

Universidad Técnica Nacional

Sede: San Carlos

ISW-911 Minería de Datos, Grupo Innova S.A

Tema:

DataWarehouse, Grupo Innova S.A

Integrantes:

Luis Adrián Boza Muñoz

Esteban Enrique Salas Guzmán

Romario Reyes

Fecha de entrega: 12/03/2021

**Marco de referencia**

Al iniciar un proyecto es importante tener un marco de referencia sobre de que se va a tratar el proyecto y que se va a desarrollar en él, dado que consiste en elaborar una construcción de un DataWarehouse que permita conocer información mas detallada y modelada de un conjunto de datos relevantes para la empresa, así como ventas mensuales, rendimientos de las sucursales y empleados, márgenes de utilidad neta, etc.

Por consiguiente, aquí entra la importancia de tener claro los conceptos básicos sobre un DataWarehouse, Al profundizar un poco sobre esto surgen nombre como Barry Devlin y Paul Murphy ellos son los generan el concepto que hoy en día se conoce como DataWarehouse, sin olvidar a William H. Inmon el cual es conocido como el padre de DataWarehousing, que permitió moldear todo este sistema unificado.

¿Por lo tanto, en que consiste y a que se refiere DataWarehouse? Todos los datos que guardan en diversos sistemas lógicos o físicos de una empresa no solo sirven para respaldo de información, sino también con fines analíticos, acá es donde un data warehouse entra al papel protagónico capturando los datos de diferentes aplicaciones de procesamiento de transacciones online (OLTP). Alojándose en un servidor corporativo o cada vez mas en la nube, permite hacer consultas por usuarios como una aplicación analítica que ayuda a organizar, comprender y utilizar sus datos para beneficios de su empresa, ayudando a dar un panorama más amplio y así tomar decisiones estratégicas.

¿Además, como se conforma un DataWarehouse? Se caracteriza por una estructura básica, compuesta por subunidades de información agregada (Data Marts), estas permiten que cada una se especialice de acuerdo con un área en especifico o a un proceso; además añadiendo puntos de análisis o dimensiones, utiliza indicadores aumentando la relevancia estratégica.

* Generación de reportes
* Analítica de información
* Tableros de control
* Minería de datos

Por otra parte, Surge una tecnología que genera un gran impacto en el mercado que al contrario de un DataWarehouse que solo almacena datos que han sido modelados o estructurados; esta no hace excepción de datos al contrario los almacena, datos no estructurados, estructurados, semiestructurados. Almacenando grandes cantidades de datos en bruto que se mantiene ahí hasta que sean necesario, todo esto hace referencia a Data Lake un repositorio de almacenamiento que se diferencia de DataWarehouse que es jerárquico y almacena datos en ficheros o carpetas, por lo contrario, Data Lake utiliza una arquitectura plana para almacenar los datos.

Data Lake a cada elemento se le asigna un identificador único y se etiqueta con un conjunto de etiquetas de metadatos extendidas, que ellos optan por solo utilizar los datos necesarios; por ejemplo, Data Lake si se le presenta una cuestión de negocios que se debe resolver, tiende a solicitar los datos solo que estén relacionados con esa cuestión. Una vez obtenidos se analizan específicamente ese conjunto de datos pequeños para agilizar la respuesta.

Por consiguiente, a menudo se asocia con el almacenamiento de objetos orientado a Hadoop que se utilizan principalmente para procesar y almacenar datos no relacionales, como archivos de registro, objetos JSON, imágenes y publicaciones en redes sociales, flujo de clicks de internet. En este escenario, los datos primero se llevan a Hadoop, y se le aplica las herramientas de análisis y de minería de datos que residen en los nodos clúster de Hadoop.

El motivo por el cual revoluciono Data Lake especialmente es la centralización de fuentes de contenidos dispares ya que tiene beneficios que pueden ser combinadas y procesadas utilizando big data. Búsquedas, análisis con apropiadas medidas de seguridad que pueden ser asignadas de manera que se otorga cierta información a los usuarios que no tienen acceso a la fuente de contenido original, que de otro modo se imposibilita.

Una vez que el contenido está en el Data Lake, puede normalizarse y enriquecerse. Esto puede incluir extracción de metadatos, conversión de formatos, aumento, extracción de entidades, reticulación, agregación, desnormalización o indexación.

Es importante recalcar, que los datos al ser preparados según lo necesario reducen los costos de preparación sobre el procesamiento inicial. Al tener acceso flexible a un Data Lake esto aumenta la reutilización y ayuda la organización al recopilar mas fácilmente los datos necesarios para impulsar decisiones empresariales.

