

### API - 3

Los códigos SQL para generar tres datamarts aislados en Amazon Redshift, utilizando la capa maestra y las relaciones definidas en los Hubs, Satélites y Links previamente establecidos. Estos datamarts se centrarán en obtener información sobre la foto del stock del día anterior, los proveedores y el escaneo de productos.

--Primero creamos los usuarios:

```
CREATE LOGIN Teclab1 WITH PASSWORD = 'Teclab01';
```

```
CREATE LOGIN Teclab2 WITH PASSWORD = 'Teclab02';
```

```
USE master; --Nombre de la base de datos de SQL Server Management Studio
```

```
CREATE USER Usuario1 FOR LOGIN Teclab1;
```

```
CREATE USER Usuario2 FOR LOGIN Teclab2;
```

#### 1. Datamart de Foto de Stock del Día Anterior:

--Este SQL generará un datamart que contiene información sobre la foto del stock del día anterior utilizando la llave de negocio correspondiente.

```
CREATE SCHEMA datamart_foto_stock;
```

```
CREATE TABLE DATAMART_FOTO_STOCK.FOTO_STOCK (  
    idArticulo INT PRIMARY KEY,  
    Idnumero INT,  
    TipoArticulo VARCHAR(50),  
    DescripcionArticulo VARCHAR(100),  
    Scanning INT,  
    UxB INT,  
    StockDiaAnterior INT  
);
```

--A continuación se insertan los datos en la tabla:

```
INSERT INTO DATAMART_FOTO_STOCK.FOTO_STOCK (idArticulo, Idnumero, TipoArticulo, DescripcionArticulo, Scanning, UxB, StockDiaAnterior)
```

```
VALUES ('Tipo1', 1, 101, 'Descripción1', 100, 5, 500),
```

```
      ('Tipo2', 2, 102, 'Descripción2', 200, 6, 600),
```

```
      ('Tipo3', 3, 103, 'Descripción3', 150, 4, 450);
```

## SELECT

```
ARTICULO.IDARTICULO,  
ARTICULO.IDNUMERO,  
ARTICULO.MNA AS TIPOARTICULO,  
ARTICULO.DESCRIP AS DESCRIPCIONARTICULO,  
DETALLESTOCK.SCANNING,  
DETALLESTOCK.UXB,  
DETALLESTOCK.CANTSTOCK AS STOCKDIAANTERIO
```

## FROM

```
MAESTRA.HUB_MNADARTICULOS AS ARTICULO
```

## JOIN

```
MAESTRA.SAT_DETALLESTOCK AS DETALLESTOCK ON ARTICULO.IDARTICULO = DETALLESTOCK.IDARTICULO
```

## WHERE

```
DETALLESTOCK.Ts = (SELECT MAX(Ts) FROM MAESTRA.SAT_DETALLESTOCK WHERE Ts < CURRENT_DATE);
```

--A continuación se dan los permisos a los usuarios creados:

```
GRANT SELECT ON DATAMART_FOTO_STOCK.FOTO_STOCK TO Usuario1;
```

```
GRANT SELECT ON DATAMART_FOTO_STOCK.FOTO_STOCK TO Usuario2;
```

## 2. Datamart de Proveedores:

--Este SQL generará un datamart que contiene información sobre los proveedores.

```
CREATE SCHEMA datamart_proveedores;
```

```
CREATE TABLE DATAMART_PROVEEDORES.PROVEEDORES (
```

```
  IdProveedor INT PRIMARY KEY,  
  NombreProveedor VARCHAR(100),  
  Direccion VARCHAR(255),  
  Telefono VARCHAR(20),  
  Email VARCHAR(100)
```

```
);
```

--A continuación se insertan los datos en la tabla:

```
INSERT INTO Datamart_Proveedores.Proveedores (IdProveedor, NombreProveedor, Direccion, Telefono, Email)
```

## VALUES

```
(1, 'Proveedor1', 'Mitre 256', '011-456-7890', 'proveedor1@yahoo.com'),  
(2, 'Proveedor2', 'Estomba 1203', '011-654-3210', 'proveedor2@hotmail.com'),
```

```
(3, 'Proveedor3', 'Urquiza 128', '0291-922-4633', 'proveedor3@gmail.com');
```

#### SELECT

```
PROVEEDOR.IDPROVEEDOR,  
PROVEEDOR.NOMBRE AS NOMBREPROVEEDOR,  
PROVEEDOR.DIRECCION AS DIRECCION,  
PROVEEDOR.TELEFONO AS TELEFONO  
PROVEEDOR.EMAIL AS EMAIL
```

#### FROM

```
MAESTRA.HUB_PROVEEDOR AS PROVEEDOR;
```

--A continuación se dan los permisos a los usuarios creados:

```
GRANT SELECT ON DATAMART_PROVEEDORES.PROVEEDORES TO Usuario1;
```

```
GRANT SELECT ON DATAMART_PROVEEDORES.PROVEEDORES TO Usuario2;
```

### 3. Datamart de Escaneo de Productos:

--Este SQL generará un datamart que contiene información sobre el escaneo de productos.

```
CREATE SCHEMA datamart_escaneo_productos;
```

```
CREATE TABLE DATAMART_ESCA NEO_PRODUCTOS.ESCA NEO_PRODUCTOS (  
    IdEscaneo INT PRIMARY KEY,  
    idArticulo INT,  
    Idnumero INT,  
    TipoArticulo VARCHAR(50),  
    DescripcionArticulo VARCHAR(100),  
    Scanning INT,  
    UxB INT,  
    StockActual INT  
);
```

--A continuación se insertan los datos en la tabla:

```
INSERT INTO DATAMART_ESCA NEO_PRODUCTOS.ESCA NEO_PRODUCTOS (IdEscaneo , idArticulo, Idnumero, TipoArticulo, DescripcionArticulo, Scanning, UxB,  
StockActual)
```

#### VALUES

```
('Tipo1', 1, 101, 'Descripción1', 120, 5, 550),
```

```
('Tipo2', 2, 102, 'Descripción2', 210, 6, 590),  
( 'Tipo3', 3, 103, 'Descripción3', 155, 4, 455);
```

#### SELECT

```
ESCANE0. IDESCANE0,  
ARTICULO. IDARTICULO,  
ARTICULO. IDNUMERO,  
ARTICULO.MNA AS TIPOARTICULO,  
ARTICULO.DESCRIP AS DESCRIPCIONARTICULO,  
DETALLESTOCK.SCANING,  
DETALLESTOCK.UXB,  
DETALLESTOCK.CANTSTOCK AS STOCKACTUAL
```

#### FROM

```
MAESTRA.HUB_MNADARTICULOS AS ARTICULO
```

#### JOIN

```
MAESTRA.SAT_DETALLESTOCK AS DETALLESTOCK ON ARTICULO.IDARTICULO = DETALLESTOCK.IDARTICULO
```

#### WHERE

```
DETALLESTOCK.Ts = CURRENT_DATE;
```

----A continuación se dan los permisos a los usuarios creados:

```
GRANT SELECT ON DATAMART_ESCANE0_PRODUCTOS.ESCANE0_PRODUCTOS TO Usuario1;
```

```
GRANT SELECT ON DATAMART_ESCANE0_PRODUCTOS.ESCANE0_PRODUCTOS TO Usuario2;
```

Estos códigos SQL crean tres esquemas de datamart separados y generan tablas en cada uno de ellos para almacenar la información deseada. Los usuarios de negocio pueden acceder a estos datamarts sin necesidad de acceder a la capa maestra.

#### Diagrama de secuencia de carga:



