

Documento de Estrategia

Gestión de Datos 1C 2022

Grupo 18: LOCRO

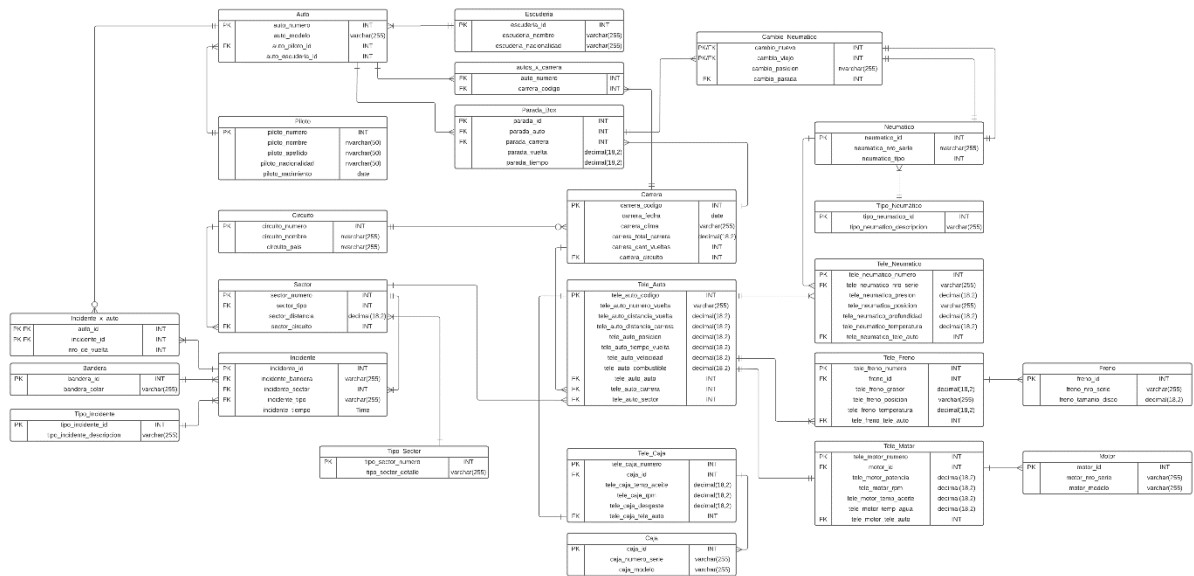
Integrantes:

- **Thiago Cabrera Lavezzi**

Contents

DER:	3
Escudería:	3
Piloto:	3
Auto:	4
Circuito:	4
Tipo_Sector:.....	5
Sector:.....	5
Carrera:.....	6
Bandera:	6
Tipo_Incidente:.....	7
Incidente:.....	7
Tele_Auto:	8
Caja:	9
Tele_Caja:	9
Motor:.....	10
Tele_Motor:.....	10
Freno:.....	11
Tele_Freno:.....	11
Tipo_Neumatico:	12
Neumatico:	13
Tele_Neumatico:.....	15
Incidente_Auto:	16
Auto_Carrera:	17
Parada_box:.....	17
Cambio_Neumatico:	18
BI.....	20

DER:



Armado de tablas:

Escudería:

Son los “equipos” a los que pertenecerán autos. A cada escudería puede pertenecer un máximo de 2 autos. Nos interesa guardar las siguientes columnas:

- ESCUDERIA_ID (autoincremental)
- ESCUDERIA_NOMBRE
- ESCUDERIA_NACIONALIDAD

Para hacer eso, definimos el procedure que se encargará de migrar las escuderías de la tabla maestra a la nueva tabla tendrá que sacar los diferentes nombres y la nacionalidad del respectivo nombre. El ID será un auto incremental, por lo que no debemos preocuparnos de eso. Al utilizar el nombre de la escudería como ID, filtraremos a los que tengan NULL en ese campo.

```
CREATE PROCEDURE migrar_escuderia
AS
BEGIN
    INSERT INTO LOCRO.ESCUDERIA (ESCUDERIA_NOMBRE, ESCUDERIA_NACIONALIDAD)
    SELECT DISTINCT ESCUDERIA_NOMBRE, ESCUDERIA_NACIONALIDAD
    FROM gd_esquema.Maestra
    WHERE ESCUDERIA_NOMBRE IS NOT NULL
END
GO
```

Piloto:

Son los que corren en sus respectivos autos.

Columnas:

- PILOTO_ID (autoincremental)
- PILOTO_NOMBRE

- PILOTO_APELLIDO
- PILOTO_NACIONALIDAD
- PILOTO_NACIMIENTO

Simplemente seleccionamos los diferentes campos del piloto, mientras que el nombre del piloto sea NOT NULL.

```
CREATE PROCEDURE migrar_piloto
AS
BEGIN
    INSERT INTO LOCRO.PILOTO (PILOTO_NOMBRE, PILOTO_APELLIDO, PILOTO_NACIONALIDAD,
PILOTO_NACIMIENTO)
    SELECT DISTINCT PILOTO_NOMBRE, PILOTO_APELLIDO, PILOTO_NACIONALIDAD,
PILOTO_FECHA_NACIMIENTO
    FROM gd_esquema.Maestra
    WHERE PILOTO_NOMBRE IS NOT NULL
END
GO
```

Auto:

Cada auto estará corriendo carreras y pertenece a un piloto en específico y a una escudería.

Columnas:

- AUTO_ID (autoincremental)
- AUTO_NUMERO
- AUTO_MODELO
- AUTO_PILOTO_ID
- AUTO_ESCUDERIA_ID

Para migrar los autos, debemos primero haber migrado los pilotos y las escuderías, ya que los autos tienen foreign keys que apuntan hacia dichas tablas.

Para esto insertaremos los distintos autos con dos JOIN: uno a los pilotos, a los que identificaremos por nombre y apellido, y otro a las escuderías, a las que identificaremos por nombre. El filtro que pondremos será que el numero de auto debe ser diferente de NULL.

```
CREATE PROCEDURE migrar_auto
AS
BEGIN
    INSERT INTO LOCRO.AUTO (AUTO_NUMERO, AUTO_MODELO, AUTO_PILOTO, AUTO_ESCUDERIA)
    SELECT DISTINCT AUTO_NUMERO, AUTO_MODELO, p.PILOTO_ID, e.ESCUDERIA_ID
    FROM gd_esquema.Maestra m
    JOIN LOCRO.PILOTO p ON m.PILOTO_NOMBRE = p.PILOTO_NOMBRE AND m.PILOTO_APELLIDO =
p.PILOTO_APELLIDO
    JOIN LOCRO.ESCUDERIA e ON m.ESCUDERIA_NOMBRE = e.ESCUDERIA_NOMBRE
    WHERE m.AUTO_NUMERO IS NOT NULL
END
GO
```

Circuito:

Es el lugar en el que se llevará a cabo una carrera.

