

We build value in a world of data

Data Process I

Ejercicio Integrador

Implementación de una Solución de Datos



Índice del contenido

Objetivos 2 Premisas 2 Orígenes de datos 2 Reglas de negocio 3

Base de datos 3 Nombre BD 3 Nomenclatura por cada Entidad 4 Detalle por tabla 4

**Métricas 9 Procesos ETL's (Data Process) 9 Validaciones de datos 10
Modalidad de Entrega 10**

Anexo: 11 Stored Procedure – Genera Dim Tiempo 11

1



Objetivos

El objetivo de este ejercicio es realizar una implementación de una Solución de Datos. Desde la detección y recolección de datos, relevamiento de requerimientos y construcción del Datawarehouse, desarrollo de ETLs (Stored Procedure) y validaciones de datos.

Premisas

Partimos de la base de que las etapas de descubrimiento y relevamiento de negocio

Orígenes de datos

Las fuentes de información relevadas son las siguientes:

→ Excel que contiene la información de Ventas y Dimensiones detectadas: [QG - Ejercicio Integrador - Data Source.xls](#)

La fuente de información es un único conjunto de datos, que simula ser una base de datos con distintas tablas donde se encuentra la información de un sistema de gestión de ventas.

Contiene las siguientes tablas:

- Productos
- Categorías
- Clientes
- País
- Sucursal
- Vendedor
- Ventas



Reglas de negocio

Se identificaron las siguientes reglas de negocio:

Ventas:

- Un producto puede estar en más de una venta.
- En una venta un producto puede tener más de una cantidad vendida.
- Una Sucursal puede contener más de una venta.

- Un Producto corresponde a una Categoría.
- Un vendedor puede estar en más de una venta.
- El nivel de detalle de ventas es diario.

Deberán crear el modelo de datos que justifique estas reglas de negocio.

Base de datos

A continuación se describen los objetos de Base de Datos a desarrollar:

Nombre BD

DW_COMERCIAL

Nomenclatura para las creaciones de las tablas:

Dimensiones

Tablas Stage: STG_DIM_[Nombre_entidad]

Tablas Interfaz: INT_DIM_[Nombre_entidad]

Tablas Finales: DIM_[Nombre_entidad]

Hechos

Tablas Stage: STG_FACT_[Nombre_entidad]

Tablas Interfaz: INT_FACT_[Nombre_entidad]

Tablas Finales: FACT_[Nombre_entidad]



Nomenclatura por cada Entidad

Entidad	STG	INT	FINAL
Producto	STG_DIM_PRODUCTO	INT_DIM_PRODUCTO	DIM_PRODUCTO
Categoría	STG_DIM_CATEGORIA	INT_DIM_CATEGORIA	DIM_CATEGORIA
Cliente	STG_DIM_CLIENTE	INT_DIM_CLIENTE	DIM_CLIENTE
País	STG_DIM_PAIS	INT_DIM_PAIS	DIM_PAIS
Vendedor	STG_DIM_VENDEDOR	INT_DIM_VENDEDOR	DIM_VENDEDOR
Sucursal	STG_DIM_SUCURSAL	INT_DIM_SUCURSAL	DIM_SUCURSAL
Tiempo			DIM_TIEMPO
Ventas	STG_FACT_VENTAS	INT_FACT_VENTAS	FACT_VENTAS

Detalle por tabla

Producto

Tabla: STG_DIM_PRODUCTO:

Campo Tipo

COD_PRODUCTO Varchar(500)

DESC_PRODUCTO Varchar(500)

Tabla: INT_DIM_PRODUCTO

Campo Tipo

COD_PRODUCTO Varchar(500)

DESC_PRODUCTO Varchar(500)

Tabla: DIM_PRODUCTO

Campo Tipo

PRODUCTO_KEY Integer

COD_PRODUCTO Varchar(500)

DESC_PRODUCTO Varchar(500)

FECHA_ALTA Datetime

USUARIO_ALTA Varchar(500)

FECHA_UPDATE Datetime

USUARIO_UPDATE Varchar(500)



Categoría

Tabla: STG_DIM_CATEGORIA

Campo Tipo
COD_CATEGORIA Varchar(500)
DESC_CATEGORIA Varchar(500)

Tabla: INT_DIM_CATEGORIA
Campo Tipo
COD_CATEGORIA Varchar(500)
DESC_CATEGORIA Varchar(500)

Tabla: DIM_CATEGORIA
Campo Tipo
CATEGORIA_KEY Integer
COD_CATEGORIA Varchar(500)
DESC_CATEGORIA Varchar(500)
FECHA_ALTA Datetime
USUARIO_ALTA Varchar(500)
FECHA_UPDATE Datetime
USUARIO_UPDATE Varchar(500)

