea con tarea, sin embargo al necesitar el top de materiales, nos resultó necesario quebrar un poco el modelo estrella, para poder obtener los materiales por tarea, según consultado en clases nuestra dimensión de Tipo Tareas estará compuesta por además, tareas y materiales.



Hechos

Viajes

Abandonamos la tabla **Camión Viaje**, ahora se sumarán los costos de combustible y de paquetes por Recorrido por camión por cuatrimestre.

Ordenes de Trabajo

Nuevamente descartamos la anterior tabla de hechos **Camión Mantenimiento**, de manera similar a viajes, se sumarán por cuatrimestre las tareas realizadas en el mismo taller por camión tanto costos como promedio de desvíos.

Vistas

Máximo Tiempo Fuera de Servicio

Anteriormente:

```
CREATE VIEW [SIN_NOMBRE].[V_Camion_Maximo_Tiempo_FueraDeServicio] AS
select cm.Patente_Camion
      , DATEPART(quarter, cm.Fecha_Creacion_OT) as 'Cuatrimestre'
      , max(DATEDIFF(DAY, cm.Fecha_Creacion_OT, cm.Fecha_Fin_OT)) as 'Maxima_Duracion_OT (Dias)'
from SIN_NOMBRE.BI_CAMION_MANTENIMIENTO cm
group by cm.Patente_Camion, DATEPART(quarter, cm.Fecha_Creacion_OT);
GO
```

Con la nueva revisión

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Camion_Maximo_Tiempo_FueraDeServicio] AS
SELECT
    T.Anio
    ,T.Cuatrimestre
    ,OT.Patente AS [Camion]
    ,MAX(OT.Horas_Sin_Servicio) AS [Sin_Servicio]
FROM [SIN_NOMBRE].[BI_HECHO_ORDEN_TRABAJO] OT
JOIN [SIN_NOMBRE].[BI_TIEMPO] T ON T.Id = OT.Tiempo
GROUP BY T.Anio, T.Cuatrimestre, OT.Patente
GO
```

Costo Total Por Mantenimiento de Camion

De igual manera con los costos de mantenimiento, anteriormente:

```
CREATE VIEW [SIN_NOMBRE].[V_Camion_Costo_Mant] AS
Select cm.patente_camion, cm.id_taller
      , DATEPART(quarter, cm.Fecha_Fin_Real) as 'Cuatrimestre'
      , sum( (m.precio * cm.Material_Cantidad) + (cm.cantidad_horas_trabajadas * me.costos_hora)) as costo_total
from SIN_NOMBRE.BI_CAMION_MANTENIMIENTO cm
inner join SIN_NOMBRE.BI_MECANICO me on cm.Legajo = me.legajo
inner join SIN_NOMBRE.BI_MATERIAL m on cm.material = m.codigo
group by cm.patente_camion, cm.id_taller, DATEPART(quarter, cm.Fecha_Fin_Real)
GO
```

Actualizamos a

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Camion_Costo_Mant]
AS
SELECT
    T.Anio
    ,T.Cuatrimestre
    ,OT.Patente
    ,SUM(OT.Costo) AS [Costo Total Cuatrimestral]
FROM [SIN_NOMBRE].[BI_HECHO_ORDEN_TRABAJO] OT
JOIN [SIN_NOMBRE].[BI_TIEMPO] T ON T.Id = OT.Tiempo
GROUP BY T.Anio, T.Cuatrimestre, OT.Patente
GO
```


Facturación por Recorrido

Anteriormente:

```
CREATE VIEW [SIN_NOMBRE].[V_Facturacion_Recorrido_Cuatrimestre] AS
SELECT r.Codigo as 'Recorrido'
      , r.Origem as 'Ciudad_Origen'
      , r.Destino as 'Ciudad_Destino'
      , DATEPART(QUARTER, Fecha_Inicio) as 'Cuatrimestre'
      , SUM(Precio_Final) as 'Facturacion_Total'
FROM SIN_NOMBRE.BI_CAMION_VIAJE cv
JOIN SIN_NOMBRE.BI_RECORRIDO r on cv.Recorrido = r.Codigo
GROUP BY r.Codigo, r.Origem, r.Destino, DATEPART(QUARTER, Fecha_Inicio);
GO
```