Columnas:

- CIRCUITO_CODIGO
- CIRCUITO_NOMBRE
- CIRCUITO_PAIS

Para migrar esto basta con seleccionar los distintos campos que componen a un circuito, asegurándonos de que el código no sea null.

```
CREATE PROCEDURE migrar_circuito
AS
BEGIN
    INSERT INTO LOCRO.CIRCUITO (CIRCUITO_CODIGO, CIRCUITO_NOMBRE, CIRCUITO_PAIS)
    SELECT DISTINCT CIRCUITO_CODIGO, CIRCUITO_NOMBRE, CIRCUITO_PAIS
    FROM gd_esquema.Maestra
    WHERE CIRCUITO_CODIGO IS NOT NULL
END
GO
```

Tipo_Sector:

Esta tabla almacenará los diferentes tipos de sectores.

Columnas:

- TIPO_SECTOR_NUMERO (autoincremental)
- TIPO_SECTOR_DESCRIPCION

Para migrar esta tabla basta con seleccionar todos los distintos tipos (menos el null).

```
CREATE PROCEDURE migrar_tipo_sector
AS
BEGIN
    INSERT INTO LOCRO.TIPO_SECTOR (TIPO_SECTOR_DETALLE)
    SELECT DISTINCT SECTO_TIPO
    FROM gd_esquema.Maestra
    WHERE SECTO_TIPO IS NOT NULL
END
GO
```

Sector:

Como su nombre lo indica, es un sector de un circuito en particular.

Columnas:

- SECTOR_CODIGO
- SECTOR_TIPO
- SECTOR_DISTANCIA
- SECTOR_CIRCUITO

Esta tabla posee una foreign key hacia la tabla de circuitos, por lo que deberemos, además de seleccionar los diferentes campos del sector, seleccionar el código del circuito que se encuentre en dicha columna, que será la que nos interesará guardar. También posee una foreign key hacia la tabla de tipo de sector y, como la clave primaria de dicha tabla fue generada por migración (es decir, no podemos obtenerla de la tabla maestra) tendremos que hacer un JOIN.

Para que todo esto funcione debemos asegurarnos de que el código del sector no sea nulo.

```
CREATE PROCEDURE migrar_sector
AS
BEGIN
    INSERT INTO LOCRO.SECTOR (SECTOR_CODIGO, SECTOR_TIPO, SECTOR_DISTANCIA, SECTOR_CIRCUITO)
    SELECT DISTINCT CODIGO_SECTOR, TIPO_SECTOR_NUMERO, SECTOR_DISTANCIA, CIRCUITO_CODIGO
    FROM gd_esquema.Maestra
    JOIN LOCRO.TIPO_SECTOR
        ON TIPO_SECTOR_DETALLE = SECTOR_TIPO
    WHERE CODIGO_SECTOR IS NOT NULL
END
GO
```

Carrera:

Es el evento en sí, que transcurre en un circuito en específico, en una fecha en particular, con un clima, cantidad de vueltas, etc.

Columnas

- CARRERA_CODIGO
- CARRERA_FECHA
- CARRERA_CLIMA
- CARRERA_TOTAL_CARRERA
- CARRERA_CANT_VUELTAS
- CARRERA_CIRCUITO

El campo total carrera se refiere a la cantidad total de metros que supone dicha carrera.

Para migrar la carrera debemos también seleccionar el circuito en el que se realiza, asegurándonos de que el código de la carrera no sea nulo.

```
CREATE PROCEDURE migrar_carrera
AS
BEGIN
    INSERT INTO LOCRO.CARRERA (CARRERA_CODIGO, CARRERA_FECHA, CARRERA_CLIMA, CARRERA_TOTAL_CARRERA, CARRERA_CANT_VUELTAS, CARRERA_CIRCUITO)
    SELECT DISTINCT CODIGO_CARRERA, CARRERA_FECHA, CARRERA_CLIMA, CARRERA_TOTAL_CARRERA, CARRERA_CANT_VUELTAS, CIRCUITO_CODIGO
    FROM gd_esquema.Maestra
    WHERE CODIGO_CARRERA IS NOT NULL
END
GO
```

Bandera:

Columnas:

- BANDERA_ID (autoincremental)
- BANDERA_COLOR

Para realizar esta migración muy sencilla basta con hacer un SELECT DISTINCT al campo incidente bandera de la tabla maestra.

```

CREATE PROCEDURE migrar_bandera
AS
BEGIN
    INSERT INTO LOCRO.BANDERA (BANDERA_COLOR)
    SELECT DISTINCT INCIDENTE_BANDERA
    FROM gd_esquema.Maestra
    WHERE INCIDENTE_BANDERA IS NOT NULL
END
GO

```

Tipo_Incidente:

Columnas:

- TIPO_INCIDENTE_ID (autoincremental)
- TIPO_INCIDENTE_DESCRIPCION

Para migrar esta tabla basta con hacer un SELECT DISTINCT al campo del tipo de incidente.

```

CREATE PROCEDURE migrar_incidente_tipo
AS
BEGIN
    INSERT INTO LOCRO.TIPO_INCIDENTE(TIPO_INCIDENTE_DESCRIPCION)
    SELECT DISTINCT INCIDENTE_TIPO
    FROM gd_esquema.Maestra
    WHERE INCIDENTE_TIPO IS NOT NULL
END
GO

```

Incidente:

En esta tabla no guardaremos los autos involucrados porque la cantidad de ellos puede ser variable. Guardaremos su id, la bandera, el tipo de incidente, el sector en donde ocurrió y el tiempo en el que sucedió.

Columnas:

- INCIDENTE_ID (autoincremental)
- INCIDENTE_BANDERA
- INCIDENTE_SECTOR
- INCIDENTE_TIPO
- INCIDENTE_TIEMPO

Al migrar incidentes, seleccionaremos los campos que conforman a un incidente y haremos dos JOIN; uno con las banderas por el color, tomando su id, y otro con los tipos de incidente por la descripción, tomando también su id. Para filtrar, utilizamos el campo TIPO_INCIDENTE, ya que no puede ser NULL en un incidente cualquiera.

```

CREATE PROCEDURE migrar_incidente
AS
BEGIN
    INSERT INTO LOCRO.INCIDENTE(INCIDENTE_BANDERA, INCIDENTE_SECTOR, INCIDENTE_TIPO, INCIDENTE_TIEMPO)
    SELECT DISTINCT b.BANDERA_ID, CODIGO_SECTOR, t.TIPO_INCIDENTE_ID, INCIDENTE_TIEMPO
    FROM gd_esquema.Maestra m
    JOIN LOCRO.BANDERA b ON b.BANDERA_COLOR = m.INCIDENTE_BANDERA
    JOIN LOCRO.TIPO_INCIDENTE t ON t.TIPO_INCIDENTE_DESCRIPCION = m.INCIDENTE_TIPO
    WHERE INCIDENTE_TIPO IS NOT NULL
END
GO

```

Tele_Auto:

En esta tabla guardaremos todos los valores referentes a la telemetría de un auto en general, sin meternos en los detalles de los componentes que aparecen en la tabla. También guardaremos datos como el tiempo de vuelta, el número, el sector, la carrera, etc.