Cliente

Tabla: STG_DIM_CLIENTE
Campo Tipo
COD_CLIENTE Varchar(500)
DESC_CLIENTE Varchar(500)

Tabla: INT_DIM_CLIENTE
Campo Tipo
COD_CLIENTE Varchar(500)
NOMBRE Varchar(500)
APELLIDO Varchar(500)

Tabla: DIM_CLIENTE
Campo Tipo
CLIENTE_KEY Integer
COD_CLIENTE Varchar(500)
NOMBRE Varchar(500)
APELLIDO Varchar(500)



Varchar(500)
FECHA_UPDATE Datetime
USUARIO_UPDATE
Varchar(500)

FECHA_ALTA Datetime
USUARIO_ALTA

	Tabla: DIM_PAIS
	Campo Tipo PAIS_KEY
Tabla: STG_DIM_PAIS	Integer COD_PAIS
Campo Tipo COD_PAIS	Varchar(3) DESC_PAIS
Varchar(500) DESC_PAIS	Varchar(500) FECHA_ALTA
Varchar(500)	Datetime USUARIO_ALTA
	Varchar(500)
Tabla: INT_DIM_PAIS	FECHA_UPDATE Datetime
Campo Tipo COD_PAIS	USUARIO_UPDATE
Varchar(3) DESC_PAIS	Varchar(500)
Varchar(500)	

Vendedor

Tabla: STG_DIM_VENDEDOR
 Campo Tipo
 COD_VENDEDOR Varchar(500)
 DESC_VENDEDOR Varchar(500)

Tabla: INT_DIM_VENDEDOR
 Campo Tipo
 COD_VENDEDOR Varchar(500)
 NOMBRE Varchar(500)
 APELLIDO Varchar(500)

Tabla: DIM_VENDEDOR
 Campo Tipo
 VENDEDOR_KEY Integer



COD_VENDEDOR Integer
 NOMBRE Varchar(500) APELLIDO Varchar(500)
 FECHA_ALTA Datetime USUARIO_ALTA
 Varchar(500) FECHA_UPDATE Datetime
 USUARIO_UPDATE Varchar(500)

Sucursal

Tabla: STG_DIM_SUCURSAL
 Campo Tipo
 COD_SUCURSAL Varchar(500)
 DESC_SUCURSAL Varchar(500)

Tabla: INT_DIM_SUCURSAL

Campo Tipo

COD_SUCURSAL Varchar(500)

DESC_SUCURSAL Varchar(500)

Tabla: DIM_SUCURSAL

Campo Tipo

SUCURSAL_KEY Integer

COD_SUCURSAL Varchar(500)

DESC_SUCURSAL Varchar(500)

FECHA_ALTA Datetime

USUARIO_ALTA Varchar(500)

FECHA_UPDATE Datetime

USUARIO_UPDATE Varchar(500)

Tiempo

Tabla: DIM_TIEMPO

Campo Tipo

TIEMPO_KEY smalldatetime (hace referencia al día)

ANIO integer

MES_NRO integer

MES_NOMBRE varchar(15)

SEMESTRE integer

TRIMESTRE integer

SEMANA_ANIO integer

SEMANA_NRO_MES integer

DIA integer

DIA_NOMBRE varchar(20) DIA_SEMANA_NRO
integer

FECHA_ALTA Datetime USUARIO_ALTA

Varchar(500) FECHA_UPDATE Datetime

USUARIO_UPDATE Varchar(500)



Ventas

Tabla: STG_FACT_VENTAS

Campo Tipo

COD_PRODUCTO Varchar(500)

COD_CATEGORIA Varchar(500)

COD_CLIENTE Varchar(500)

COD_PAIS Varchar(500)
COD_VENDEDOR Varchar(500)
COD_SUCURSAL Varchar(500)
FECHA Varchar(500)
CANTIDAD_VENDIDA Varchar(500)
MONTO_VENDIDO Varchar(500)
PRECIO Varchar(500)
COMISION_COMERCIAL Varchar(500)

Tabla: INT_FACT_VENTAS

Campo Tipo

COD_PRODUCTO Varchar(100)
COD_CATEGORIA Varchar(100)
COD_CLIENTE Varchar(100)
COD_PAIS Varchar(100)
COD_VENDEDOR Varchar(100)
COD_SUCURSAL Varchar(100)
Fecha smalldatetime
CANTIDAD_VENDIDA decimal(18,2)
MONTO_VENDIDO decimal(18,2)
PRECIO decimal(18,2)
COMISION_COMERCIAL decimal(18,2)