Simplifica a

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Facturacion_Recorrido_Cuatrimestre] AS
SELECT
    T.Anio
  , T.Cuatrimestre
  , V.Recorrido
  , R.Origem AS [Ciudad_Origen]
  , R.Destino AS [Ciudad_Destino]
  , SUM(V.Total_Facturado) AS [Facturacion_Cuatrimestral]
FROM [SIN_NOMBRE].BI_HECHO_VIAJE V
JOIN SIN_NOMBRE.BI_RECORRIDO R on V.Recorrido = R.Codigo
JOIN [SIN_NOMBRE].[BI_TIEMPO] T ON T.Id = V.Tiempo
GROUP BY T.Anio, T.Cuatrimestre, V.Recorrido, R.Origem, R.Destino
GO
```

Ganancias por Camión

Anteriormente

```
CREATE VIEW [SIN_NOMBRE].[V_Camion_Ganancias] AS
select cm.Patente_Camion
      , SUM(cv.Precio_Final) as 'Ingresos'
      , SUM(ch.Costo_Hora * (DATEDIFF(HOUR, cv.Fecha_Inicio, cv.Fecha_Fin)) + 100 * cv.Consumo_Combustible) as
'Costo_Viaje'
      , SUM(cm.Material_Cantidad * mt.Precio + cm.Cantidad_Horas_Trabajadas * mec.Costo_Hora) as
'Costo_Mantenimiento'
      , SUM(cv.Precio_Final)
      -
      SUM(ch.Costo_Hora * (DATEDIFF(HOUR, cv.Fecha_Inicio, cv.Fecha_Fin)) + 100 * cv.Consumo_Combustible)
      -
      SUM(cm.Material_Cantidad * mt.Precio + cm.Cantidad_Horas_Trabajadas * mec.Costo_Hora)
      as 'Ganancia'
from SIN_NOMBRE.BI_CAMION_VIAJE cv
join SIN_NOMBRE.BI_CAMION_MANTENIMIENTO cm on cv.Patente = cm.Patente_Camion
join SIN_NOMBRE.BI_CHOFER ch on cv.Legajo_Chofer = ch.Legajo
join SIN_NOMBRE.BI_MATERIAL mt on cm.Material = mt.Codigo
join SIN_NOMBRE.BI_MECANICO mec on cm.Legajo = mec.Legajo
group by cm.Patente_Camion;
GO
```

Fue necesaria utilizar una función

```
CREATE OR ALTER FUNCTION
[SIN_NOMBRE].FN_COSTO_MANT(@patente NVARCHAR(15))
RETURNS INT
AS
BEGIN
    RETURN (
        SELECT
            SUM(OT.Costo) AS [Costo_Mantenimiento]
        FROM [SIN_NOMBRE].BI_HECHO_ORDEN_TRABAJO OT
        WHERE OT.Patente = @patente
    )
END
GO

CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Camion_Ganancias]
AS
SELECT
    V.Patente
    , SUM(V.Total_Facturado - V.Costo_Total - [SIN_NOMBRE].FN_COSTO_MANT(V.Patente)) AS [Ganancia]
FROM [SIN_NOMBRE].BI_HECHO_VIAJE V
GROUP BY V.Patente
GO
```

Costo por Choferes

Anteriormente:

```
CREATE VIEW [SIN_NOMBRE].[V_Promedio_Costo_Chofer]
AS SELECT
    CASE WHEN Edad BETWEEN 18 AND 30 THEN '18 - 30'
         WHEN Edad BETWEEN 31 AND 50 THEN '31 - 50'
         ELSE '> 50' END AS [Rango_Etario]
    ,AVG(Costo_Hora) AS [Promedio]
FROM [SIN_NOMBRE].[BI_CHOFER]
GROUP BY CASE WHEN Edad BETWEEN 18 AND 30 THEN '18 - 30'
         WHEN Edad BETWEEN 31 AND 50 THEN '31 - 50'
         ELSE '> 50' END
GO
```

Realizando el tratamiento previo en el modelo de BI simplifica a

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Promedio_Costo_Chofer]
AS
SELECT
    E.Descripcion
    ,AVG(Costo_Hora) AS [Promedio]
FROM [SIN_NOMBRE].[BI_CHOFER] C
JOIN [SIN_NOMBRE].[BI_RANGO_EDAD] E ON C.Rango_Edad = E.Id
GROUP BY E.Descripcion
GO
```

