



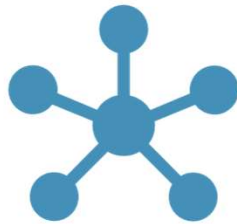
PROYECTO FINAL

ABRIL

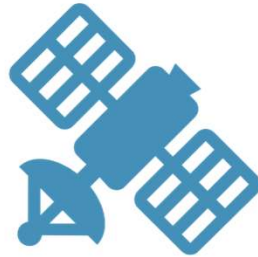
INTRODUCCIÓN

Con la información que esta subida en la plataforma deberán de crear la estructura de tablas que necesitaran para crear el DW del proyecto final en cada Ambiente, estará dividido en varias fases semanales comenzando en abril y a partir de este día tendrán que hacer sprint de trabajos semanales para lograr finalizar el proyecto.

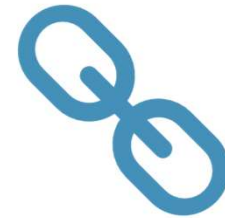
Deberá de Crear Tres ambientes de trabajo los cuales se tienen que sincronizar uno con el otro



1.- Stage



2.- DW



3.- Producción

REQUISITOS DEL PROYECTO

FASE CERO DEL PROYECTO.

Crear o ejecutar los siguientes archivos que están Subidos en Canvas.

- Usuarios (pueda dar error, capturar el error y cambiar el USERS de table Space)
- Creación de Tres ambientes
 - Tablas de Stage (usar el Usuario de Stage)
 - Tablas de DW (usar el Usuario de DW)
 - Tablas de Producción (usar el Usuario de Producción)
 - Inserción de Registros de Producción

FASE CERO PROYECTO

Los Archivos Excel serán utilizados para llenar la información correspondiente.

Deberán de investigar e instalar la función SQL LOUDER para cargar la información que esta en los archivos Excel y cargarlas a sus respectivas tablas, recuerde que son 3 ambientes.

Investigar La función Merge en SQL.

Investigar el tema de Lockup como concepto para poder aplicar en Sql.

Crear Las estructuras de Tablas en cada uno de los ambientes.

Recordarse de darle permisos de update y select al usuario stage en producción. Y permisos de lectura a todas las demás tablas.

Trigger tablas producción, El Trigger escucha cuando se hace un select o update deberá de llenar la tabla Watermark. A traves del Trigger .TRG_TABLA_SINC en ambiente producción.

Procedimiento PRC_SINCRONIZACION en ambiente STAGE. Permiso de select y update a con esto se debe de llenar la tabla Watermark, cuando se hace update, debe de borrar los datos en Stage y cargar nuevamente la información-. Recordase que al momento de hacer un update sobre la tabla borrar en Stage e insertar el registro nuevamente.

FASE UNO PROYECTO

Sincronización de Información entre ambientes.

- Crear un Procedimiento para este manejo... `PRC_DEVUELVE_CORRELATIVOS` (consulta en la tabla `control_numeracion` y que devuelva el que es, si se envía una que no exista se debe de insertar y que inicie con uno y luego se le va sumando uno cada vez que la encuentre) recordar que el procedimiento tiene dos parámetros uno de entrada (nombre de la dimensión) y otro de salida (El valor). Esto en Ambiente DataWhareHouse.
- Crear Triggers (a nivel de `INSERT` unicamente) en cada dimensión para el manejo de `KEY'S`. Basadas en el procedimiento `PRC_DEVUELVE_CORRELATIVOS`. Ambiente DW. Se envía como parámetro el nombre de la dimensión.

FASE UNO DEL PROYECTO.

- Crear un procedimiento por cada dimensión que exista, para llenar todas las tablas de dimensiones. (este deberá de llevar como Parámetro 1 – UTILICE MERGE), Ambiente DataWhereHouse) deberá de tener Permisos de select a las tablas de Stage. Cuando hacen select a las Tablas de Stage y el destino de este select son las dimensiones y se unen a través del Cod (Llaves primarias del transaccional), Dar permisos al usuario DW para que pueda realizar el proceso, sobre las tablas del Stage.
- El Procedimiento llenará cuando el parámetro sea 1 y se estará grabando en la tabla de HECHOS (PRC_CONSTRUYE_HECHOS del lado de DW), Parámetro 1. Stage_TRANSACCION LEFT JOIN Todas las dimensiones por medio de los COD. Ambiente DataWhereHouse. Lleno tabla de hechos (todos los key estén llenos si hay un key nulo se llena la tabla de no_hechos (key nulos). Cuando se dispare el Trigger ira a llenar los N cantidad de Keys.
- Llenado de tabla NO_HECHOS, Esta debe de llevar KEY'S Y Datos de Stage que son los COD.
- En el caso de la dim_tiempo proceder a insertar el valor en formato YYYYMMDD

EJEMPLO-

```
CREATE PROCEDURE PRC_DIM_PATITO
BEGIN
MERGE INTO DIMENSION D
USING (SELECT * FROM STAGE) F
ON (D.PK_TRANSACCIONAL = F.PK_TRANSACCIONAL) -- REGULA
WHEN MATCHED THEN
UPDATE SET
DSC
WHEN NOT MATCHED THEN
INSERT COD, DESCRIPCION
VALUES(F.empno, F.first_name);
END;
```

```
CREATE PROCEDURE PRC_DIM_PATITO
BEGIN
IF PARAMETRO = 1 THEN
MERGE INTO DIMENSION D
USING (SELECT * FROM STAGE) F
ON (D.PK_TRANSACCIONAL = F.PK_TRANSACCIONAL)
WHEN MATCHED THEN
UPDATE SET
DSC
WHEN NOT MATCHED THEN
INSERT COD, DESCRIPCION
VALUES(F.empno, F.first_name);
END IF;
END;
```

```
PROCEDURE PRC_CONSTRUYE_HECHOS(PARA IN INTEGER)
CURSOR
SELECT T.CORRELATIVO, T.FECHA, T.HORA_OCURRENCIA, D_CAT_HOR_OCC.HORA_KEY
FROM STG_TRANSACCION T LEFT JOIN DIM_CAT_HORA_OCURRENCIA D_CAT_HOR_OCC ON (T.HORA_OCURRENCIA = D_CAT_HOR_OCC.ID_HORA)

BEGIN
IF PARA = 1 THEN
ABRO EL CURSOR
RECORRO.
SI TODOS LOS VALORES KEY SON DIFERENTES A NULOS.
INSERT TABLA DE HECHOS.
SI ENCONTRE POR LO MENOS UN VALOR DONDE KEY ES NULO.
INSERT TABLA DE NO_HECHOS.
CIERRO EL CURSOR.
END IF;
END;
```

Buscar

Buscar: DIM_CAT_COLOR_VEHICULO

Buscar siguiente

Cancelar

EJEMPLO-

```
PROCEDURE PRC_CONSTRUYE_HECHOS(PARA IN INTEGER)
CURSOR
SELECT T.CORRELATIVO, T.FECHA, T.HORA_OCURRENCIA, D_CAT_HOR_OCC.HORA_KEY
FROM STG_TRANSACCION T LEFT JOIN DIM_CAT_HORA_OCURRENCIA D_CAT_HOR_OCC ON (T.HORA_OCURRENCIA = D_CAT_HOR_OCC.ID_HORA)

BEGIN
    IF PARA = 1 THEN
        ABRO EL CURSOR
        RECORRO.
        SI TODOS LOS VALORES KEY SON DIFERENTES A NULOS.
            INSERT TABLA DE HECHOS.
        SI ENCONTRE POR LO MENOS UN VALOR DONDE KEY ES NULO.
            INSERT TABLA DE NO_HECHOS.
        CIERRO EL CURSOR.
    END IF;
END;
```



A screenshot of a search dialog box titled "Buscar". It features a search input field containing the text "DIM_CAT_COLOR_VEHICULO". To the right of the input field is a button labeled "Buscar siguiente". Below the input field, there are two buttons: "Definición" and "Cancelar". In the top right corner of the dialog, there is a small icon resembling a magnifying glass.

EJEMPLO-

MERGE. Ejemplo.

- PRC_DIM_TRIPLE_XXX(PARAMETRO)
BEGIN
IF PARAMETRO = 1 THEN
BEGIN
SELECT STAGE FUENTE.
DESTINO. LAS DIMENSIONES.
ON (COD.)
END
END

- PRC_CONSTRUYE_HECHOS(PARAMETRO)
BEGIN
IF PARAMETRO = 1 THEN
BEGIN
SELECT STAGE_TRANSACCION T
LEFT JOIN DIM_1 D1 ON (T.COD1 = D1.COD1)
LEFT JOIN DIM_2 D2 ON (T.COD2 = D2.COD2)
LEFT JOIN DIM_3 D3 ON (T.COD3 = D3.COD3)
LEFT JOIN DIM_3
LEFT JOIN DIM_4
END;
END;

FASE UNO DEL PROYECTO.

