|  |
| --- |
| INTELIGENCIA DE NEGOCIOS |
| Tutorial: INTERCORDOBA S.A. |
| Tercer Carpeta |
|  |
| **Grupo Nro. 5**  **Boretto, José Leg. 65558**  **Diaz, Efrain Leg. 65414**  **Neil, Gabriel Leg. 62731** |
|  |

|  |
| --- |
| **AÑO 2016** |

**Índice**

[Introducción](#_gjdgxs)

Diseño y construcción de la Base de Datos Decisional

a. Análisis y Definición de Dimensiones y Métricas.

b. Diseño de el/los modelo/s de datos.

c. Definición de volumen de información.

d. Análisis de Sumarizaciones y Particionamientos.

e. Determinación de Índices necesarios.

f. Proyección del Crecimiento de la Base de Datos.

g. Presentación del Script de creación de la base de datos.

h. Miniespecificaciones de los programas de regeneración de Tablas de

Dimensión. (Tablas de dimensión).

i. Miniespecificaciones de los programas extracción de Datos. (Tablas de

hechos).

j. Miniespecificaciones de los Programas de Sumarización.

k. Miniespecificaciones de implementación de los procesos ETL (Scheduler,

dependencias, etc.)

# **Introducción**

En la siguiente presentación se realizará un análisis encarado al diseño e implementación de la solución que tratará a los requerimientos provistos por la compañía en estudio, a través de la base de datos decisional.

Se presentarán las decisiones técnicas y estrategias necesarias teniendo en cuenta las características de la organización y su flujo de trabajo.

# **Diseño y construcción de la Base de Datos Decisional**

## **a.Análisis y definición de dimensiones y métricas.**

Las dimensiones que tendremos con las cuales filtraremos a los hechos “Cantidad de Novedades” y también a la “Duracion en Dias” (dias de demora para terminar de reparar una novedad) serán:

-Tiempo (básica, para ver un seguimiento,un historial, por dia, mes o año. El id Fecha es un int porque es más eficiente hacer un JOIN para un int que para un date)

-Conductor (saber el nombre de quien conduce cada interno)

-Interno (tener el número de interno con el cual reconocerlo, y, entre otras cosas, conocer de qué año es su motor, su chasis, su marca y modelo, etc.)

-Ruta (para ver el camino que se recorre, pues no es lo mismo una ruta donde el coche va derecho todo el tiempo y a las calles de tierra dentro de un barrio o ciudad)

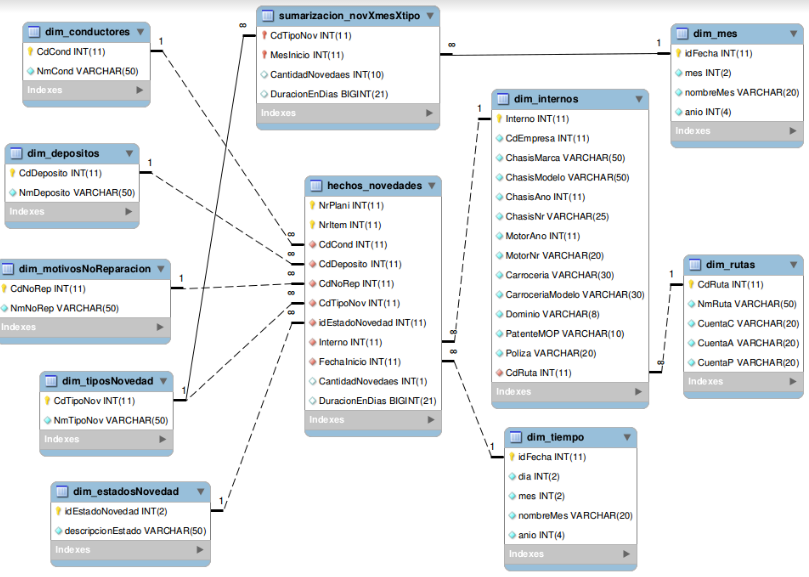
-Tipo de Novedades (El tipo es Mecánica, Gomería, Lavadero, Tapiceria, Electricidad, Carrocería, Chapa y Pintura, Aire Acondicionado, Trabajo de Terceros y Elástico)

-Depósito (se refiere a si la novedad se encuentra en la Central de Unquillo, Río Tercero o Córdoba)

-Motivos de No reparación (porque no fue posible que no se cerró una novedad)

-Estado de cada novedad (si se encuentra terminada, cerrada, pendiente, anulada)

## **b.Diseño de el/los modelo/s de datos.**



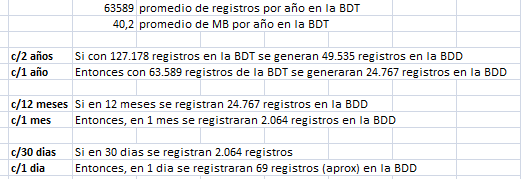
## **c.Definición de volúmen de información.**

Se manejan datos histórico de 2 años.