Desvio de Tareas

Anteriormente:

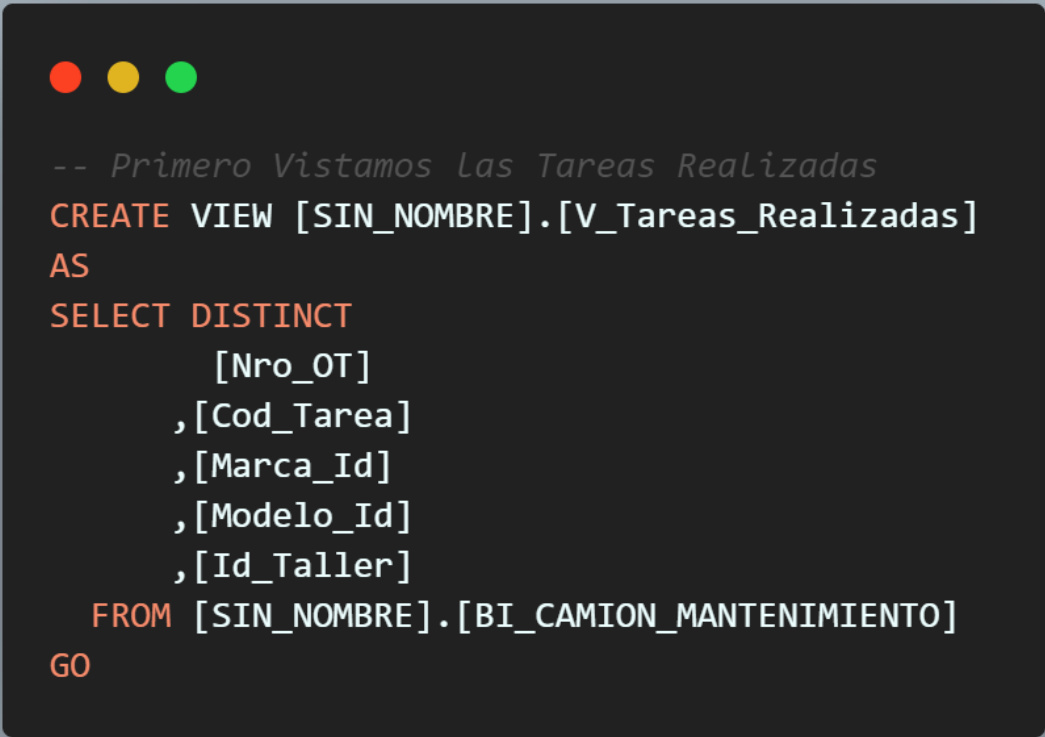
```
CREATE VIEW [SIN_NOMBRE].[V_Desvio_Tareas]
AS
SELECT DISTINCT
    [Id_taller] AS [Taller]
    ,[Cod_Tarea] AS [Tarea]
    ,AVG ( T.Tiempo_Estimado - [Cantidad_Horas_Trabajadas]/8) AS [Desvio]
FROM (SELECT DISTINCT
    CM2.Nro_Ot,
    CM2.Cod_Tarea,
    CM2.Id_taller,
    CM2.Cantidad_Horas_Trabajadas
    FROM [SIN_NOMBRE].[BI_CAMION_MANTENIMIENTO] CM2) CM
JOIN [SIN_NOMBRE].[BI_TAREA] T ON T.Codigo = CM.Cod_Tarea
GROUP BY CM.Id_taller, CM.Cod_Tarea
ORDER BY CM.Id_taller, CM.Cod_Tarea
GO
```

Se actualizo a:

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Desvio_Tareas]
AS
SELECT DISTINCT
    OT.[Id_taller] AS [Taller]
    ,OT.[Tarea] AS [Tarea]
    ,AVG(Desvio_Tarea) AS [Promedio Desvio]
FROM [SIN_NOMBRE].[BI_HECHO_ORDEN_TRABAJO] OT
GROUP BY OT.Id_taller, OT.Tarea
GO
```

Top 5 Tareas por Camión

Anteriormente fueron necesarias dos vistas más para poder lograr el top 5. Si bien Camión Mantenimiento facilitó algunas vistas anteriores, para este requerimiento en particular resultó una complicación.