Tabla: FACT_VENTAS

Campo Tipo

PRODUCTO_KEY Integer
CATEGORIA_KEY Integer

8



CLIENTE_KEY Integer
PAIS_KEY Integer VENDEDOR_KEY Integer
SUCURSAL_KEY Integer
TIEMPO_KEY smalldatetime
CANTIDAD_VENDIDA decimal(18,2)
MONTO_VENDIDO decimal(18,2)
PRECIO decimal(18,2) COMISION_COMERCIAL
decimal(18,2)
FECHA_ALTA Datetime USUARIO_ALTA
Varchar(500)

Métricas

Ventas:

- ◆ Monto Total de Ventas (\$)
- ◆ Cantidad vendida (#)
- ◆ Monto promedio de Ventas (\$)
- ◆ Importe Comisión Comercial (\$)
- ◆ Cantidad de Clientes (#)

Procesos ETL's (Data Process)

Deberán crear los Stored Procedures que realizarán la carga de las tablas INT (Tablas intermedias) en una primera instancia, y luego otros SPs que poblará las tablas DIM y FACT, según corresponda.

Nomenclatura de los stored procedure para la carga de Int: sp_carga_int_<nombre_dimension>

Nomenclatura de los stored procedure para la carga de dimensiones:

sp_carga_dim_<nombre_dimension>

Nomenclatura de los stored procedure para la carga de las tablas de

Hechos: Int: sp_carga_int_<nombre_fact>

Fact: sp_carga_fact_<nombre_fact>



Validaciones de datos

Para la tabla Fact validar cantidad de registros, Montos totales, Suma de cantidades aperturados por distintas dimensiones

Para las tablas de dimensiones validar integridad, cantidad de registros

Modalidad de Entrega

A continuación se describen los entregables del proyecto

Fecha de Entrega jueves 23/06

Entregables:

1. Diagrama de Modelo de Datos lógico:

- a. Imagen en PDF o PPT o Link si utilizan alguna herramienta de diagramas. El entregable deberá tener el siguiente nombre:
NOMBRE_APELLIDO_DW_COMERCIAL_MODELO

2. Script con el modelo físico de datos:

- a. Script de Creación de Tablas con nombre
NOMBRE_APELLIDO_DW_COMERCIAL_CREATE_TABLES.sql
- b. Script con el Alter de las tablas con nombre
NOMBRE_APELLIDO_DW_COMERCIAL_ALTER_TABLES.sql
- c. Script de Inserts Iniciales de las tablas de Dimensiones con el nombre
NOMBRE_APELLIDO_DW_COMERCIAL_INSERTS.sql
- d. Script con los Stored Procedures de las cargas de Tablas Intermedias y Dimensiones con el nombre NOMBRE_APELLIDO_DW_COMERCIAL_STORED_PROCEDURES.sql

3. Documento con Validaciones (Word donde se describa las validaciones que realizaron. Se puede agregar imágenes de consultas, queries, imágenes del tablero e imágenes de las validaciones sobre el excel). El word/ doc debera tener el siguiente nombre: NOMBRE_APELLIDO_DW_COMERCIAL_VALIDACIONES.docx y el script con las validaciones con el nombre: NOMBRE_APELLIDO_DW_COMERCIAL_SCRIPT_VALIDACIONES.sql

10



4. Documento con Supuestos (Word que explique los supuestos que tomaron en cuenta a la hora de cargar el DW y/o el desarrollo del tablero) en caso de que aplique. El word/ doc debera tener el siguiente nombre: NOMBRE_APELLIDO_DW_COMERCIAL_SUPUESTOS.docx

Anexo:

Stored Procedure – Genera Dim Tiempo

```
CREATE PROCEDURE [dbo].[Sp_Genera_Dim_Tiempo]
```

```
@anio Int
```

```
As
```

```
SET NOCOUNT ON
```

```
SET arithabort off
```

```
SET arithignore on
```

```
/*  
*****  
*/
```

```
/* Variables */
```

```
/*  
*****  
*/
```

```
SET DATEFIRST 1;
```

```
SET DATEFORMAT mdy
```

```
DECLARE @dia smallint
```

```
DECLARE @mes smallint
```

```
DECLARE @f_txt varchar(10)
```

```
DECLARE @fecha smalldatetime
```

```
DECLARE @key int
```



```
DECLARE @vacio smallint
```

```
DECLARE @fin smallint
```

```
DECLARE @fin_mes int
```

```
DECLARE @anioperiodicidad int
```

```
SELECT @dia = 1
```

```
SELECT @mes = 1
```

```
SELECT @f_txt = Convert(char(2), @mes) + '/' + Convert(char(2), @dia) + '/' + Convert(char(4),
```

```
@anio) SELECT @fecha = Convert(smallerdatetime, @f_txt)
```

```
select @anioperiodicidad = @anio
```

```
/*  
*****  
*/
```

```
/* Se chequea que el año a procesar */
```

```
/* no exista en la tabla TIME */
```

```
/*  
*****  
*/
```

```
IF (SELECT Count(*) FROM dim_tiempo WHERE anio = @anio) > 0
```

```
BEGIN
```

```
Print 'El año que ingreso ya existe en la tabla'
```

```
Print 'Procedimiento CANCELADO.....'
```

```
Return 0
```

```
END
```



```
/*  
*****  
*/
```

```
/* Se inserta día a día */
```

```
/* hasta terminar el año */
```

```
/* **** */
```

```
SELECT @fin = @anio + 1
```

```
WHILE (@anio < @fin)
```

```
BEGIN
```

```
--Armo la fecha
```

```
IF Len(Rtrim(Convert(Char(2),Datepart(mm, @fecha))))=1
```

```
BEGIN
```

```
IF Len(Rtrim(Convert(Char(2),Datepart(dd, @fecha))))=1
```

```
SET @f_txt = Convert(char(4),Datepart(yyyy, @fecha)) + '0' +  
Rtrim(Convert(Char(2),Datepart(mm, @fecha))) + '0' + Rtrim(Convert(Char(2),Datepart(dd, @fecha)))
```

```
ELSE
```

```
SET @f_txt = Convert(char(4),Datepart(yyyy, @fecha)) + '0' +  
Rtrim(Convert(Char(2),Datepart(mm, @fecha))) + Convert(Char(2),Datepart(dd, @fecha))
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
IF Len(Rtrim(Convert(Char(2),Datepart(dd, @fecha))))=1
```

```
SET @f_txt = Convert(char(4),Datepart(yyyy, @fecha)) +  
Convert(Char(2),Datepart(mm, @fecha)) + '0' + Rtrim(Convert(Char(2),Datepart(dd, @fecha)))
```

```
ELSE
```

```
SET @f_txt = Convert(char(4),Datepart(yyyy, @fecha)) +  
Convert(Char(2),Datepart(mm, @fecha)) + Convert(Char(2),Datepart(dd, @fecha))
```

```
END
```

```
--Calculo el último día del mes
```

```
SET @fin_mes = day(dateadd(d, -1, dateadd(m, 1, dateadd(d, - day(@fecha) + 1, @fecha))))
```

```
INSERT Dim_Tiempo (Tiempo_Key, Anio, Mes, Mes_Nombre, Semestre, Trimestre, Semana_Anio  
                  ,Semana_Nro_Mes, Dia, Dia_Nombre, Dia_Semana_Nro)
```

```
SELECT
```

```
    tiempo_key = @fecha
```

```
, anio = Datepart(yyyy, @fecha)
```

```
, mes = Datepart(mm, @fecha)
```

```
--, mes_nombre = Datename(mm, @fecha)
```

```
, mes_nombre = CASE Datename(mm, @fecha)
```

```
    when 'January' then 'Enero'
```

```
    when 'February' then 'Febrero'
```

```
    when 'March' then 'Marzo'
```

```
    when 'April' then 'Abril'
```

```
    when 'May' then 'Mayo'
```

```
    when 'June' then 'Junio'
```

```
    when 'July' then 'Julio'
```

```
    when 'August' then 'Agosto'
```

```
    when 'September' then 'Septiembre'
```

```
    when 'October' then 'Octubre'
```

```
    when 'November' then 'Noviembre'
```

```
    when 'December' then 'Diciembre'
```

```
    else Datename(mm, @fecha)
```

```

        END

, semestre = CASE Datepart(mm, @fecha)

        when (SELECT Datepart(mm, @fecha)

                WHERE Datepart(mm, @fecha) between 1 and 6) then 1

        else 2

        END

, trimestre = Datepart(qq, @fecha)

, semana_anio = Datepart(wk, @fecha)

, semana_nro_mes = Datepart(wk, @fecha) - datepart(week,
dateadd(dd,-day(@fecha)+1,@fecha)) +1

, dia = Datepart(dd, @fecha)

, dia_nombre = CASE Datename(dw, @fecha)

        when 'Monday' then 'Lunes'

        when 'Tuesday' then 'Martes'

        when 'Wednesday' then 'Miercoles'

        when 'Thursday' then 'Jueves'

        when 'Friday' then 'Viernes'

        when 'Saturday' then 'Sabado'

        when 'Sunday' then 'Domingo'

        else Datename(dw, @fecha)

        END

--, dia_nombre = Datename(dw, @fecha)

, dia_semana_nro = Datepart(dw, @fecha)

SELECT @fecha = Dateadd(dd, 1, @fecha)

```



END

```
SELECT @dia = Datepart(dd, @fecha) SELECT @mes =  
Datepart(mm, @fecha) SELECT @anio = Datepart(yy,  
@fecha) CONTINUE
```