



Universidad Hispanoamericana

Ingeniería informática

Aplicación de Bases de Datos

Proyecto Programado

Allan Henry Naranjo Rojas

Presentado por

Erick Andrey Hernández Ugalde 100%

Caleb León Azofeifa 100%

Daniel Alejandro Madrigal Méndez 100%

Agosto, 2024

Resumen Ejecutivo

Este documento presenta el desarrollo del proyecto programado para el curso de Aplicación de Bases de Datos. El objetivo principal es transformar un modelo relacional en un modelo multidimensional utilizando técnicas de minería de datos para la implementación de un *Data Warehouse* (DW). A lo largo del documento, se exploran diversos conceptos clave dentro de este mecanismo de implementación.

Entre los conceptos más relevantes, se aborda la comprensión de las tres estructuras fundamentales del DW: el modelo relacional, el *staging area* y el *data warehouse*, siendo este último donde se obtienen los resultados finales. El proyecto destaca la importancia del proceso ETL (Extract, Transform, Load) en la limpieza y carga de datos desde el *staging area* hacia el DW, con el objetivo de aprovechar los beneficios que ofrece un DW, como:

- Mejora en la toma de decisiones.
- Facilita el análisis de datos.
- Optimización del rendimiento.
- Mantenimiento y escalabilidad.
- Reducción de ambigüedad y redundancia.
- Mejora de la experiencia del usuario.

El desarrollo del proyecto se organizó en varias fases para su avance, seguimiento y retroalimentación, culminando en la entrega final:

1. **Primera Fase:** Análisis inicial del caso de estudio para definir la estructura del modelo multidimensional. Se desarrolló el modelo Entidad-Relación, el *staging area* y el *data warehouse*.
2. **Segunda Fase:** Implementación del proceso ETL, automatizando y preparando la inserción de datos validados en el DW.
3. **Tercera Fase:** Finalización del ETL, realizando validaciones exhaustivas e inserciones para garantizar la calidad de los datos en la base de datos multidimensional, además de gestionar eficazmente los errores sin comprometer la integridad de la información.

El proyecto concluye con un significativo aprendizaje en la construcción de un DW, subrayando la importancia de mantener datos precisos y bien estructurados para optimizar la toma de decisiones en el ámbito empresarial. Este conocimiento es de gran valor para el desarrollo profesional, dada la relevancia y el uso extensivo de los DW en el campo laboral actual.

Las recomendaciones finales incluyen la necesidad de una capacitación previa en ETL y *Data Warehousing*, mantener un orden riguroso en los procesos y considerar los costos asociados al despliegue de un modelo multidimensional a gran escala.

Lista de Contenidos

Resumen Ejecutivo	2
Introducción	5
Desarrollo	6
<i>Avance #1</i>	6
<i>Entidad Relación</i>	6
<i>Staging Area / Data Warehouse</i>	7
<i>Avance #2</i>	10
<i>¿Qué es un ETL?</i>	10
<i>Ventajas de los ETLs:</i>	10
<i>Avance #3</i>	16
<i>La importancia de la claridad de los datos en una base de datos multidimensional:</i> .	16
<i>ETL del Data Warehouse</i>	16
Conclusiones	21
Recomendaciones	22
Capacitación previa.....	22
Orden a la hora de realizar los procesos	22
Costos a la hora de realizar un modelo multidimensional	22
Bibliografía	23

Introducción

Este documento describe el desarrollo del proyecto programado, cuyo objetivo es construir un proceso de minería de datos que transforme un modelo relacional en un modelo multidimensional. Este desafío permite comprender uno de los usos más comunes en la actualidad en la gestión de bases de datos mediante el concepto de *Data Warehouse*. A lo largo del proyecto, se investigará qué es un *Data Warehouse*, cuál es su propósito y por qué es un componente clave en el análisis de datos.

El proyecto se llevará a cabo en tres fases. En la primera fase, se realizará un análisis inicial para comprender el caso de estudio y determinar la estructura adecuada para la creación del modelo multidimensional. En la segunda fase, se explorará el concepto de ETL (Extract, Transform, Load) y su importancia como filtro en la carga de datos hacia el *Data Warehouse*, comenzando por su ingreso al *staging area*. En la tercera fase, se destacará la relevancia de la calidad de los datos, evaluando cómo el ETL final permite depurar los datos, diferenciando aquellos que cumplen con los criterios establecidos de aquellos con errores.