```
-- Primero Vistamos Las Tareas Realizadas
CREATE VIEW [SIN_NOMBRE].[V_Tareas_Realizadas]
AS
SELECT DISTINCT
    [Nro_OT]
    ,[Cod_Tarea]
    ,[Marca_Id]
    ,[Modelo_Id]
    ,[Id_Taller]
FROM [SIN_NOMBRE].[BI_CAMION_MANTENIMIENTO]
GO
```



-- Despues Las Tareas Por Camion

```
CREATE VIEW [SIN_NOMBRE].[V_Tareas_Por_Camion]
AS
SELECT [Cod_Tarea]
      ,[Marca_Id]
      ,[Modelo_Id]
      ,ISNULL(COUNT(*), 0) AS [Ejecutadas]
FROM [SIN_NOMBRE].[V_Tareas_Realizadas]
GROUP BY [Cod_Tarea], [Marca_Id], [Modelo_Id]
GO
```



-- Finalmente Los Camiones con Sus respectivas Tareas

```
CREATE VIEW [SIN_NOMBRE].[V_TOP_5_Tareas_Por_Camion]
AS
SELECT DISTINCT
      T.[Marca_ID]
      ,T.[Modelo_ID]
      ,T.[Cod_Tarea]
      ,TE.Ejecutadas AS [Veces Realizas]
FROM [SIN_NOMBRE].[V_Tareas_Realizadas] T
JOIN [SIN_NOMBRE].[V_Tareas_Por_Camion] TE ON
      TE.Modelo_Id = T.Modelo_Id AND TE.Cod_Tarea = T.Cod_Tarea
      AND TE.Marca_Id = T.Marca_Id
WHERE T.Cod_Tarea IN (SELECT TOP 5 Cod_Tarea
                      FROM [SIN_NOMBRE].[V_Tareas_Por_Camion] TxC
                      WHERE T.Marca_Id = TxC.Marca_Id
                      AND T.Modelo_Id = TxC.Modelo_Id
                      ORDER BY Ejecutadas DESC)
GO
```

Con el nuevo modelo BI esto no fue necesario para nada,

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_TOP_5_Tareas_Por_Camion]
AS
SELECT DISTINCT
    OT.Marca
    ,OT.Modelo
    ,OT.Tarea
FROM [SIN_NOMBRE].BI_HECHO_ORDEN_TRABAJO OT
WHERE OT.Tarea IN (
    SELECT TOP 5
        OT2.Tarea
    FROM [SIN_NOMBRE].BI_HECHO_ORDEN_TRABAJO OT2
    WHERE OT2.Marca = OT.Marca AND OT2.Modelo = OT.Modelo
    GROUP BY OT2.Tarea
    ORDER BY COUNT(OT2.Tarea) DESC
)
GO
```

Top 10 Materiales por Taller

El mismo caso se repite en este top, antes cuando se utilizaba Camión Mantenimiento optamos por utilizar dos vistas:

Materiales por taller



```
CREATE VIEW [SIN_NOMBRE].[V_Materiales_Taller]
AS
SELECT
    [Id_taller]
    ,[Material]
    ,SUM([Material_Cantidad]) AS [Veces_Utilizado]
FROM [GD2C2021].[SIN_NOMBRE].[BI_CAMION_MANTENIMIENTO]
GROUP BY Id_taller, Material
GO
```


Finalmente, el Top 10

```
CREATE VIEW [SIN_NOMBRE].[V_Materiales_Taller]
AS
SELECT
    [Id_taller]
    ,[Material]
    ,SUM([Material_Cantidad]) AS [Veces_Utilizado]
FROM [GD2C2021].[SIN_NOMBRE].[BI_CAMION_MANTENIMIENTO]
GROUP BY Id_taller, Material
GO

CREATE VIEW [SIN_NOMBRE].[V_Top_10_Materiales_Taller]
AS
SELECT DISTINCT
    CM.[Id_taller] AS [Taller]
    ,CM.[Material]
    ,VMT.Veces_Utilizado
FROM [SIN_NOMBRE].[BI_CAMION_MANTENIMIENTO] CM
JOIN [SIN_NOMBRE].[V_Materiales_Taller] VMT ON VMT.Id_taller = CM.Id_taller
AND VMT.Material = CM.Material
WHERE CM.Material IN (SELECT TOP 10
    Material
    FROM [SIN_NOMBRE].[V_Materiales_Taller] MT
    WHERE MT.Id_taller = CM.Id_taller
    ORDER BY MT.[Veces_Utilizado] DESC)
GO
```