Columnas:

- TELE_AUTO_CODIGO
- TELE_AUTO_NRO_VUELTA
- TELE_AUTO_DISTANCIA_VUELTA
- TELE_AUTO_DISTANCIA_CARRERA
- TELE_AUTO_POSICION
- TELE_AUTO_TIEMPO_VUELTA
- TELE_AUTO_VELOCIDAD
- TELE_AUTO_COMBUSTIBLE
- TELE_AUTO_AUTO
- TELE_AUTO_CARRERA
- TELE_AUTO_SECTOR

Para migrar, seleccionaremos los diferentes campos que componen a la tabla siempre y cuando el código de la telemetría no sea NULL. Además, realizaremos un JOIN a la tabla de autos por el modelo y el número, ya que detectamos que se repetían algunos números de autos.


```

CREATE PROCEDURE migrar_tele_auto
AS
BEGIN
    INSERT INTO LOCRO.TELE_AUTO(TELE_AUTO_CODIGO, TELE_AUTO_NRO_VUELTA,
    TELE_AUTO_DISTANCIA_VUELTA, TELE_AUTO_DISTANCIA_CARRERA,
    TELE_AUTO_POSICION, TELE_AUTO_TIEMPO_VUELTA, TELE_AUTO_VELOCIDAD,
    TELE_AUTO_COMBUSTIBLE, TELE_AUTO_AUTO, TELE_AUTO_CARRERA,
    TELE_AUTO_SECTOR)
    SELECT DISTINCT TELE_AUTO_CODIGO, TELE_AUTO_NUMERO_VUELTA,
    TELE_AUTO_DISTANCIA_VUELTA, TELE_AUTO_DISTANCIA_CARRERA,
    TELE_AUTO_POSICION, TELE_AUTO_TIEMPO_VUELTA, TELE_AUTO_VELOCIDAD,
    TELE_AUTO_COMBUSTIBLE, a.AUTO_ID, CODIGO_CARRERA, CODIGO_SECTOR
    FROM gd_esquema.Maestra m
    JOIN LOCRO.AUTO a ON a.AUTO_MODELO = m.AUTO_MODELO and a.AUTO_NUMERO =
    m.AUTO_NUMERO
    WHERE TELE_AUTO_CODIGO IS NOT NULL
END
GO

```

Caja:

En este campo no migraremos telemetrías, sino los diferentes motores que se pueden presentar en las telemetrías que se migrarán. De esta manera no repetimos en cada telemetría los campos que permanecen invariables.

Columnas:

- CAJA_ID (autoincremental)
- CAJA_NRO_SERIE
- CAJA_MODELO

Para migrar la tabla, debemos seleccionar el número de serie de la caja y el modelo, sacándolos directamente de las telemetrías (siempre que no sean NULL).

```

CREATE PROCEDURE migrar_caja
AS
BEGIN
    INSERT INTO LOCRO.CAJA(CAJA_NRO_SERIE, CAJA_MODELO)
    SELECT DISTINCT TELE_CAJA_NRO_SERIE, TELE_CAJA_MODELO
    FROM gd_esquema.Maestra m
    WHERE TELE_CAJA_NRO_SERIE IS NOT NULL AND TELE_CAJA_MODELO IS NOT NULL
END
GO

```

Tele_Caja:

Este campo corresponde a la telemetría de las cajas. Guardaremos datos que irán variando con las diferentes telemetrías que se tomen de las cajas, dejando los datos constantes como el número de serie o el modelo para su respectiva tabla, y vinculando aquellos datos con una foreign key. De esta manera evitamos la repetición de datos en esta tabla.

Columnas:

- TELE_CAJA_NUMERO (autoincremental)
- TELE_CAJA_CAJA

- TELE_CAJA_TEMP_ACEITE
- TEMP_CAJA_RPM
- TELE_CAJA_DESGASTE
- TELE_CAJA_TELE_AUTO

Para migrar, seleccionaremos todos los campos correspondientes a las telemetrías de cajas menos los mencionados en el párrafo anterior, vinculándolos realizando un JOIN a la tabla de cajas por número de serie. Por último, debemos

```
CREATE PROCEDURE migrar_tele_caja
AS
BEGIN
    INSERT INTO LOCRO.TELE_CAJA (TELE_CAJA_CAJA, TELE_CAJA_TEMP_ACEITE, TEMP_CAJA_RPM,
    TELE_CAJA_DESGASTE, TELE_CAJA_TELE_AUTO)
    SELECT DISTINCT c.CAJA_ID, TELE_CAJA_TEMP_ACEITE, TELE_CAJA_RPM,
    TELE_CAJA_DESGASTE, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m
    JOIN LOCRO.CAJA c ON c.CAJA_NRO_SERIE = m.TELE_CAJA_NRO_SERIE
    WHERE TELE_CAJA_NRO_SERIE IS NOT NULL AND TELE_CAJA_MODELO IS NOT NULL
END
GO
```

Motor:

Columnas:

- MOTOR_ID (autoincremental)
- MOTOR_NRO_SERIE
- MOTOR_MODELO

De la misma manera que dividimos las telemetrías de las cajas, lo haremos con los motores.

Para migrar:

```
CREATE PROCEDURE migrar_motor
AS
BEGIN
    INSERT INTO LOCRO.MOTOR (MOTOR_MODELO, MOTOR_NRO_SERIE)
    SELECT DISTINCT TELE_MOTOR_MODELO, TELE_MOTOR_NRO_SERIE
    FROM gd_esquema.Maestra m
    WHERE TELE_MOTOR_NRO_SERIE IS NOT NULL AND TELE_MOTOR_MODELO IS NOT NULL
END
GO
```

Tele_Motor:

En esta tabla guardaremos todos los datos vinculados a la telemetría de motores, vinculando el motor en específico con una foreign key a la tabla de motores.