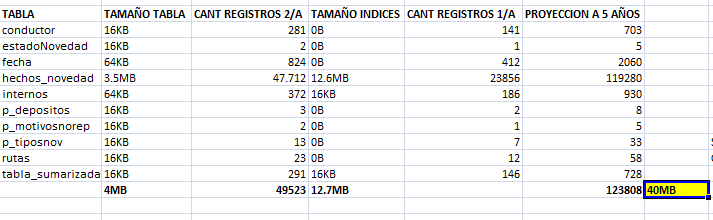
En el transacional tenemos 80.4 MB en la base de datos de 127.178 registros en la tabla principal, que es aproximadamente el 90,95% del tamaño total.

El decisional, con ese histórico del transaccional, pesa actualmente 16.4 MB en total.

Entonces, en promedio, por año en la BDT habrá cerca de 63.500 registros, que serán unos 40MB.

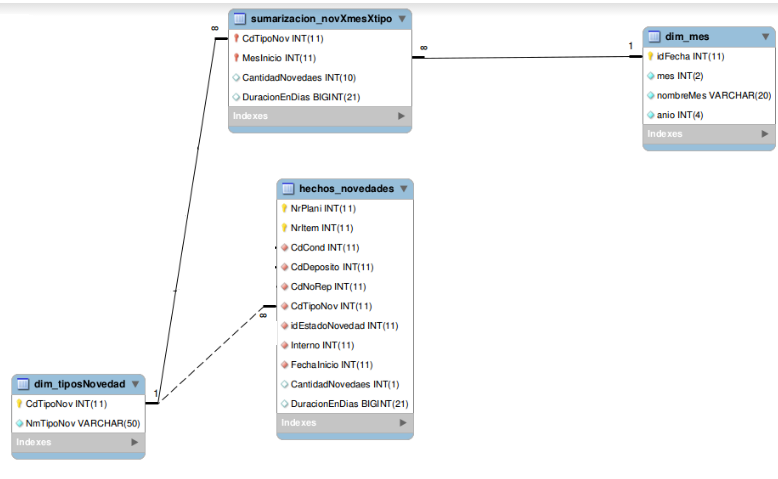


Se calcula para una proyección lineal de acá a 5 años





## **d.Análisis de particiones y sumarizaciones.**



Como el hecho más utilizado es la Cantidad de Novedades por Tipo de Novedad, por Mes, sumarizamos esa consulta, obteniendo mayor velocidad.

 Velocidad en la consulta Sumarizada

Velocidad en la consulta para la Fact Table

## **e.Determinación de índices necesarios.**

Analizando las consultas decidimos crear los siguiente índices para aquellas columnas que no son pk o fk pero si son requeridas para los reporte.

Vale aclarar que en los ETL ordenamos los resultados para que le motor de base de datos pueda recorrerlos rápidamente.  
  
En el DDL de la tabla fecha tenemos  
INDEX `inidiceAnio` (`anio`)

## **f.Proyección del crecimiento de la Base de Datos.**

Todos los días se cargan novedades en la BDT, por lo que la BDD también crecerá, con una estimación de aproximadamente 2000 novedades (registros) nuevas cada mes. En el punto c) estimamos que si la BDD crece de manera lineal, de aca a 5 años tendremos un peso de 40MB aproximadamente. Aunque cabe aclarar que en esta primera iteración no se contemplan todos los requerimientos (se aclara esto en la carpeta 2), e incluso se tiene planeado seguir expandiendo el sistema para otras áreas, no solo la del pañol.