En cambio, el nuevo TOP 10 no requiere otra vista.

```
CREATE OR ALTER VIEW [SIN_NOMBRE].[V_Top_10_Materiales_Taller]
AS
SELECT DISTINCT
    OT.Id_taller AS [Taller]
    ,M.Material AS [Material]
FROM [SIN_NOMBRE].BI_HECHO_ORDEN_TRABAJO OT
INNER JOIN [SIN_NOMBRE].BI_TAREA T ON T.Tarea = OT.Tarea
INNER JOIN [SIN_NOMBRE].BI_MATERIAL M ON M.Material = T.Material
WHERE M.Material IN (
    SELECT TOP 10
        T2.Material
    FROM [SIN_NOMBRE].BI_HECHO_ORDEN_TRABAJO OT2
    INNER JOIN [SIN_NOMBRE].BI_TAREA T2 ON T2.Tarea = OT2.Tarea
    INNER JOIN [SIN_NOMBRE].BI_MATERIAL M2 ON M2.Material = T2.Material
    WHERE OT2.Id_taller = OT.Id_taller
    GROUP BY T2.Material
    ORDER BY COUNT(T2.Material) DESC
)
GO
```

Apendice

Diagrama Entidad Relación - BI V2

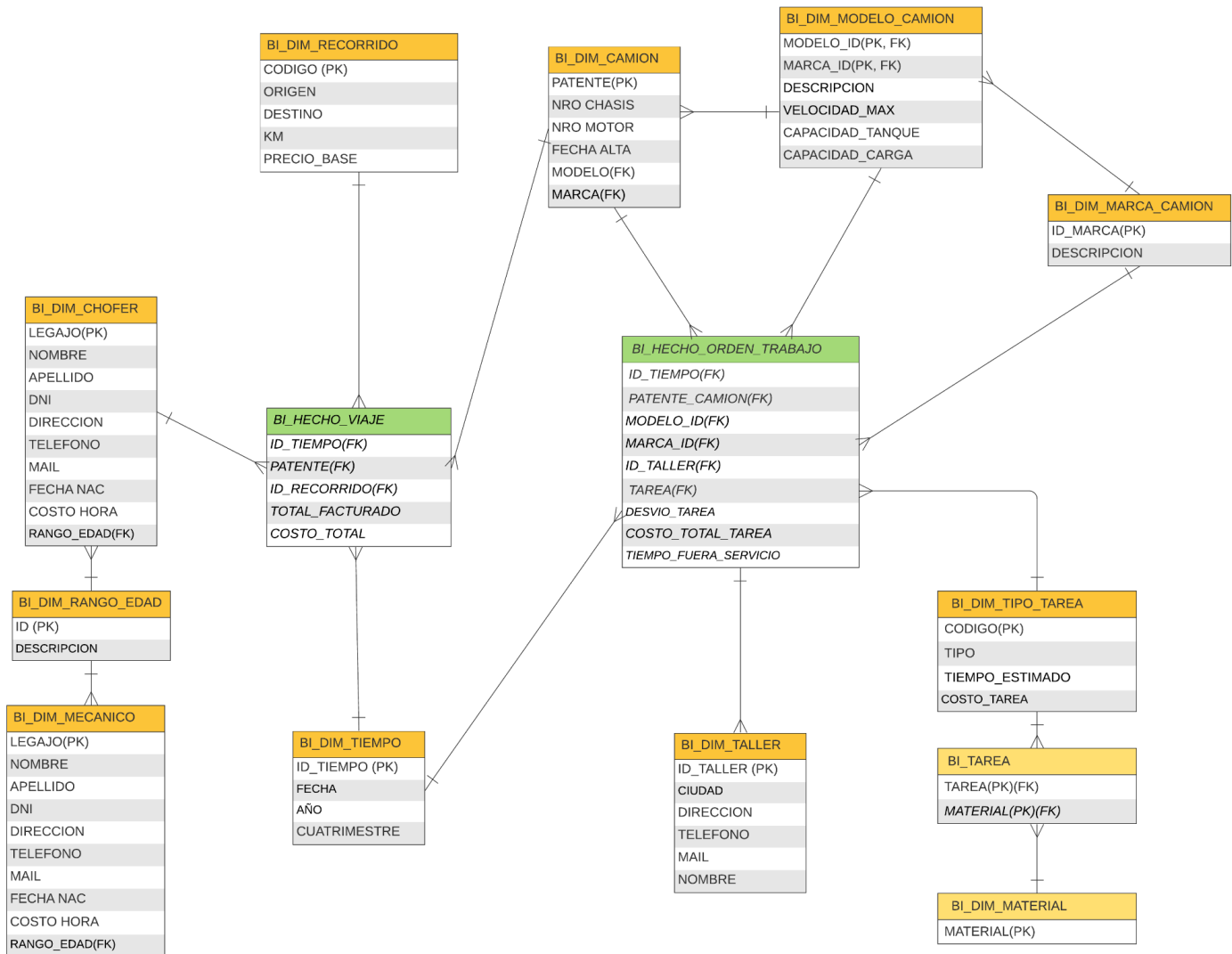


Diagrama Entidad Relación - BI V1