Como resultado de este proyecto, los participantes adquirirán un conocimiento profundo sobre una de las áreas más importantes y aplicadas en el ámbito de las bases de datos, lo cual fortalecerá sus competencias profesionales y permitirá aplicar estos aprendizajes en futuros desafíos laborales.

Desarrollo

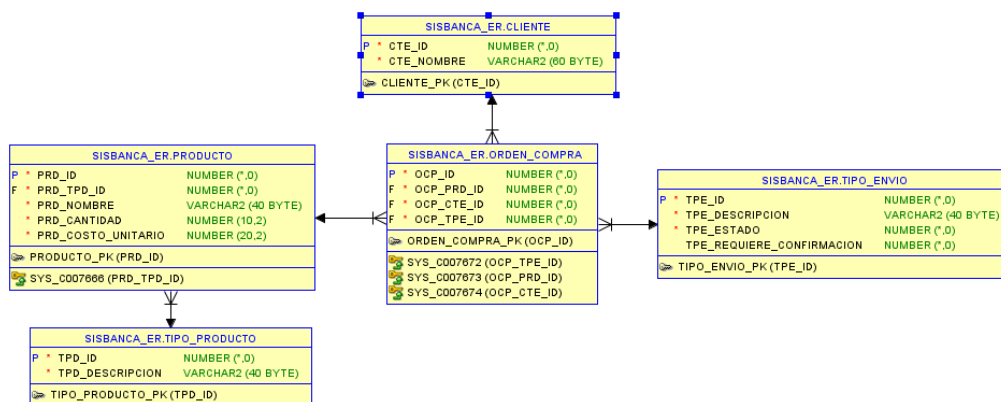
Avance #1

Para el primer avance realizado del proyecto se solicita la creación de los 3 modelos:

1. Entidad Relación
2. Staging Area
3. Data Warehouse

Entidad Relación

El Script para realizar este modelo fue dado con los siguientes datos:



Consta de 5 tablas que contendrán la información cruda que será luego procesada por los distintos ETLs. En esta se pueden ver las tablas de:

- Cliente: Tabla encargada de almacenar todo con respecto al cliente, contiene el identificador como llave primaria y el nombre del cliente.
- Producto: Tabla encargada de almacenar todo con respecto a los productos en venta, contiene el identificador como llave primaria, el identificador de tipo de

producto como llave foránea, y adicional cuenta con el nombre, cantidad y costo unitario como información complementaria.

- Tipo de Producto: Tabla encargada de almacenar todo con respecto a los distintos tipos de producto que se ofrecen, contiene el identificador del tipo de producto como llave primaria conectada únicamente hacia la tabla Producto, y un campo para la descripción.
- Tipo de Envío: Tabla encargada de almacenar todo con respecto a los distintos tipos de envío con los que se cuenta, contiene el identificador del tipo de envío como llave primaria, y un campo para la descripción.
- Orden de Compra: Tabla encargada de almacenar todo con respecto a las distintas órdenes de compra cargadas al sistema, esta tabla comparte información con prácticamente todas las demás tablas del modelo, contiene el identificador de la orden de compra como llave primaria, mientras que contiene como llaves foráneas los campos ID de Producto, ID de Cliente y el ID del tipo de envío

Seguido se realizó el llenado de las tablas de acuerdo con los requisitos previamente solicitados, se realizaron usando el comando INSERT añadiendo los datos requeridos según el enunciado.

Staging Area / Data Warehouse

El siguiente paso en el proyecto fue realizar el modelo del Staging Area y el Datawarehouse, ambos procesos deben llevar precedido el respectivo análisis, que se comentara brevemente a continuación:

1. Para la realización de la tabla de Hechos se tomaron las tablas necesarias del modelo Entidad Relación, se procedieron a ordenar según si contiene o no llave llaves foráneas, las tablas Tipo_Envío, Tipo_Producto, Cliente de primeras seguidas por las tablas Producto y Orden_Compra, estas últimas conteniendo al menos 1 llave foránea.
2. Se procedió luego eliminar campos duplicadas, revisando de izquierda a derecha por campos que puedan existir en otras tablas un ejemplo es el campo de la tabla

Tipo_Envio que contiene el campo TPE_ID el cual también existe en la tabla Orden_Compra como OCP_TPE_ID.