Columnas:

- TELE_MOTOR_NUMERO (autoincrement)

- TELE_MOTOR_MOTOR
- TELE_MOTOR_POTENCIA
- TELE_MOTOR_RPM
- TELE_MOTOR_TEMP_ACEITE
- TELE_MOTOR_TEMP_AGUA
- TELE_MOTOR_TELE_AUTO

Para migrar:

```
CREATE PROCEDURE migrar_tele_motor
AS
BEGIN
    INSERT INTO LOCRO.TELE_MOTOR(TELE_MOTOR_MOTOR, TELE_MOTOR_TEMP_ACEITE,
    TELE_MOTOR_RPM, TELE_MOTOR_POTENCIA, TELE_MOTOR_TEMP_AGUA, TELE_MOTOR_TELE_AUTO)
    SELECT DISTINCT mo.MOTOR_ID, TELE_MOTOR_TEMP_ACEITE, TELE_MOTOR_RPM,
    TELE_MOTOR_POTENCIA, TELE_MOTOR_TEMP_AGUA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m
    JOIN LOCRO.MOTOR mo ON mo.MOTOR_NRO_SERIE = m.TELE_MOTOR_NRO_SERIE
    WHERE TELE_MOTOR_NRO_SERIE IS NOT NULL AND TELE_MOTOR_MODELO IS NOT NULL
END
GO
```

Freno:

Esta tabla tiene la misma lógica que las tablas de motores y cajas.

Columnas:

- FRENO_ID (autoincremental)
- FRENO_NRO_SERIE
- FRENO_TAMANIO_DISCO

Migracion:

```
CREATE PROCEDURE migrar_freno
AS
BEGIN
    INSERT INTO LOCRO.FRENO (FRENO_NRO_SERIE, FRENO_TAMANIO_DISCO)
    SELECT DISTINCT TELE_FRENO1_NRO_SERIE, TELE_FRENO1_TAMANIO_DISCO
    FROM gd_esquema.Maestra m
    WHERE TELE_FRENO1_NRO_SERIE IS NOT NULL AND TELE_FRENO1_TAMANIO_DISCO IS NOT NULL
END
GO
```

Tele_Freno:

Columnas:

- TELE_FRENO_NUMERO (autoincremental)
- TELE_FRENO_FRENO
- TELE_FRENO_GROSOR
- TELE_FRENO_POSICION
- TELE_FRENO_TEMPERATURA

- TELE_FRENO_TELE_AUTO

Cabe aclarar que como el código de la telemetría lo sacamos de la tabla maestra, no es necesario realizar el JOIN. Lo que si es necesario es utilizar UNIONS, porque hay 4 frenos en cada telemetría, y a nosotros nos interesa ponerlos en filas distintas, representando cada una el estado de uno de los frenos en el instante dado.

```
CREATE PROCEDURE migrar_tele_freno
AS
BEGIN
    INSERT INTO LOCRO.TELE_FRENO (TELE_FRENO_FRENO, TELE_FRENO_GROSOR, TELE_FRENO_POSI-
CION, TELE_FRENO_TEMPERATURA, TELE_FRENO_TELE_AUTO)
    SELECT DISTINCT f.FRENO_ID, TELE_FRENO1_GROSOR_PASTILLA, TELE_FRENO1_POSICION,
    TELE_FRENO1_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.FRENO f ON f.FRENO_NRO_SERIE =
m.TELE_FRENO1_NRO_SERIE
    WHERE TELE_FRENO1_NRO_SERIE IS NOT NULL AND TELE_FRENO1_TAMANIO_DISCO IS NOT NULL
    UNION
    SELECT DISTINCT f.FRENO_ID, TELE_FRENO2_GROSOR_PASTILLA, TELE_FRENO2_POSICION,
    TELE_FRENO2_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.FRENO f ON f.FRENO_NRO_SERIE =
m.TELE_FRENO2_NRO_SERIE
    WHERE TELE_FRENO2_NRO_SERIE IS NOT NULL AND TELE_FRENO2_TAMANIO_DISCO IS NOT NULL
    UNION
    SELECT DISTINCT f.FRENO_ID, TELE_FRENO3_GROSOR_PASTILLA, TELE_FRENO3_POSICION,
    TELE_FRENO3_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.FRENO f ON f.FRENO_NRO_SERIE =
m.TELE_FRENO3_NRO_SERIE
    WHERE TELE_FRENO3_NRO_SERIE IS NOT NULL AND TELE_FRENO3_TAMANIO_DISCO IS NOT NULL
    UNION
    SELECT DISTINCT f.FRENO_ID, TELE_FRENO4_GROSOR_PASTILLA, TELE_FRENO4_POSICION,
    TELE_FRENO4_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.FRENO f ON f.FRENO_NRO_SERIE =
m.TELE_FRENO4_NRO_SERIE
    WHERE TELE_FRENO4_NRO_SERIE IS NOT NULL AND TELE_FRENO4_TAMANIO_DISCO IS NOT NULL
END
GO
```

Tipo_Neumatico:

Guardaremos los tipos de neumáticos. Aclaramos que los registros de las telemetrías de autos no cuentan con el campo de tipo, por lo que solo pudimos conseguir el tipo de neumático de los neumáticos que fueron cambiados en una parada de boxes.

Columnas:

- TIPO_NEUMATICO_ID (autoincremental)
- TIPO_NEUMATICO_DESCRIPCION

Con esta tabla llegamos a un problema: todos los neumáticos deben tener un tipo, pero solo los que pasaron por una parada de box lo tienen registrado en la tabla maestra. Para solucionar esta problemática decidimos que al migrar los tipos de neumático le sumemos uno a los registrados en el sistema, uno que represente a los neumáticos sin tipo, por lo que luego de realizar la consulta SELECT para el primer INSERT, hacemos un segundo INSERT que define un nuevo tipo de neumático, el tipo SIN TIPO.

Como dijimos antes, esta tabla la sacamos de los campos de neumáticos nuevos y viejos. Al ser 4 neumáticos, debemos realizar 7 UNIONS.

```
CREATE PROCEDURE migrar_tipo_neumatico
AS
BEGIN
    INSERT INTO LOCRO.TIPO_NEUMATICO(TIPO_NEUMATICO_DESCRIPCION)
    SELECT NEUMATICO1_TIPO_NUEVO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO1_TIPO_NUEVO IS NOT NULL
    union
    SELECT NEUMATICO2_TIPO_NUEVO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO2_TIPO_NUEVO IS NOT NULL
    union
    SELECT NEUMATICO3_TIPO_NUEVO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO3_TIPO_NUEVO IS NOT NULL
    union
    SELECT NEUMATICO4_TIPO_NUEVO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO4_TIPO_NUEVO IS NOT NULL
    union
    SELECT NEUMATICO1_TIPO_VIEJO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO1_TIPO_VIEJO IS NOT NULL
    union
    SELECT NEUMATICO2_TIPO_VIEJO FROM gd_esquema.Maestra m WHERE NEU-
MATICO2_TIPO_VIEJO IS NOT NULL
    union
    SELECT NEUMATICO3_TIPO_VIEJO FROM gd_esquema.Maestra m WHERE NEU-
MATICO3_TIPO_VIEJO IS NOT NULL
    union
    SELECT NEUMATICO4_TIPO_VIEJO FROM gd_esquema.Maestra m WHERE NEUMA-
TICO4_TIPO_VIEJO IS NOT NULL
    -- Como hay neumáticos sin tipo, agregamos un valor para ellos
    INSERT INTO LOCRO.TIPO_NEUMATICO(TIPO_NEUMATICO_DESCRIPCION)
    VALUES ('SIN TIPO')
END
GO
```

Neumatico:

Nos interesa guardar el tipo de neumático y su número de serie.