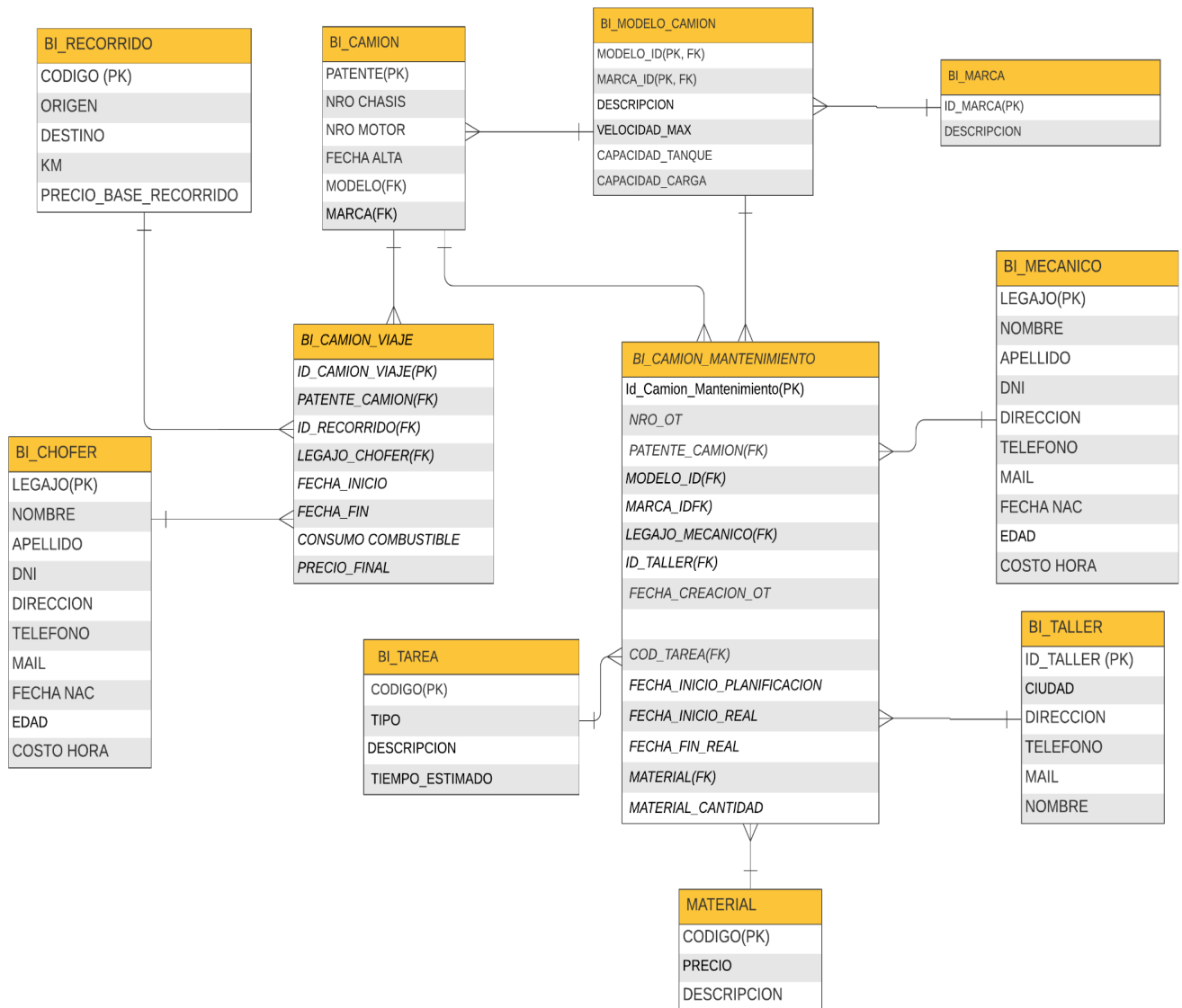


Diagrama Entidad Relación Transaccional - V3

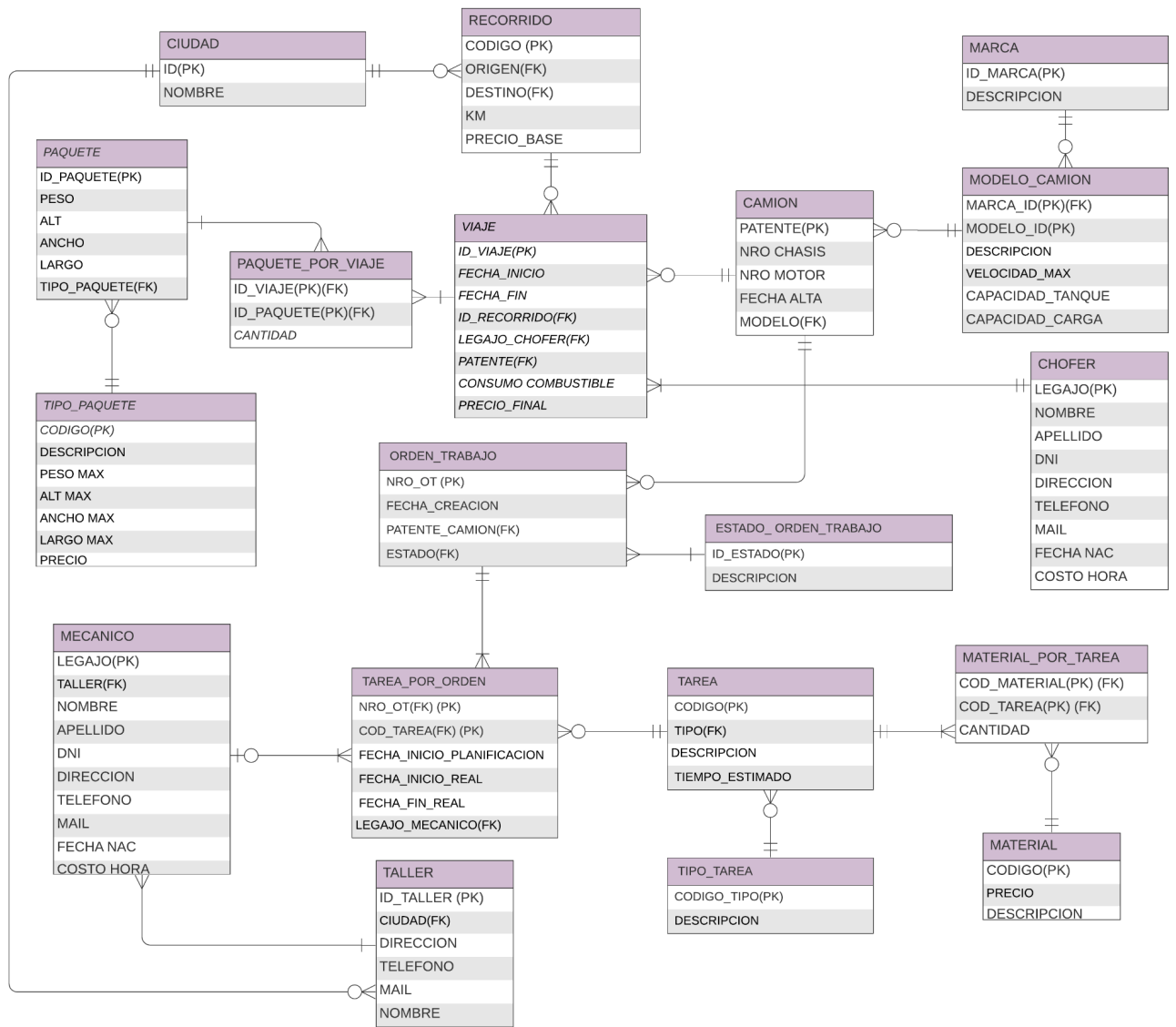


Diagrama Entidad Relación Transaccional - V2

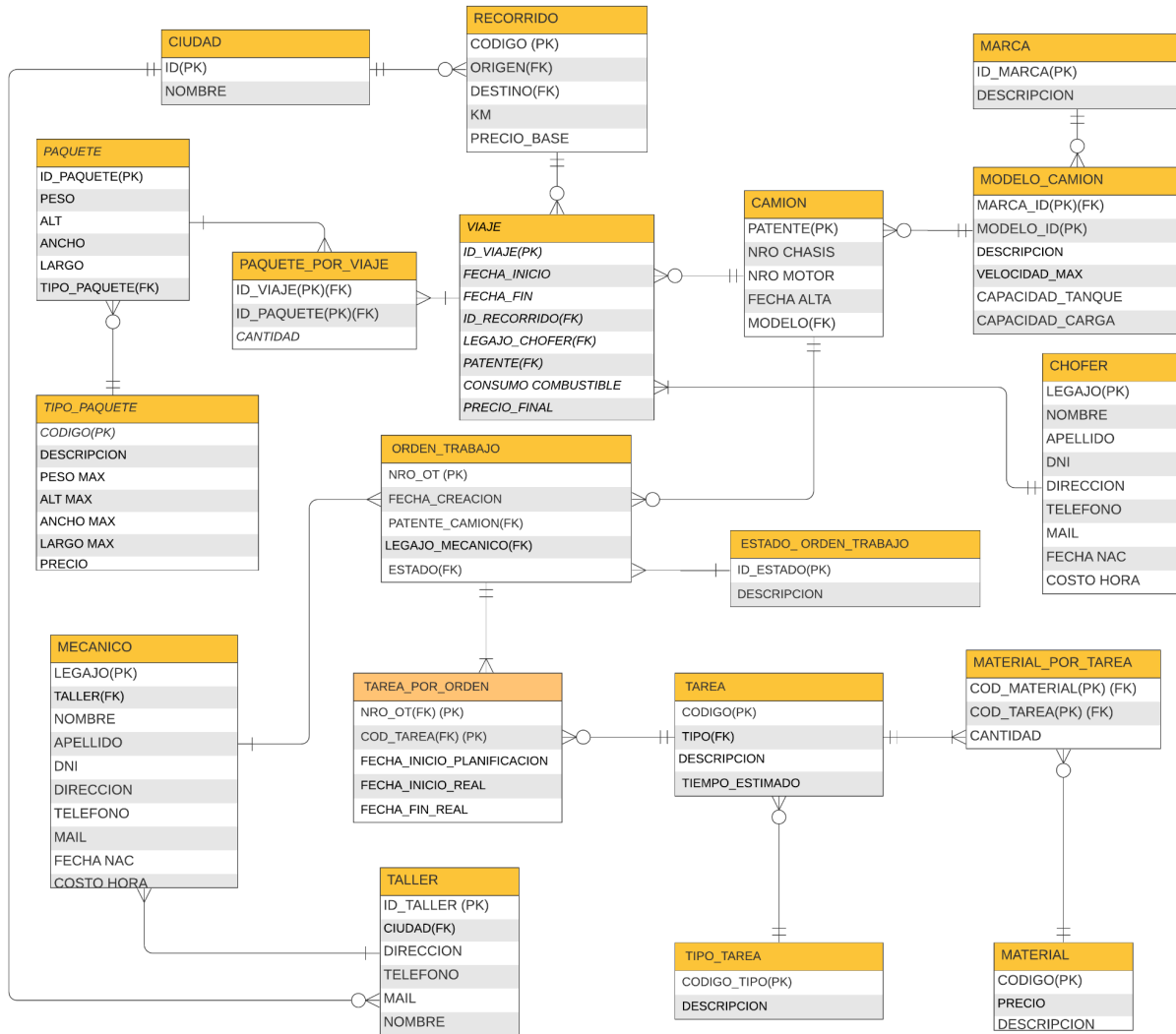


Diagrama Entidad Relación Transaccional - V1