## **g.Presentación del script de creación de la Base de Datos.**

CREATE DATABASE IF NOT EXISTS dm\_novedades;

USE dm\_novedades;

CREATE TABLE IF NOT EXISTS `dim\_tiempo` (

idFecha INT(11) PRIMARY KEY,

dia INT(2) NOT NULL,

mes INT(2) NOT NULL,

nombreMes VARCHAR(20) NOT NULL,

anio INT(4) NOT NULL,

INDEX `inidiceAnio` (`anio`)

);

CREATE TABLE IF NOT EXISTS `dim\_conductores` (

CdCond INT(11) PRIMARY KEY,

NmCond VARCHAR(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_estadosNovedad` (

idEstadoNovedad INT(2) PRIMARY KEY,

descripcionEstado VARCHAR(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_motivosNoReparacion` (

CdNoRep INT(11) PRIMARY KEY,

NmNoRep VARCHAR(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_tiposNovedad` (

CdTipoNov INT(11) PRIMARY KEY,

NmTipoNov VARCHAR(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_depositos` (

CdDeposito INT(11) PRIMARY KEY,

NmDeposito VARCHAR(50) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_rutas` (

`CdRuta` INT(11) PRIMARY KEY,

`NmRuta` VARCHAR(50) NOT NULL,

`CuentaC` VARCHAR(20) NOT NULL,

`CuentaA` VARCHAR(20) NOT NULL,

`CuentaP` VARCHAR(20) NOT NULL

);

CREATE TABLE IF NOT EXISTS `dim\_internos` (

`Interno` INT(11) PRIMARY KEY,

`CdEmpresa` INT(11) NOT NULL,

`ChasisMarca` VARCHAR(50) NOT NULL,

`ChasisModelo` VARCHAR(50) NOT NULL,

`ChasisAno` INT(11) NOT NULL,

`ChasisNr` VARCHAR(25) NOT NULL,

`MotorAno` INT(11) NOT NULL,

`MotorNr` VARCHAR(20) NOT NULL,

`Carroceria` VARCHAR(30) NOT NULL,

`CarroceriaModelo` VARCHAR(30) NOT NULL,

`Dominio` VARCHAR(8) NOT NULL,

`PatenteMOP` VARCHAR(10) NOT NULL,

`Poliza` VARCHAR(20) NOT NULL,

`CdRuta` INT(11) NOT NULL,

CONSTRAINT `fkRiuta` FOREIGN KEY (`CdRuta`)

REFERENCES `dim\_rutas` (`cdRuta`)

ON DELETE RESTRICT ON UPDATE RESTRICT

) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS `hechos\_novedades` (

`NrPlani` INT(11) NOT NULL,

`NrItem` INT(11) NOT NULL,

`CdCond` INT(11) NOT NULL,

`CdDeposito` INT(11) NOT NULL,

`CdNoRep` INT(11) NOT NULL,

`CdTipoNov` INT(11) NOT NULL,

`idEstadoNovedad` INT(11) NOT NULL,

`Interno` INT(11) NOT NULL,

`FechaInicio` INT NOT NULL,

`CantidadNovedaes` INT(1) DEFAULT 1,

`DuracionEnDias` BIGINT(21),

CONSTRAINT `pk\_todos` PRIMARY KEY (`NrPlani` , `NrItem`),

CONSTRAINT `fk\_Conductor` FOREIGN KEY (`CdCond`)

REFERENCES `dim\_conductores` (`CdCond`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Estado` FOREIGN KEY (`idEstadoNovedad`)

REFERENCES `dim\_estadosNovedad` (`idEstadoNovedad`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_tipos\_novedad` FOREIGN KEY (`CdTipoNov`)

REFERENCES `dim\_tiposNovedad` (`CdTipoNov`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_FechaInicio` FOREIGN KEY (`FechaInicio`)

REFERENCES `dim\_tiempo` (`idFecha`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Internos` FOREIGN KEY (`Interno`)

REFERENCES `dim\_internos` (`Interno`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_NoReparacion` FOREIGN KEY (`CdNoRep`)

REFERENCES `dim\_motivosNoReparacion` (`CdNoRep`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Deposito` FOREIGN KEY (`CdDeposito`)

REFERENCES `dim\_depositos` (`CdDeposito`)

ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=INNODB;

## **h.Miniespecificaciones de los programas de regeneración de Tablas de Dimensión.**

USE dm\_novedades;

-- DIMENSION TIEMPO---------------------------

INSERT INTO dm\_novedades.dim\_tiempo

SELECT distinct

UNIX\_TIMESTAMP(p\_partenov.Fecha),day(p\_partenov.Fecha),month(p\_partenov.Fecha),DATE\_FORMAT(p\_partenov.Fecha,'%M'),year(p\_partenov.Fecha)