Columnas:

- NEUMATICO_ID (autoincremental)
- NEUMATICO_NRIO_SERIE
- NEUMATICO_TIPO

Primero haremos la migración de todos los neumáticos que pasaron por una parada box, obteniendo así los neumáticos que cuentan con un tipo específico registrado en la tabla maestra. Luego debemos migrar los neumáticos restantes que son los que no pasaron nunca por una parada box.

Para la primera migración, realizamos la misma cantidad de UNIONS que para la tabla de tipos, con la diferencia de que ahora seleccionaremos el número de serie y el ID del tipo, esto tras hacer un JOIN a la tabla de tipos.

Para la segunda migración, realizaremos 3 UNIONS, uniendo todos los neumáticos de las telemetrías. El tipo será fijo y será el que tenga como descripción “SIN TIPO” que fue el agregado por migración para este tipo de casos, y para que no se repitan neumáticos (porque algunos ya fueron migrados en

la primera migración) usaremos una clausula WHERE que nos verifique que el neumático todavía no se haya migrado.

```
CREATE PROCEDURE migrar_neumatico
AS
BEGIN
    INSERT INTO LOCRO.NEUMATICO (NEUMATICO_NRO_SERIE, NEUMATICO_TIPO)
    SELECT DISTINCT NEUMATICO1_NRO_SERIE_NUEVO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO1_TIPO_NUEVO
    WHERE NEUMATICO1_NRO_SERIE_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO2_NRO_SERIE_NUEVO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO2_TIPO_NUEVO
    WHERE NEUMATICO2_NRO_SERIE_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO3_NRO_SERIE_NUEVO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO3_TIPO_NUEVO
    WHERE NEUMATICO2_NRO_SERIE_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO4_NRO_SERIE_NUEVO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO4_TIPO_NUEVO
    WHERE NEUMATICO2_NRO_SERIE_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO1_NRO_SERIE_VIEJO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO1_TIPO_VIEJO
    WHERE NEUMATICO1_NRO_SERIE_VIEJO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO2_NRO_SERIE_VIEJO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO2_TIPO_VIEJO
    WHERE NEUMATICO2_NRO_SERIE_VIEJO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO3_NRO_SERIE_VIEJO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO3_TIPO_VIEJO
    WHERE NEUMATICO3_NRO_SERIE_VIEJO IS NOT NULL
    UNION
    SELECT DISTINCT NEUMATICO4_NRO_SERIE_VIEJO, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
    JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = NEUMA-
TICO4_TIPO_VIEJO
    WHERE NEUMATICO4_NRO_SERIE_VIEJO IS NOT NULL
```

```

INSERT INTO LOCRO.NEUMATICO (NEUMATICO_NRO_SERIE, NEUMATICO_TIPO)
SELECT DISTINCT TELE_NEUMATICO01_NRO_SERIE, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = 'SIN
TIPO'
WHERE TELE_NEUMATICO01_NRO_SERIE IS NOT NULL
AND (SELECT COUNT(*) FROM LOCRO.NEUMATICO n WHERE n.NEUMATICO_NRO_SERIE
= TELE_NEUMATICO01_NRO_SERIE) = 0
UNION
SELECT DISTINCT TELE_NEUMATICO02_NRO_SERIE, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = 'SIN
TIPO'
WHERE TELE_NEUMATICO02_NRO_SERIE IS NOT NULL
AND (SELECT COUNT(*) FROM LOCRO.NEUMATICO n WHERE n.NEUMATICO_NRO_SERIE
= TELE_NEUMATICO02_NRO_SERIE) = 0
UNION
SELECT DISTINCT TELE_NEUMATICO03_NRO_SERIE, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = 'SIN
TIPO'
WHERE TELE_NEUMATICO03_NRO_SERIE IS NOT NULL
AND (SELECT COUNT(*) FROM LOCRO.NEUMATICO n WHERE n.NEUMATICO_NRO_SERIE
= TELE_NEUMATICO03_NRO_SERIE) = 0
UNION
SELECT DISTINCT TELE_NEUMATICO04_NRO_SERIE, tn.TIPO_NEUMATICO_ID FROM
gd_esquema.Maestra m
JOIN LOCRO.TIPO_NEUMATICO tn ON tn.TIPO_NEUMATICO_DESCRIPCION = 'SIN
TIPO'
WHERE TELE_NEUMATICO04_NRO_SERIE IS NOT NULL
AND (SELECT COUNT(*) FROM LOCRO.NEUMATICO n WHERE n.NEUMATICO_NRO_SERIE
= TELE_NEUMATICO04_NRO_SERIE) = 0
END
GO

```

Tele_Neumatico:

En esta tabla nos interesa guardar los datos relacionados a la medición. Los datos intrínsecos del neumático estarán guardados en la tabla de neumáticos (tipo, modelo, nro de serie, etc) y los vincularemos a cada telemetría con una foreign key para no duplicar datos.

Columnas:

- TELE_NEUMATICO_NUMERO (autoincremental)
- TELE_NEUMATICO_NEUMATICO
- TELE_NEUMATICO_PRESION
- TELE_NEUMATICO_POSICION
- TELE_NEUMATICO_PROFUNDIDAD
- TELE_NEUMATICO_TEMPERATURA
- TELE_NEUMATICO_TELE_AUTO

Para realizar la migración, debemos realizar un JOIN a la tabla de neumáticos y seleccionar su id junto con todo el resto de los datos de la telemetría. Esto lo deberemos repetir 4 veces, una vez para cada neumático de los 4 que trae el auto.

```

CREATE PROCEDURE migrar_tele_neumatico
AS
BEGIN
    INSERT INTO LOCRO.TELE_NEUMATICO(TELE_NEUMATICO_NEUMATICO, TELE_NEUMATICO_PRESION,
    TELE_NEUMATICO_POSICION, TELE_NEUMATICO_PROFUNDIDAD, TELE_NEUMATICO_TEMPERATURA,
    TELE_NEUMATICO_TELE_AUTO)
    SELECT DISTINCT n.NEUMATICO_ID, TELE_NEUMATICO1_PRESION, TELE_NEUMATICO1_POSICION,
    TELE_NEUMATICO1_PROFUNDIDAD, TELE_NEUMATICO1_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.NEUMATICO n ON n.NEUMATICO_NRO_SERIE =
    m.TELE_NEUMATICO1_NRO_SERIE
    WHERE TELE_NEUMATICO1_NRO_SERIE IS NOT NULL
        AND TELE_NEUMATICO1_POSICION IS NOT NULL
        AND TELE_NEUMATICO1_PRESION IS NOT NULL
        AND TELE_NEUMATICO1_PROFUNDIDAD IS NOT NULL
        AND TELE_NEUMATICO1_TEMPERATURA IS NOT NULL

    UNION

    SELECT DISTINCT n.NEUMATICO_ID, TELE_NEUMATICO2_PRESION, TELE_NEUMATICO2_POSICION,
    TELE_NEUMATICO2_PROFUNDIDAD, TELE_NEUMATICO2_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.NEUMATICO n ON n.NEUMATICO_NRO_SERIE =
    m.TELE_NEUMATICO2_NRO_SERIE
    WHERE TELE_NEUMATICO2_NRO_SERIE IS NOT NULL
        AND TELE_NEUMATICO2_POSICION IS NOT NULL
        AND TELE_NEUMATICO2_PRESION IS NOT NULL
        AND TELE_NEUMATICO2_PROFUNDIDAD IS NOT NULL
        AND TELE_NEUMATICO2_TEMPERATURA IS NOT NULL

    UNION

    SELECT DISTINCT n.NEUMATICO_ID, TELE_NEUMATICO3_PRESION, TELE_NEUMATICO3_POSICION,
    TELE_NEUMATICO3_PROFUNDIDAD, TELE_NEUMATICO3_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.NEUMATICO n ON n.NEUMATICO_NRO_SERIE =
    m.TELE_NEUMATICO3_NRO_SERIE
    WHERE TELE_NEUMATICO3_NRO_SERIE IS NOT NULL
        AND TELE_NEUMATICO3_POSICION IS NOT NULL
        AND TELE_NEUMATICO3_PRESION IS NOT NULL
        AND TELE_NEUMATICO3_PROFUNDIDAD IS NOT NULL
        AND TELE_NEUMATICO3_TEMPERATURA IS NOT NULL

    UNION

    SELECT DISTINCT n.NEUMATICO_ID, TELE_NEUMATICO4_PRESION, TELE_NEUMATICO4_POSICION,
    TELE_NEUMATICO4_PROFUNDIDAD, TELE_NEUMATICO4_TEMPERATURA, TELE_AUTO_CODIGO
    FROM gd_esquema.Maestra m JOIN LOCRO.NEUMATICO n ON n.NEUMATICO_NRO_SERIE =
    m.TELE_NEUMATICO4_NRO_SERIE
    WHERE TELE_NEUMATICO4_NRO_SERIE IS NOT NULL
        AND TELE_NEUMATICO4_POSICION IS NOT NULL
        AND TELE_NEUMATICO4_PRESION IS NOT NULL
        AND TELE_NEUMATICO4_PROFUNDIDAD IS NOT NULL
        AND TELE_NEUMATICO4_TEMPERATURA IS NOT NULL

END
GO

```

Incidente_Auto:

En esta columna queremos vincular a los autos con los incidentes en los que se vieron involucrados. Además, tendremos que guardar el número de vuelta correspondiente, ya que, en un incidente, dos autos involucrados pueden estar en vueltas distintas.

Columnas:

- AUTO_ID
- INCIDENTE_ID

- NRO_DE_VUELTA

Migramos esto realizando 3 JOINS, uno al auto por modelo y número, otro al incidente por tiempo y sector y otro al tipo de incidente por el tipo, filtrando los que tengan el tiempo de incidente, el número de vuelta o el número de auto en NULL

```
CREATE PROCEDURE migrar_incidente_auto
AS
BEGIN
    INSERT INTO LOCRO.INCIDENTE_AUTO(AUTO_ID, INCIDENTE_ID, NRO_DE_VUELTA)
    SELECT DISTINCT a.AUTO_ID, i.INCIDENTE_ID, INCIDENTE_NUMERO_VUELTA
    FROM gd_esquema.Maestra m
    JOIN LOCRO.AUTO a
        ON a.AUTO_MODELO = m.AUTO_MODELO AND a.AUTO_NUMERO = m.AUTO_NUMERO
    JOIN LOCRO.INCIDENTE i
        ON i.INCIDENTE_TIEMPO = m.INCIDENTE_TIEMPO AND i.INCIDENTE_SECTOR =
m.CODIGO_SECTOR
    JOIN LOCRO.TIPO_INCIDENTE
        ON i.INCIDENTE_TIPO = TIPO_INCIDENTE_ID AND m.INCIDENTE_TIPO = TIPO_IN-
CIDENTE_DESCRIPCION
    WHERE M.INCIDENTE_TIEMPO IS NOT NULL AND M.INCIDENTE_NUMERO_VUELTA IS NOT
NULL AND m.AUTO_NUMERO IS NOT NULL
END
GO
```

Auto_Carrera:

Una tabla intermedia en la que vincularemos a los autos con las carreras en las que participaron.

Columnas:

- AUTO_ID
- CARRERA_CODIGO

Simplemente seleccionamos el código de la carrera y hacemos un JOIN al auto para realizar la vinculación, y para identificar correctamente al auto haremos otros JOINS más a la escudería del auto y al piloto del auto.

```
CREATE PROCEDURE migrar_auto_x_carrera
AS
BEGIN
    INSERT INTO LOCRO.AUTO_CARRERA(AUTO_ID, CARRERA_CODIGO)
    SELECT DISTINCT a.AUTO_ID, CODIGO_CARRERA
    FROM gd_esquema.Maestra m
    JOIN LOCRO.AUTO a ON a.AUTO_MODELO = m.AUTO_MODELO
    JOIN LOCRO.ESCUERIA e ON e.ESCUERIA_NOMBRE = m.ESCUERIA_NOMBRE
    JOIN LOCRO.PILOTO p ON p.PILOTO_NOMBRE+p.PILOTO_APELLIDO =
m.PILOTO_NOMBRE+m.PILOTO_APELLIDO
    WHERE m.AUTO_NUMERO IS NOT NULL AND CODIGO_CARRERA IS NOT NULL
END
GO
```

Parada_box:

Esta tabla tendrá la información de una parada de un auto en boxes. No guardaremos los cambios de neumático acá, pues podría ser que algún neumático no se cambie (o quizás ninguno).