En conclusión, las diferencias claves que se logran destacar entre Data Lake y DataWarehouse son principalmente en los puntos de datos, procesamiento, almacenamiento, agilidad y seguridad.

• Datos: Un DataWarehouse sólo almacena datos que han sido modelados o estructurados, mientras que un Data Lake no hace excepción de datos. Lo almacena todo, estructurado, semiestructurado y no estructurado.

• Procesamiento: Antes de que una empresa pueda cargar datos en un data warehouse, primero debe darles forma y estructura, es decir, los datos deben ser modelados. Eso se llama schema-on-write. Con un Data Lake, sólo se cargan los datos sin procesar, tal y como están, y cuando esté listo para usar los datos, es cuando se le da forma y estructura. Eso se llama schema-on-read. Dos enfoques muy diferentes.

• Almacenamiento: Una de las principales características de las tecnologías de big data, como Hadoop, es que el coste de almacenamiento de datos es relativamente bajo en comparación con el de un DataWarehouse. Hay dos razones principales para esto una de ellas es que Hadoop es software de código abierto, por lo que la concesión de licencias y el soporte de la comunidad es gratuito. Y segundo, Hadoop está diseñado para ser instalado en hardware de bajo coste.

• Agilidad: Un almacén de datos es un repositorio altamente estructurado, por definición. No es técnicamente difícil cambiar la estructura, pero puede tomar mucho tiempo dado todos los procesos de negocio que están vinculados a ella. Un Data Lake, por otro lado, carece de la estructura de un DataWarehouse, lo que da a los desarrolladores y a los científicos de datos la capacidad de configurar y reconfigurar fácilmente y en tiempo real sus modelos, consultas y aplicaciones.

• Seguridad: La tecnología de la DataWarehouse existe desde hace décadas, mientras que la tecnología de big data (la base de un Data Lake) es relativamente nueva. Por lo tanto, la capacidad de asegurar datos en un DataWarehouse es mucho más madura que asegurar datos en un Data Lake. Cabe señalar, sin embargo, que se está realizando un importante esfuerzo en materia de seguridad en la actualidad en la industria de Big Data.

Modelos de William H. Inmon y Ralph Kimball

El DataWarehouse de Kimball está orientado a la consulta de la información, por lo que su estructura interna está especialmente diseñada para garantizar una explotación de los datos rápida y sencilla, no requiriendo usuarios especializados para ello.

El DataWarehouse de Inmon persigue la integración de todos los datos de la compañía, estando orientado hacia el almacenaje de grandes volúmenes de datos, por lo que su estructura interna normalizada se diseña para evitar la redundancia de datos, simplificar las labores de mantenimiento, etc. cuestiones que complican las consultas de la información, requiriendo que los usuarios finales estén mucho más especializados.

Así, es posible decir que el enfoque de Kimball se ajusta más a proyectos pequeños en los que se persiga un sistema fácilmente explotable y entendible por el usuario y de rápido desarrollo, siendo el modelo de Inmon más apropiado para sistemas complejos de mayor envergadura.

|  |  |  |
| --- | --- | --- |
|  | Inmon | Kimball |
| Presupuesto | Coste inicial alto | Coste inicial bajo |
| Plazos | Requiere más tiempo de desarrollo | Tiempo de desarrollo inferior |
| Expertise | Equipo con especialización alta | Equipo con especialización media |
| Alcance | Toda la compañía | Departamentos individuales |
| Mantenimiento | Fácil mantenimiento | Mantenimiento más complejo |

**Base de datos original (Oracle 11g)**

-- SUCURSALES

CREATE TABLE PROYECTO1.SUCURSALES (

    ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    PRIMARY KEY (ID)

);

CREATE SEQUENCE PROYECTO1.SUCURSALES\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.SUCURSALES\_ID

    BEFORE INSERT ON PROYECTO1.SUCURSALES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.SUCURSALES\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

-- PERSONAS

CREATE TABLE PROYECTO1.PERSONAS (

    ID INT NOT NULL,

    CEDULA INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    APELLIDOS VARCHAR(50) NOT NULL,

    CORREO VARCHAR(50) NOT NULL,

    FEC\_NACIMIENTO DATE NOT NULL,

    ESTADO\_CIVIL INT NOT NULL,

    SEXO INT NOT NULL,

    OCUPACION INT NOT NULL,

    EDUCACION INT NOT NULL,

    HIJOS INT NOT NULL,

    PRIMARY KEY (ID)

);

CREATE SEQUENCE PROYECTO1.PERSONAS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.PERSONAS\_ID

    BEFORE INSERT ON PROYECTO1.PERSONAS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.PERSONAS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