3. Como tercer punto se procedió a eliminar los campos de texto de las tablas, dejando la siguiente distribución de tablas y campos:

TIPO_ENVIO			TIPO_PRODUCTO	CLIENTE	PRODUCTO	
TPE_ID	TPE_ESTADO	TPE_REQUIERE_CONFIRMACION	TPD_ID	CTE_ID	PRD_ID	PRD_CAN
Número	Número	Número	Número	Número	Número	Número

En el paso siguiente se busca la creación de las dimensiones, en la cual se realiza el mismo proceso que para la tabla de hechos con la diferencia de que acá se eliminan los campos numéricos para solo dejar los campos de Texto, esto sin incluir las llaves Primarias

Quedando un planteamiento de las dimensiones de la siguiente manera:

TIPO_ENVIO	
TPE_ID	TPE_DESCRIPCION
Número	Texto

TIPO_PRODUCTO	
TPD_ID	TPD_DESCRIPCION
Número	Texto

CLIENTE	
CTE_ID	CTE_NOMBRE
Número	Texto

PRODUCTO	
PRD_ID	PRD_NOMBRE
Número	Texto

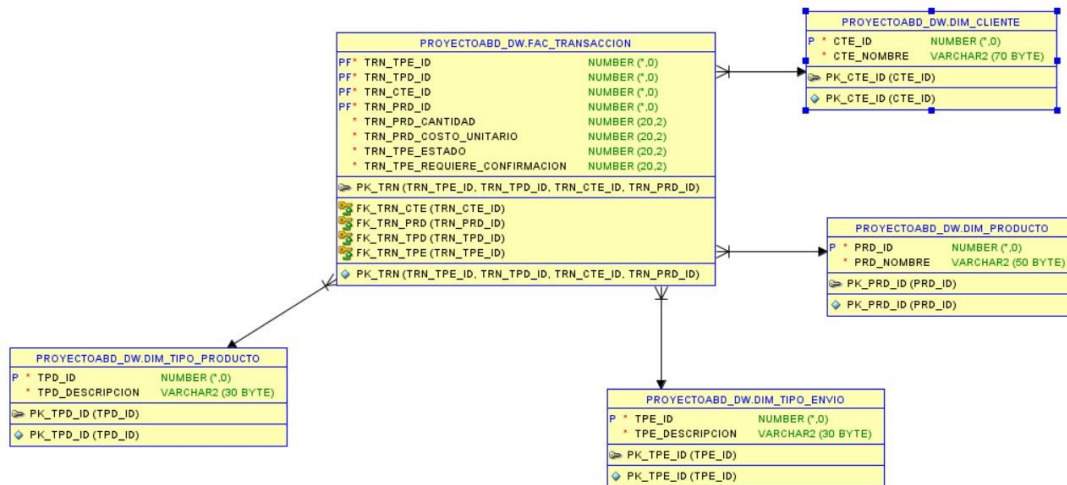
También se crea la tabla de transacciones la cual usaría la información de las dimensiones como llaves foráneas quedando integrada de la siguiente manera:


```

TRN_TPE_ID INTEGER NOT NULL,
TRN_TPD_ID INTEGER NOT NULL,
TRN_CTE_ID INTEGER NOT NULL,
TRN_PRD_ID INTEGER NOT NULL,
TRN_FEC_ID INTEGER NOT NULL,
TRN_PRD_CANTIDAD NUMBER(20,2) NOT NULL,
TRN_PRD_COSTO_UNITARIO NUMBER(20,2) NOT NULL,
TRN_TPE_ESTADO NUMBER(20,2) NOT NULL,
TRN_TPE_REQUIERE_CONFIRMACION NUMBER(20,2) NOT NULL,

```

Para finalizar el modelo del Data Warehouse quedaría conteniendo las siguientes tablas con sus respectivas llaves primarias y foráneas:



Tras finalizar el análisis y la creación de los modelos de tabla de hechos y de dimensiones, se procedió a crear los usuarios para almacenar las tablas según su función:

Usuario SISBANCA_ER contiene las tablas del modelo entidad relación

Usuario SISBANCA_SA contiene las tablas del modelo entidad relación, pero normalizadas a VARCHAR2(255)

Usuario SISBANCA_DW contiene las tablas creadas a partir del modelo entidad relación, son las tablas llamadas dimensiones.

En el script para la creación de las tablas de dimensiones se agregaron validadores de datos los cuales ayudan a asegurar que un dato cumple con lo requerido por ejemplo que los meses sean números únicamente de 1 a 12 o los días solo pueden ser números entre 1 y 31.

Avance #2

Para el Segundo avance se solicitó la creación del ETL del SA.

¿Qué es un ETL?

Para dar un contexto previo a la elaboración del trabajo se formula la pregunta ¿Qué es un ETL? Para esto (Santos, 2023) comenta en su blog “Qué es ETL: definición, proceso y herramientas.” Que el ETL es un método que se utiliza para extraer, limpiar y almacenar datos de diversas fuentes e integrarlos en 1 solo destino o almacén, lo que simplifica su análisis para la toma de decisiones, en este caso utilizamos el ETL del SA para extraer datos desde la base relacional e insertarlos en las tablas creadas en el staging area.

Ventajas de los ETLs:

Según comenta (Pedamkar, 2023) en su artículo “WHAT IS ETL?” las principales ventajas que ofrece ETL son:

1. Fácil de usar.
2. Funcionalidad para manejar errores de forma integrada.
3. Mejora el rendimiento de las bases de datos.
4. Puede cargar diferentes objetivos al mismo tiempo.
5. Puede realizar transformación de datos tanto como sea necesario.

El proceso de la creación del SA inicia con la creación de un package que contiene todos los distintos procedimientos para el correcto funcionamiento del mismo ETL (Extract, Transform and Load).

```
CREATE OR REPLACE PACKAGE PROYECTOABD_DW.ETL_SA AS
    PROCEDURE CARGA_CLIENTE;
    PROCEDURE CARGA_ORDEN_COMPRA;
    PROCEDURE CARGA_PRODUCTO;
    PROCEDURE CARGA_TIPO_ENVIO;
    PROCEDURE CARGA_TIPO_PRODUCTO;
    PROCEDURE INICIA_PROCESO;
END ETL_SA;
/ }
```

En el siguiente código se realiza el package en el usuario PROYECTOABD_DW, que tiene una conexión a la base de datos donde se encuentra el DW, por lo tanto, se almacena ahí el package.

```
PROCEDURE CARGA_CLIENTE AS
    CURSOR C_CLIENTE IS
        SELECT CTE.CTE_ID,
               CTE.CTE_NOMBRE
        FROM PROYECTOABD_ER.CLIENTE CTE
        WHERE NOT EXISTS (SELECT 'X' FROM PROYECTOABD_SA.SA_CLIENTE CSA
                          WHERE CSA.CTE_ID = CTE.CTE_ID)
        ORDER BY CTE.CTE_ID;
```

En el siguiente código se aprecia el body del package anterior, llamado ETL_SA, donde se encuentra el desarrollo de los procedimientos anteriores, dándole funcionalidad al package, el procedimiento “CARGA_CLIENTE” inicia un cursor “C_Cliente” que selecciona los datos que se encuentran en la base de datos ER, es decir la ID y el nombre de la tabla cliente y se asegura de que esos datos no existan en el staging area con antelación, luego lo ordena por su ID.

```

ORDER BY CTE.CTE_ID;
BEGIN
  FOR D_CLIENTE IN C_CLIENTE LOOP
    INSERT
      INTO PROYECTOABD_SA.SA_CLIENTE(CTE_ID, CTE_NOMBRE)
      VALUES(D_CLIENTE.CTE_ID, D_CLIENTE.CTE_NOMBRE);
  END LOOP;
  COMMIT;
END;

```

En este segmento, una vez se comprueba que los datos no están en el SA procede a tomar los datos de la BD relacional y los inserta en las tablas del staging area.

```

PROCEDURE CARGA_ORDEN_COMPRA AS
  CURSOR C_ORDEN_COMPRA IS
    SELECT OCP.OCP_ID,
           OCP.OCP_PRD_ID,
           OCP.OCP_CTE_ID,
           OCP.OCP_TPE_ID
    FROM PROYECTOABD_ER.ORDEN_COMPRA OCP
    WHERE NOT EXISTS (SELECT 'X' FROM PROYECTOABD_SA.SA_ORDEN_COMPRA
                      OSA WHERE OSA.OCP_ID = OCP.OCP_ID)
    ORDER BY OCP.OCP_ID;

```

En el siguiente proceso “Carga_Orden_Compra” vuelve a tomar datos desde la base de datos ER, en este caso la ID del Orden compra, la ID de producto, la ID de cliente, y la ID de tipo envio, luego se asegura que no exista la información en el SA con antelación y lo ordena por la ID de Orden compra.

```

ORDER BY OCP.OCP_ID;
BEGIN
  FOR D_ORDEN_COMPRA IN C_ORDEN_COMPRA LOOP
    INSERT
      INTO PROYECTOABD_SA.SA_ORDEN_COMPRA(OCP_ID, OCP_PRD_ID,
      OCP_CTE_ID, OCP_TPE_ID)
      VALUES(D_ORDEN_COMPRA.OCP_ID, D_ORDEN_COMPRA.OCP_PRD_ID,
      D_ORDEN_COMPRA.OCP_CTE_ID, D_ORDEN_COMPRA.OCP_TPE_ID);
  END LOOP;
  COMMIT;
END;

```

En esta línea se inicia un ciclo for donde se inserta la información en la tabla Orden compra creada en el SA con los valores desde la base de datos ER.

```
PROCEDURE CARGA_PRODUCTO AS
  CURSOR C_PRODUCTO IS
    SELECT PRD.PRD_ID,
           PRD.PRD_TPD_ID,
           PRD.PRD_NOMBRE,
           PRD.PRD_CANTIDAD,
           PRD.PRD_COSTO_UNITARIO
    FROM PROYECTOABD_ER.PRODUCTO PRD
   WHERE NOT EXISTS (SELECT 'X' FROM PROYECTOABD_SA.SA_PRODUCTO PSA
                     WHERE PSA.PRD_ID = PRD.PRD_ID)
   ORDER BY PRD.PRD_ID;
BEGIN
  FOR D_PRODUCTO IN C_PRODUCTO LOOP
    INSERT
      INTO PROYECTOABD_SA.SA_PRODUCTO (PRD_ID, PRD_TPD_ID, PRD_NOMBRE,
      PRD_CANTIDAD, PRD_COSTO_UNITARIO)
    VALUES (D_PRODUCTO.PRD_ID, D_PRODUCTO.PRD_TPD_ID,
            D_PRODUCTO.PRD_NOMBRE, D_PRODUCTO.PRD_CANTIDAD,
            D_PRODUCTO.PRD_COSTO_UNITARIO);
  END LOOP;
  COMMIT;
END;
```

En el siguiente segmento se presenta el procedimiento Carga_Producto, al igual que los segmentos anteriores, recolecta los datos presentados en la base de datos ER, es decir la ID, la id de Tipo producto, y lo que compone la tabla producto, es decir el nombre, la cantidad y el costo unitario, se asegura que no se encuentre en el staging area con antelación y luego inserta los datos de la base de datos relacional a la tabla dentro del SA.

```

PROCEDURE CARGA_TIPO_ENVIO AS
  CURSOR C_TENVIO IS
    SELECT TPE.TPE_ID,
           TPE.TPE_DESCRIPCION,
           TPE.TPE_ESTADO,
           TPE.TPE_REQUIERE_CONFIRMACION
    FROM PROYECTOABD_ER.TIPO_ENVIO TPE
   WHERE NOT EXISTS (SELECT 'X' FROM PROYECTOABD_SA.SA_TIPO_ENVIO TSA
                     WHERE TSA.TPE_ID = TPE.TPE_ID)
   ORDER BY TPE.TPE_ID;

BEGIN
  FOR D_TENVIO IN C_TENVIO LOOP
    INSERT
      INTO PROYECTOABD_SA.SA_TIPO_ENVIO(TPE_ID, TPE_DESCRIPCION,
      TPE_ESTADO, TPE_REQUIERE_CONFIRMACION)
      VALUES(D_TENVIO.TPE_ID, D_TENVIO.TPE_DESCRIPCION,
      D_TENVIO.TPE_ESTADO, D_TENVIO.TPE_REQUIERE_CONFIRMACION);
  END LOOP;
  COMMIT;
END;

```

El mismo proceso se realiza aquí con la tabla tipo_envio, al igual que los segmentos anteriores, recolecta los datos presentados en la base de datos relacional, es decir, la ID, la descripción, el estado y si requiere confirmación, se asegura que no se encuentre en el staging area con antelación y luego inserta los datos de la base de datos relacional a la tabla dentro del SA.

```

PROCEDURE CARGA_TIPO_PRODUCTO AS
  CURSOR C_TPRODUCTO IS
    SELECT TPD.TPD_ID,
           TPD.TPD_DESCRIPCION
    FROM PROYECTOABD_ER.TIPO_PRODUCTO TPD
   WHERE NOT EXISTS (SELECT 'X' FROM PROYECTOABD_SA.SA_TIPO_PRODUCTO
                      DSA
                     WHERE DSA.TPD_ID = TPD.TPD_ID)
   ORDER BY TPD.TPD_ID;
BEGIN
  FOR D_TPRODUCTO IN C_TPRODUCTO LOOP
    INSERT
      INTO PROYECTOABD_SA.SA_TIPO_PRODUCTO(TPD_ID, TPD_DESCRIPCION)
      VALUES (D_TPRODUCTO.TPD_ID, D_TPRODUCTO.TPD_DESCRIPCION);
  END LOOP;
  COMMIT;
END;

```

El penúltimo proceso realiza el mismo procedimiento que los anteriores, esta vez con la tabla tipo producto de la base de datos relacional, recolecta la ID y descripción de tipo producto, comprueba que no esté en el SA e inserta los datos en el mismo.

```

PROCEDURE INICIA_PROCESO AS
BEGIN
  CARGA_CLIENTE;
  CARGA_ORDEN_COMPRA;
  CARGA_PRODUCTO;
  CARGA_TIPO_ENVIO;
  CARGA_TIPO_PRODUCTO;
END;

EXECUTE ETL_SA.INICIA_PROCESO;

```

El último proceso es creado con la finalidad de automatizar todos los procesos anteriores, entonces al ejecutar el proceso “INICIA_PROCESO” comienza a ejecutar cada uno de los procesos por orden con la finalidad de hacer funcionar el ETL.

Avance #3

Para esta última entrega, se nos solicita completar el proyecto realizando como parte final el desarrollo del ETL del Data Warehouse.

La importancia de la claridad de los datos en una base de datos multidimensional:

Con el pasar de los años y el aumento de la utilización de las bases de las empresas, en conjunto con una de sus mayores herramientas la cuál es el análisis de datos, en donde aparece el término tal como explica “Un data warehouse almacena datos consolidados de diversas fuentes o sistemas de la empresa. Se trata de datos estructurados, que tiene como objetivo principal ser precisos y de alta calidad para de esta forma poder dar soporte a la toma de decisiones de la empresa.” (PowerData, 2024), se presenta esta manera de modelar los datos con el objetivo de beneficiar la toma de decisiones y facilitar el análisis de los datos, esto mediante la claridad que brinda este proceso optimizando el rendimiento, permitiendo una mayor facilidad para el mantenimiento, la escalabilidad y evitando la ambigüedad y redundancia.

Por lo cuál mediante el proceso del ETL podemos aplicar un filtro para tener las bases de datos de la manera más clara y eficiente para su correspondiente análisis en la industria, se procederá a ejemplificar en como se realiza este proceso en el proyecto para analizar a como se obtuvieron los datos finales dentro de la base de datos multidimensional, filtrando los posibles errores que pudieran afectar a la calidad de estos.

ETL del Data Warehouse

Para la correcta realización del script se tienen que seguir una serie de pasos para poder pasar y verificar los datos del modelo relacional, pasando por el staging area hasta entrar en el modelo multidimensional.

En una primera parte en necesario es establecimiento de funciones para poder realizar la validación de números tanto enteros como decimales


```

CREATE OR REPLACE FUNCTION VALIDA_NUMERO_ENTERO(P_NUMERO VARCHAR2) RETURN CHAR AS
    V_NUMERO NUMBER;
BEGIN
    V_NUMERO := TO_NUMBER(P_NUMERO);
    IF V_NUMERO = ROUND(V_NUMERO) THEN
        RETURN 'S';
    ELSE
        RETURN 'N';
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 'N';
END;
/

CREATE OR REPLACE FUNCTION VALIDA_NUMERO_DECIMAL(P_NUMERO VARCHAR2) RETURN CHAR AS
    V_NUMERO NUMBER(20,2);
BEGIN
    V_NUMERO := TO_NUMBER(P_NUMERO);
    IF V_NUMERO <> ROUND(V_NUMERO,0) THEN
        RETURN 'S';
    ELSE
        RETURN 'N';
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 'N';
END;
/

```

Posterior a esto se realiza la creación de las tablas de errores, es una parte vital de la verificación del data warehouse, debido a que permite obtener cuales fueron los datos erróneos y el porque de que no fueron cargados en la base de datos de manera satisfactoria pasando por los filtros de los procedimientos que se analizaran en los siguientes puntos.