Columnas:

- PARADA_ID (autoincremental)
- PARADA_AUTO
- PARADA_CARRERA
- PARADA_VUELTA
- PARADA_TIEMPO

Al migrar los datos a la tabla, haremos un JOIN a la tabla de autos por número y modelo, ya que esa es su PK en la tabla maestra.

```
CREATE PROCEDURE migrar_parada_box
AS
BEGIN
    INSERT INTO LOCRO.PARADA_BOX (PARADA_AUTO, PARADA_CARRERA, PARADA_VUELTA, PARADA_TIEMPO)
    SELECT DISTINCT a.AUTO_ID, CODIGO_CARRERA, PARADA_BOX_VUELTA, PARADA_BOX_TIEMPO
    FROM gd_esquema.Maestra m
    JOIN LOCRO.AUTO a ON a.AUTO_MODELO = m.AUTO_MODELO AND a.AUTO_NUMERO = m.AUTO_NUMERO
    WHERE m.AUTO_NUMERO IS NOT NULL AND PARADA_BOX_VUELTA IS NOT NULL
END
GO
```

Cambio_Neumatico:

Por último, en esta tabla guardaremos los cambios de neumático, lo que implica guardar el neumático viejo, el nuevo, la posición del neumático y el ID de la parada en la que se produjo. De esta manera, no condicionamos a siempre cargar 4 neumáticos, otorgando más flexibilidad.

Columnas:

- CAMBIO_NUEVO
- CAMBIO_VIEJO
- CAMBIO_POSICION
- CAMBIO_PARADA

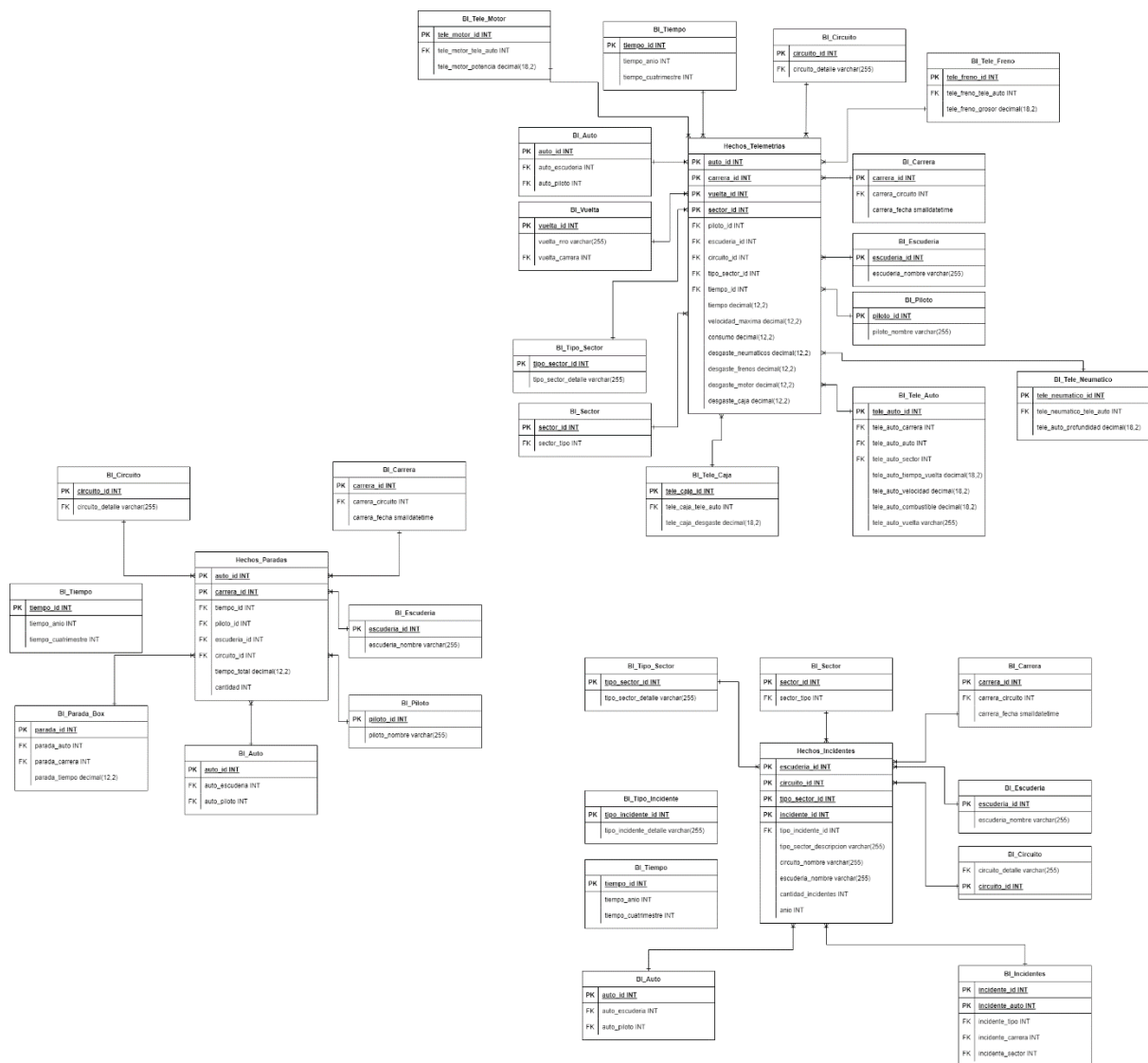
Para esta migración unimos los 4 neumáticos viejos y nuevos presentes en la tabla maestra. Hacemos un JOIN al neumático viejo y al nuevo, para obtener los ID de los neumáticos. Luego hacemos otro JOIN a la parada para obtener el ID de la parada y por último, al auto para evitar la duplicación de paradas.

```

CREATE PROCEDURE migrar_cambio_neumatico
AS
BEGIN
    INSERT INTO LOCRO.CAMBIO_NEUMATICO (CAMBIO_NUEVO, CAMBIO_VIEJO, CAMBIO_POSICION,
    CAMBIO_PARADA)
    SELECT DISTINCT nn.NEUMATICO_ID, nv.NEUMATICO_ID, NEUMATICO1_POSICION_NUEVO,
    p.PARADA_ID FROM gd_esquema.Maestra m
    JOIN LOCRO.NEUMATICO nn ON nn.NEUMATICO_NRO_SERIE = m.NEUMATICO1_NRO_SERIE_NUEVO
    JOIN LOCRO.NEUMATICO nv ON nv.NEUMATICO_NRO_SERIE = m.NEUMATICO1_NRO_SERIE_VIEJO
    JOIN LOCRO.PARADA_BOX p ON p.PARADA_VUELTA + p.PARADA_TIEMPO = m.PARADA_BOX_VUELTA
    + m.PARADA_BOX_TIEMPO
    JOIN LOCRO.AUTO a ON a.AUTO_ID = p.PARADA_AUTO AND a.AUTO_MODELO = m.AUTO_MODELO
    AND a.AUTO_NUMERO = m.AUTO_NUMERO
    WHERE m.AUTO_NUMERO IS NOT NULL AND PARADA_BOX_VUELTA IS NOT NULL AND
    NEUMATICO1_POSICION_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT nn.NEUMATICO_ID, nv.NEUMATICO_ID, NEUMATICO2_POSICION_NUEVO,
    p.PARADA_ID FROM gd_esquema.Maestra m
    JOIN LOCRO.NEUMATICO nn ON nn.NEUMATICO_NRO_SERIE = m.NEUMATICO2_NRO_SERIE_NUEVO
    JOIN LOCRO.NEUMATICO nv ON nv.NEUMATICO_NRO_SERIE = m.NEUMATICO2_NRO_SERIE_VIEJO
    JOIN LOCRO.PARADA_BOX p ON p.PARADA_VUELTA + p.PARADA_TIEMPO = m.PARADA_BOX_VUELTA
    + m.PARADA_BOX_TIEMPO
    JOIN LOCRO.AUTO a ON a.AUTO_ID = p.PARADA_AUTO AND a.AUTO_MODELO = m.AUTO_MODELO
    AND a.AUTO_NUMERO = m.AUTO_NUMERO
    WHERE m.AUTO_NUMERO IS NOT NULL AND PARADA_BOX_VUELTA IS NOT NULL AND
    NEUMATICO2_POSICION_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT nn.NEUMATICO_ID, nv.NEUMATICO_ID, NEUMATICO3_POSICION_NUEVO,
    p.PARADA_ID FROM gd_esquema.Maestra m
    JOIN LOCRO.NEUMATICO nn ON nn.NEUMATICO_NRO_SERIE = m.NEUMATICO3_NRO_SERIE_NUEVO
    JOIN LOCRO.NEUMATICO nv ON nv.NEUMATICO_NRO_SERIE = m.NEUMATICO3_NRO_SERIE_VIEJO
    JOIN LOCRO.PARADA_BOX p ON p.PARADA_VUELTA + p.PARADA_TIEMPO = m.PARADA_BOX_VUELTA
    + m.PARADA_BOX_TIEMPO
    JOIN LOCRO.AUTO a ON a.AUTO_ID = p.PARADA_AUTO AND a.AUTO_MODELO = m.AUTO_MODELO
    AND a.AUTO_NUMERO = m.AUTO_NUMERO
    WHERE m.AUTO_NUMERO IS NOT NULL AND PARADA_BOX_VUELTA IS NOT NULL AND
    NEUMATICO3_POSICION_NUEVO IS NOT NULL
    UNION
    SELECT DISTINCT nn.NEUMATICO_ID, nv.NEUMATICO_ID, NEUMATICO4_POSICION_NUEVO,
    p.PARADA_ID FROM gd_esquema.Maestra m
    JOIN LOCRO.NEUMATICO nn ON nn.NEUMATICO_NRO_SERIE = m.NEUMATICO4_NRO_SERIE_NUEVO
    JOIN LOCRO.NEUMATICO nv ON nv.NEUMATICO_NRO_SERIE = m.NEUMATICO4_NRO_SERIE_VIEJO
    JOIN LOCRO.PARADA_BOX p ON p.PARADA_VUELTA + p.PARADA_TIEMPO = m.PARADA_BOX_VUELTA
    + m.PARADA_BOX_TIEMPO
    JOIN LOCRO.AUTO a ON a.AUTO_ID = p.PARADA_AUTO AND a.AUTO_MODELO = m.AUTO_MODELO
    AND a.AUTO_NUMERO = m.AUTO_NUMERO
    WHERE m.AUTO_NUMERO IS NOT NULL AND PARADA_BOX_VUELTA IS NOT NULL AND
    NEUMATICO4_POSICION_NUEVO IS NOT NULL
END
GO