--EMPLEADOS

CREATE TABLE PROYECTO1.EMPLEADOS (

    ID INT NOT NULL,

    PERSONA\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    SALARIO INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT EMPLEADO\_PERSONA\_ID\_FK FOREIGN KEY (PERSONA\_ID)

        REFERENCES PROYECTO1.PERSONAS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.EMPLEADOS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.EMPLEADOS\_ID

    BEFORE INSERT ON PROYECTO1.EMPLEADOS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.EMPLEADOS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

CREATE SEQUENCE PROYECTO1.EMPLEADOS\_CODIGO\_SEQ;

CREATE TRIGGER PROYECTO1.EMPLEADOS\_CODIGO

    BEFORE INSERT ON PROYECTO1.EMPLEADOS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.EMPLEADOS\_CODIGO\_SEQ.nextval

    INTO :new.CODIGO

    FROM dual;

END;

-- CLIENTES

CREATE TABLE PROYECTO1.CLIENTES (

    ID INT NOT NULL,

    PERSONA\_ID NOT NULL,

    CODIGO INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT CLINTE\_PERSONA\_ID\_FK FOREIGN KEY (PERSONA\_ID)

        REFERENCES PROYECTO1.PERSONAS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.CLIENTES\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.CLIENTES\_ID

    BEFORE INSERT ON PROYECTO1.CLIENTES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.CLIENTES\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

CREATE SEQUENCE PROYECTO1.CLIENTES\_CODIGO\_SEQ;

CREATE TRIGGER PROYECTO1.CLIENTES\_CODIGO

    BEFORE INSERT ON PROYECTO1.CLIENTES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.CLIENTES\_CODIGO\_SEQ.nextval

    INTO :new.CODIGO

    FROM dual;

END;

--PROVINCIAS

CREATE TABLE PROYECTO1.PROVINCIAS (

    ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    PRIMARY KEY (ID)

);

CREATE SEQUENCE PROYECTO1.PROVINCIAS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.PROVINCIA\_ID

    BEFORE INSERT ON PROYECTO1.PROVINCIAS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.PROVINCIAS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

--CANTONES

CREATE TABLE PROYECTO1.CANTONES (

    ID INT NOT NULL,

    PROVINCIA\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT CANTON\_PROVINCIA\_ID\_FK FOREIGN KEY (PROVINCIA\_ID)

        REFERENCES PROYECTO1.PROVINCIAS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.CANTONES\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.CANTONES\_ID

    BEFORE INSERT ON PROYECTO1.CANTONES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.CANTONES\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

--DISTRITOS

CREATE TABLE PROYECTO1.DISTRITOS (

    ID INT NOT NULL,

    CANTON\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    CODIGO\_POSTAL INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT DISTRITO\_CANTON\_ID FOREIGN KEY (CANTON\_ID)

        REFERENCES PROYECTO1.CANTONES (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.DISTRITOS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.DISTRITOS\_ID

    BEFORE INSERT ON PROYECTO1.DISTRITOS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.DISTRITOS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

-- DIRECCIONES

CREATE TABLE PROYECTO1.DIRECCIONES (

    ID INT NOT NULL,

    PERSONA\_ID INT NOT NULL,

    DISTRITO\_ID INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT DIRECCION\_PERSONA\_ID FOREIGN KEY (PERSONA\_ID)

        REFERENCES PROYECTO1.PERSONAS (ID) ON DELETE CASCADE,

    CONSTRAINT DIRECCION\_DISTRITO\_ID FOREIGN KEY (DISTRITO\_ID)

        REFERENCES PROYECTO1.DISTRITOS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.DIRECCIONES\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.DIRECCIONES\_ID

    BEFORE INSERT ON PROYECTO1.DIRECCIONES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.DIRECCIONES\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

--PROVEEDORES

CREATE TABLE PROYECTO1.PROVEEDORES (

    ID INT NOT NULL,

    DISTRITO\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR(50) NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT PROVEEDOR\_DISTRITO\_ID FOREIGN KEY (DISTRITO\_ID)

        REFERENCES PROYECTO1.DISTRITOS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.PROVEEDORES\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.PROVEEDORES\_ID

    BEFORE INSERT ON PROYECTO1.PROVEEDORES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.PROVEEDORES\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

CREATE SEQUENCE PROYECTO1.PROVEEDORES\_CODIGO\_SEQ;