Para efectos de visualizar cada una de las partes satisfactoriamente, se tomará con guía los procedimientos realizados con la tabla DIM_CLIENTE, que correspondiente a los datos del cliente.

```

CREATE TABLE PROYECTOABD_DW.ERROR_DIM_CLIENTE (
    CTE_ID                VARCHAR2(255),
    CTE_NOMBRE            VARCHAR2(255),
    CTE_ERROR             VARCHAR2(4000)
);

```

En la siguiente sección se declararon todos los procedimientos de migración que estarán contenidos en el paquete del ETL, con cada una de las tablas del modelo.

```
CREATE OR REPLACE PACKAGE PROYECTOABD_DW.ETL_DW AS
    PROCEDURE MigrarDatos;
    PROCEDURE MigrarCliente;
    PROCEDURE MigrarProducto;
    PROCEDURE MigrarEnvio;
    PROCEDURE MigrarTipoProducto;
    PROCEDURE MigrarTransaccion;
END ETL_DW;
/
```

Para este ejemplo como se menciona anteriormente se analizará el procedimiento de migrar cliente, en un primer paso se realiza el llamado de los datos que se van a utilizar procedentes del Staging Area, se valida que el dato no esté posteriormente existente en la tabla, a lo que almacena en un cursor de datos para su validación siguiente.

```
-- Migración de Clientes.

PROCEDURE MigrarCliente IS
    V_ERROR          INTEGER;
    V_NUMERO          INTEGER;
    V_ERROR_MENSAJE VARCHAR2(4000);
    CURSOR C_DATOS IS
        SELECT CTE.CTE_ID,
               CTE.CTE_NOMBRE
        FROM PROYECTOABD_SA.SA_CLIENTE CTE
        WHERE CTE.CTE_ID NOT IN (SELECT D.CTE_ID FROM PROYECTOABD_DW.DIM_CLIENTE D)
        ORDER BY CTE.CTE_ID;
BEGIN
    FOR D_DATOS IN C_DATOS LOOP
        BEGIN
            V_ERROR := 0;
            V_ERROR_MENSAJE := '';

```

Continuando se pasa a una de las partes más importantes y complejas de la migración de datos, donde se aplican diversos filtros para encontrar si pueden ser agregados para cada una de las columnas de la tabla, en la siguiente imagen se valida si el ID del Cliente cumple con que no sea nulo, que tenga un valor numérico y su valor no sea negativo o cero, por lo cuál si fuera así se definiría a la variable V_ERROR con un 1, por lo que no permite su ingreso como dato, también complementando con la variable V_ERROR_MENSAJE con la razón del porque fallo en su inclusión.

```

-- Validaciones del código del cliente

IF D_DATOS.CTE_ID IS NULL THEN
    V_ERROR := 1;
    V_ERROR_MENSAJE := V_ERROR_MENSAJE || 'Código nulo. ';
ELSIF VALIDA_NUMERO_ENTERO(D_DATOS.CTE_ID) = 'N' THEN
    V_ERROR := 1;
    V_ERROR_MENSAJE := V_ERROR_MENSAJE || 'Código no numérico. ';
ELSE
    V_NUMERO := TO_NUMBER(D_DATOS.CTE_ID);
    IF V_NUMERO <= 0 THEN
        V_ERROR := 1;
        V_ERROR_MENSAJE := V_ERROR_MENSAJE || 'Código negativo o cero. ';
    END IF;
END IF;

```

Como parte final en la migración de la tabla de cliente se valida si la variable V_ERROR tiene un valor de 0 y se procede con la inserción de los datos dentro del cursor D_DATOS, si contiene un valor de 1 como se menciona anteriormente se agrega a la tabla de ERROR. para así comunicar al usuario del error en el dato que no permitió su paso al modelo multidimensional.

```

-- Inserción de datos en la tabla de cliente

IF V_ERROR = 0 THEN
    INSERT INTO PROYECTOABD_DW.DIM_CLIENTE (CTE_ID, CTE_NOMBRE)
    VALUES (D_DATOS.CTE_ID, D_DATOS.CTE_NOMBRE);
ELSE
    INSERT INTO PROYECTOABD_DW.ERROR_DIM_CLIENTE (CTE_ID, CTE_NOMBRE, CTE_ERROR)
    VALUES (D_DATOS.CTE_ID, D_DATOS.CTE_NOMBRE, V_ERROR_MENSAJE);
END IF;
EXCEPTION
    WHEN OTHERS THEN
        INSERT INTO PROYECTOABD_DW.ERROR_DIM_CLIENTE (CTE_ID, CTE_NOMBRE, CTE_ERROR)
        VALUES (D_DATOS.CTE_ID, D_DATOS.CTE_NOMBRE, 'Error al insertar');
END;
END LOOP;
END MigrarCliente;

```

Y de última parte se da la creación del procedimiento de MigrarDatos en donde simplemente se contienen las migraciones de todas las tablas para así con una línea de comando se puede efectuar el paso de los datos al data warehouse.

```

PROCEDURE MigrarDatos IS
BEGIN
    MigrarCliente;
    MigrarProducto;
    MigrarEnvio;
    MigrarTipoProducto;
    MigrarTransaccion;
END;
END ETL_DW;
/

```

Así es como se da este proceso del ETL para poder obtener finalmente los datos verificados de la forma más beneficiosa posible, quedando para la facilidades mencionadas anteriormente.

Conclusiones

Realizar este proyecto nos permitió entender de una mejor manera cómo funciona el procesamiento de datos o al menos tener la idea de cómo sería el método correcto y más ágil para aplicar. Realizar este análisis y posterior creación de las bases de datos, aunque engorroso al inicio puede traer muchos beneficios como lo puede ser:

1. Contar con la data limpia y estandarizada en el menor tiempo posible: Las enormes cantidades de datos que pueden ingresar cada minuto para una persona o grupo sería una ardua labor revisar cada ingreso de datos, pero gracias al proceso de validación por ETL's se puede ver agilizado el proceso ayudando a tener la data más exacta, sin errores y dentro de lo requerido.
2. Posibles ataques de inyección: Una forma de ataque es el llamado por inyección, es donde el atacante utiliza los campos a llenar en un formulario como entrada de código, enviando consultas hacia la base de datos para obtener información relevante o puede llegar hasta obtener el control del servidor, por lo cual validar cada campo para que cumpla con un estándar evitaría muchos problemas a futuro
3. Evitar el error humano: Aunque el usuario sea una persona experta en el ambiente puede cometer errores, por lo que automatizar estos procesos evita posibles fallos humanos, además de agilizar el proceso se asegura que la data este blindada ante cambios, permitiendo únicamente que los ETLs tengan contacto con la data y sean los únicos con permisos para insertar/modificar/eliminar información. Dando como fruto data fiable y sin intermediarios.

En un mundo donde la importancia de los datos crece día a día para el desarrollo de las empresas, cada decisión debe estar respaldada por información precisa para evitar errores. Realizar una exhaustiva revisión de los datos y no dar por sentado ninguna variable que pueda influir en los resultados puede marcar la diferencia. Como se mencionó anteriormente, un mal procesamiento de datos puede llevar a la ruina a una empresa por el simple hecho de no cuidar su activo más valioso en términos tecnológicos la información.

Recomendaciones

Capacitación previa

Como se explicó anteriormente en las conclusiones, es normal que exista el error humano, por lo que una capacitación previa en los procesos de ETL, creación de un data warehouse, staging area y todos estos procesos anteriores es de suma importancia.

Orden a la hora de realizar los procesos

Es de suma importancia que a la hora de realizar una base de datos multidimensional se mantenga un orden en cada uno de los pasos, con el fin de que se pueda ejecutar cada una de las partes con corrección, especialmente en la creación del ETL para el SA y el DW, donde se realizan cada una de las comprobaciones y un error mínimo podría ser nefasto para la BD.

Costos a la hora de realizar un modelo multidimensional

Como última recomendación, es conocer la viabilidad de los modelos multidimensionales, es decir, ser conscientes de los beneficios que trae las bases de datos de este estilo, ya que al tener protocolos para el manejo de errores de manera integrada y estar muchos más optimizadas sus costos pueden ser mejores que las bases de datos relacionales.

Bibliografía

Pedamkar, P. (19 de mayo de 2023). *Educba*. Obtenido de educba:
<https://www.educba.com/what-is-etl/>

Santos, D. (31 de mayo de 2023). *HubSpot*. Obtenido de blog.hubspot:
<https://blog.hubspot.es/marketing/que-es-etl>

PowerData (2024). Powerdata.es. <https://www.powerdata.es/data-warehouse#:~:text=Un%20Data%20Warehouse%20es%20un,recuperar%20y%20f%C3%A1cil%20de%20administrar.>