```

DER:



Para el modelo de negocios creamos 3 tablas de hechos:

- Una para todo lo relacionado con las paradas, llamada *BI_HECHOS_PARADAS*. Esta tabla está agrupada por auto y por carrera, y tiene la cantidad de veces que un auto paró en un puesto de boxes y el tiempo total que tardó en paradas en esa carrera. También guardamos datos extra como la escudería y el piloto del auto, el circuito y el tiempo en el que se jugó la carrera.
- Otra para los incidentes, llamada *BI_HECHOS_INCIDENTES*, en la que calcularemos la cantidad de incidentes totales de cada escudería en cada tipo de sector de cada circuito.
- Por último, una para almacenar todo lo relacionado con las telemetrías, llamada *BI_HECHOS_TELEMETRIAS*. Esta agrupada por auto, carrera, vuelta y sector, y contiene los valores del desgaste que sufrieron cada tipo de componente del auto en ese sector para esa vuelta, así como la velocidad máxima del auto alcanzada, el consumo de combustible y el tiempo que tardó el auto en recorrer dicho sector en esa vuelta. Un par de consideraciones:
 - Decidimos agrupar por sector y no por tipo de sector ya que una vuelta podría contar con dos sectores del mismo tipo en diferentes puntos del circuito, causando

que las mediciones para obtener los valores relativos a ese tipo de sector sean inválidos. Agrupándolos por sector, podemos calcular estos valores para cada sector de la vuelta, haciendo que los cálculos sean exactos.

- Con el fin de no duplicar filas en la tabla de hechos, decidimos utilizar 4 campos diferentes para los desgastes, cada uno refiriéndose al desgaste de sus componentes de ese tipo en ese sector de esa vuelta. De esta manera, la tabla de hechos se nos reduce a menos de 11 mil vueltas, siendo mucho más fácil de manejar y mucho más rápida de interactuar con ella.

Otras consideraciones:

- La dimensión *TIEMPO* tiene de atributos el año y el cuatrimestre. Decidimos poner el cuatrimestre y no el mes ya que en ningún momento necesitamos el mes, sino el cuatrimestre.
- En todas las vistas que requieran mostrar columnas como la escudería, el circuito y el tipo de sector, decidimos mostrar también el detalle de cada una de estas columnas, haciendo que la vista nos devuelva una query más entendible para una persona que no esté familiarizada con la base de datos.
- La dimensión tipo de neumático fue migrada, pero no ha sido incluida en ninguna tabla de hechos, ya que no le encontramos ninguna conexión coherente con las planteadas para la realización de las vistas.
 - La tabla de hechos de paradas agrupa por auto y por carrera, haciendo imposible poder vincular de alguna forma el tipo de neumático con las filas incluídas.
 - La tabla de hechos de telemetrías tiene una única columna para el desgaste de los neumáticos, haciendo imposible también vincular el tipo de neumático con dichas filas.