CREATE TRIGGER PROYECTO1.PROVEEDORES\_CODIGO

    BEFORE INSERT ON PROYECTO1.PROVEEDORES FOR EACH ROW

BEGIN

    SELECT PROYECTO1.PROVEEDORES\_CODIGO\_SEQ.nextval

    INTO :new.CODIGO

    FROM dual;

END;

-- PRODUCTOS

CREATE TABLE PROYECTO1.PRODUCTOS (

    ID INT NOT NULL,

    PROVEEDOR\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    NOMBRE VARCHAR (50) NOT NULL,

    PRECIO\_COMPRA INT NOT NULL,

    PRECIO\_VENTA INT NOT NULL,

    INVENTARIO INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT PRODUCTO\_PROVEEDOR\_ID FOREIGN KEY (PROVEEDOR\_ID)

        REFERENCES PROYECTO1.PROVEEDORES (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.PRODUCTOS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.PRODUCTOS\_ID

    BEFORE INSERT ON PROYECTO1.PRODUCTOS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.PRODUCTOS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

-- FECHAS

CREATE TABLE PROYECTO1.FECHAS (

    ID INT NOT NULL,

    FECHA DATE NOT NULL,

    MES INT NOT NULL,

    NOMBRE\_MES VARCHAR (20) NOT NULL,

    ANNIO INT NOT NULL,

    DIA\_SEMANA INT NOT NULL,

    DIA\_ANNIO INT NOT NULL,

    PRIMARY KEY (ID)

);

CREATE SEQUENCE PROYECTO1.FECHAS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.FECHAS\_ID

    BEFORE INSERT ON PROYECTO1.FECHAS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.FECHAS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

-- VENTAS

CREATE TABLE PROYECTO1.VENTAS (

    ID INT NOT NULL,

    SUCURSAL\_ID INT NOT NULL,

    PRODUCTO\_ID INT NOT NULL,

    CLIENTE\_ID INT NOT NULL,

    EMPLEADO\_ID INT NOT NULL,

    FECHA\_ID INT NOT NULL,

    CODIGO INT NOT NULL,

    CANTIDAD INT NOT NULL,

    TOTAL INT NOT NULL,

    PRIMARY KEY (ID),

    CONSTRAINT VENTA\_SUCURSAL\_ID FOREIGN KEY (SUCURSAL\_ID)

        REFERENCES PROYECTO1.SUCURSALES (ID) ON DELETE CASCADE,

    CONSTRAINT VENTA\_PRODUCTO\_ID FOREIGN KEY (PRODUCTO\_ID)

        REFERENCES PROYECTO1.PRODUCTOS (ID) ON DELETE CASCADE,

    CONSTRAINT VENTA\_CLIENTE\_ID FOREIGN KEY (CLIENTE\_ID)

        REFERENCES PROYECTO1.CLIENTES (ID) ON DELETE CASCADE,

    CONSTRAINT VENTA\_EMPLEADO\_ID FOREIGN KEY (EMPLEADO\_ID)

        REFERENCES PROYECTO1.EMPLEADOS (ID) ON DELETE CASCADE,

    CONSTRAINT VENTA\_FECHA\_ID FOREIGN KEY (FECHA\_ID)

        REFERENCES PROYECTO1.FECHAS (ID) ON DELETE CASCADE

);

CREATE SEQUENCE PROYECTO1.VENTAS\_ID\_SEQ;

CREATE TRIGGER PROYECTO1.VENTAS\_ID

    BEFORE INSERT ON PROYECTO1.VENTAS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.VENTAS\_ID\_SEQ.nextval

    INTO :new.ID

    FROM dual;

END;

CREATE SEQUENCE PROYECTO1.VENTAS\_CODIGO\_SEQ;

CREATE TRIGGER PROYECTO1.VENTAS\_CODIGO

    BEFORE INSERT ON PROYECTO1.VENTAS FOR EACH ROW

BEGIN

    SELECT PROYECTO1.VENTAS\_CODIGO\_SEQ.nextval

    INTO :new.CODIGO

    FROM dual;

END;

**Transferencia al Staging Area**

…

**Transferencia al Datawarehouse**

…

**Estructuras OLAP**

* **Multidimensional**

…

* **Tabular**

…

**Comparativa de las estructuras OLAP**

|  |  |
| --- | --- |
| Sistema Tabular | Sistema Multidimensional |
| Menores tiempos de desarrollo | Modelo basado en Dimensiones y Hechos |
| Alto rendimiento | Soporte para Cubos de gran volumen |
| Características limitadas | Principal almacenamiento es disco duro |
| Principal almacenamiento es memoria | Disponibles en todas las ediciones comerciales |
| Disponible en ediciones BI y Empresarial |  |