FROM

INTER.p\_partenov

where p\_partenov.Fecha is not null and

UNIX\_TIMESTAMP(p\_partenov.Fecha) NOT IN (SELECT idFecha

FROM dm\_novedades.dim\_tiempo)

order by year(p\_partenov.Fecha), month(p\_partenov.Fecha), day(p\_partenov.Fecha);

-- CONDUCTORES---------------------------

INSERT INTO dm\_novedades.dim\_conductores

SELECT

p\_partenov.CdCond1, p\_partenov.NmCond1

FROM

INTER.p\_partenov

where

p\_partenov.CdCond1 NOT IN (SELECT CdCond

FROM dm\_novedades.dim\_conductores)

GROUP BY p\_partenov.CdCond1

ORDER BY p\_partenov.NmCond1 ASC;

-- RUTAS---------------------------

INSERT INTO dm\_novedades.dim\_rutas

SELECT

CdRuta, NmRuta, CuentaC, CuentaA, CuentaP

FROM

INTER.rutas

where

CdRuta NOT IN (SELECT CdRuta

FROM dm\_novedades.dim\_rutas);

-- TIPOS NOVEDAD---------------------------

INSERT INTO dm\_novedades.dim\_tiposNovedad

SELECT

CdTipoNov,NmTipoNov

FROM

INTER.p\_tiposnov

where

CdTipoNov NOT IN (SELECT CdTipoNov

FROM dm\_novedades.dim\_tiposNovedad);

-- MOTIVOS NO REPARACION---------------------------

INSERT INTO dm\_novedades.dim\_motivosNoReparacion (CdNoRep,NmNoRep)

SELECT

CdNoRep,NmNoRep

FROM

INTER.p\_motivosnorep

where

CdNoRep NOT IN (SELECT CdNoRep

FROM dm\_novedades.dim\_motivosNoReparacion);

-- DEPOSIOS ---------------------------

INSERT INTO dm\_novedades.dim\_depositos

SELECT

CdDeposito,NmDeposito FROM INTER.p\_depositos

where

CdDeposito NOT IN (SELECT CdDeposito

FROM dm\_novedades.dim\_depositos);

-- INTERNOS ---------------------------

INSERT INTO dm\_novedades.dim\_internos

SELECT

Interno,CdEmpresa,ChasisMarca,ChasisModelo,ChasisAno,ChasisNr,MotorAno,MotorNr,Carroceria,

CarroceriaModelo,Dominio,PatenteMOP,Poliza,CdRuta FROM INTER.internos

where CdRuta is not null and

Interno NOT IN (SELECT Interno

FROM dm\_novedades.dim\_internos);

-- ----------------------------------METADATOS----------------------------------

USE dm\_novedad;

-- METADATOS - - se ejecuta una sola vez

-- CONDUCTOR

INSERT INTO dm\_novedad.conductor

SELECT

0,'sin conductor'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.conductor

WHERE CdCond = 0);

-- RUTAS

INSERT INTO dm\_novedad.rutas

SELECT

0,'sin ruta','sin ruta','sin ruta','sin ruta'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.rutas

WHERE CdRuta = 0);

-- TIPO NOVEDAD

INSERT INTO dm\_novedad.p\_tiposnov

SELECT

0,'sin tipo novedad'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.p\_tiposnov

WHERE CdTipoNov = 0);

-- MOTIVOS NO REPARACION

INSERT INTO dm\_novedad.p\_motivosnorep

SELECT

0,'sin motivo no repacion'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.p\_motivosnorep

WHERE CdNoRep = 0);

-- ESTADO NOVEDAD pendiente

INSERT INTO dm\_novedad.estadoNovedad

SELECT

1,'Pendiente'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.estadoNovedad

WHERE idEstadoNovedad = 1);

-- ESTADO NOVEDAD Terminada

INSERT INTO dm\_novedad.estadoNovedad

SELECT

2,'Terminada'

FROM dual

WHERE NOT EXISTS (SELECT \*

FROM dm\_novedad.estadoNovedad

WHERE idEstadoNovedad = 2);

## **i.Miniespecificaciones de los programas de extracción de Datos (Tabla de Hechos).**

-- HECHOS -----------------------------

-- Antes elimino todos los que estan pendientes

DELETE FROM dm\_novedad.hechos\_novedad WHERE IdEstadoNovedad = 1;

-- el interno 106 no existe mas en la base de datos pero si tiene novedades

INSERT INTO dm\_novedad.hechos\_novedad

SELECT

p\_partenovitems.NrPlani,

NrItem,

CdCond1,

CdDeposito,

CdNoRep,

CdTipoNov,

(Terminada + 1),

Interno,

UNIX\_TIMESTAMP(p\_partenov.Fecha),

1,

TIMESTAMPDIFF(day, p\_partenov.Fecha, p\_partenovitems.Fecha) AS DiffDate

FROM

INTER.p\_partenov

JOIN

INTER.p\_partenovitems ON p\_partenov.NrPlani = p\_partenovitems.NrPlani

where NOT INTER.p\_partenov.Interno = 106 and

INTER.p\_partenov.NrPlani NOT IN (SELECT NrPlani

FROM dm\_novedad.hechos\_novedad);

## **j. Miniespecificaciones de los Programas de Sumarización.**

- - SUMARIZACION

CREATE TABLE IF NOT EXISTS `dim\_mes` (

idFecha INT(11) PRIMARY KEY,

mes INT(2) NOT NULL,

nombreMes varchar(20) not null,

anio INT(4) NOT NULL

);

CREATE TABLE IF NOT EXISTS `sumarizacion\_novXmesXtipo` (

`CdTipoNov` INT(11) NOT NULL,

`MesInicio` INT(11) NOT NULL,

`CantidadNovedaes` INT(10),

`DuracionEnDias` bigint(21) ,

CONSTRAINT `pk\_todos2` PRIMARY KEY (`CdTipoNov` , `MesInicio`),

CONSTRAINT `fk\_tipos\_novedad2` FOREIGN KEY (`CdTipoNov`)

REFERENCES `p\_tiposnov` (`CdTipoNov`)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_FechaInicio2` FOREIGN KEY (`MesInicio`)

REFERENCES `dim\_mes` (`idFecha`)

ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=INNODB;

-- SUMARIZACION

-- USAMOS OTROS FORMATO DE FECHA XQ EL UNIXTIMSTAMP TOMA TODA LA FECHA PARA CREAR EL ID, y no solo el mes y anio.

-- ETL DIM MES---------------------------

INSERT INTO dm\_novedades.dim\_mes

SELECT distinct

CONCAT(year(p\_partenov.Fecha), month(p\_partenov.Fecha)), month(p\_partenov.Fecha), DATE\_FORMAT(p\_partenov.Fecha,'%M') , year(p\_partenov.Fecha)

FROM INTER.p\_partenov

where p\_partenov.Fecha is not null and CONCAT(year(p\_partenov.Fecha), month(p\_partenov.Fecha)) NOT IN (SELECT idFecha FROM dm\_novedades.dim\_mes)

group by month(p\_partenov.Fecha), year(p\_partenov.Fecha)

order by month(p\_partenov.Fecha), year(p\_partenov.Fecha);

-- ETL hechos sumarizados

INSERT INTO dm\_novedades.sumarizacion\_novXmesXtipo

SELECT

CdTipoNov,

CONCAT(year(p\_partenov.Fecha), month(p\_partenov.Fecha)),

count(1),

SUM( TIMESTAMPDIFF(day, p\_partenov.Fecha, p\_partenovitems.Fecha))

FROM

INTER.p\_partenov

JOIN

INTER.p\_partenovitems ON p\_partenov.NrPlani = p\_partenovitems.NrPlani

where (p\_partenovitems.CdTipoNov , CONCAT(year(p\_partenov.Fecha) , month(p\_partenov.Fecha)))

NOT IN (SELECT CdTipoNov, CONCAT(year(p\_partenov.Fecha), month(p\_partenov.Fecha))

FROM dm\_novedades.sumarizacion\_novXmesXtipo )

group by CdTipoNov, month(p\_partenov.Fecha), year(p\_partenov.Fecha);

## **k.Miniespecificaciones de implementación de los procesos ETL.**

Se ejecuta todos los días a las 00hs un proceso que ejecute el Store Procedure para llenar (refrescar) las tablas de dimensiones, luego la de hechos, y finalmente la de sumarización.

Creamos los siguientes stored procedure

* DDL dimeniones
* DDL tabla hechos
* DDL tablas sumarizadas
* Insert de metadatos
* ETL todas las dimensiones
* ETL tabla echos
* ETL tablas sumarizadas
* Script que llama a los ETL

CREATE PROCEDURE `ETL\_XXX` ()

BEGIN

-- ETL

END$$

DELIMITER ;

Luego creamos un evento que llama al stored procedure principal.

CREATE EVENT IF NOT exists eventoETL\_Sumarizacion

ON SCHEDULE EVERY 1 DAY

DO

CALL dm\_novedad.ETL\_Principal();