- Crear un procedimiento para llenar la tabla de hechos (parámetro = 1). esto deberá de trabajarse con un cursor y validar que todos los datos que se deban de llenar (KEY's) existan. Si existen deben de insertarlo en la tabla de hechos, X un dato que no exista deberá de insertarlo en la tabla de no hechos.
- El cursor debe de hacerse entre la tabla de Stage -- left join las dimensiones.
- Tomando en cuenta las dimensiones que aparecen en la tabla de hechos.
- Luego debe de insertarse las metricas. (archivo Excel en negrilla).
- los valores tal y como estan en la tabla stage.

FASE DOS DEL PROYECTO

(Como recordatorio a la primera clase) Usted debió haber creado un procedimiento para realizar el seguimiento de las tablas dimensiones.


En el caso de la dim_tiempo proceder a insertar el valor en formato YYYYMMDD

Ejemplo:

```
PRC_LLENA_SEGUIMIENTO(PARAMETRO IN VARCHAR)
BEGIN
    INSERT INTO BD2_SEGUIMIENTO VALUES (PARAMETRO, USER, SYSDATE);
END;
```

FASE DOS DEL PROYECTO

- Modificar el procedimiento que llena la tabla Hechos: Procedimiento de HECHOS (PRC_CONSTRUYE_HECHOS), Parámetro 2. Basado en Tabla NO_HECHOS LEFT JOIN DIMENSIONES. Ambiente DW.
- Llenado DIMENSION tiempo, procedimiento PRC_DIM_TIEMPO.
- Anteriormente se creó el procedimiento que recibía como parámetro el valor de 1, ahora se enviará el parámetro 2, el cual deberá de realizar el left join pero esta vez a la tabla de No_Hechos (en lugar de la de Stage), e insertar la información que se encuentre.
- Es una copia del parámetro 1 pero ahora a la tabla de No_hechos y no se validan Keys nulos, se insertan los valores a la tabla de hechos.



```
PRC_XXXXXXXXXX(PARAMETRO IN NUMBER)
BEGIN
    IF PARAMETRO = 1 THEN
        MERGE -- STAGE
    ELSE
        NULL;
    END IF;
    PRC_LLENA_SEGUIMIENTO('DIM_XXXXXX');
END;
```

```
PRC_XXXXXXXXXX(PARAMETRO IN NUMBER)
BEGIN
    IF PARAMETRO = 1 THEN
        MERGE -- STAGE
    ELSIF PARAMETRO = 2 THEN
        MERGE -- NO_HECHOS
    END IF;
    PRC_LLENA_SEGUIMIENTO('DIM_XXXXXX');
END;
```

FASE DOS DEL PROYECTO

- Crear un procedimiento para que vaya a leer a la tabla de no_hechos e indique que dimensiones tiene que procesar, comparando key nulos. si encuentra un nulo debe de Enviarse a llamar el procedimiento de esa dimensión con parámetro 2 (recordarse solo llamar a los key nulos)

```
TRIGGER DE LA TABLA DE HECHOS.
```

```
IF :NEW.CANTIDAD_PERSONAS IS NULL THEN  
    :NEW.CANTIDAD_PERSONAS := FNC_DEVUELVE_VALOR_DEFAULT('CANTIDAD_PERSONAS');  
END IF
```

```
IF :NEW.GTOS_CALC_PER IS NULL THEN  
    :NEW.GTOS_CALC_PER := FNC_DEVUELVE_VALOR_DEFAULT('GTOS_CALC_PER');  
END IF
```

```
IF :NEW.GTOS_CALC_BIE IS NULL THEN  
    :NEW.GTOS_CALC_BIE := FNC_DEVUELVE_VALOR_DEFAULT('GTOS_CALC_BIE');  
END IF
```

```
PRC_DIM_HORA_OCURRENCIA(PARAMETRO)
```

```
BEGIN  
    IF PARAMETRO = 1 THEN  
        MERGE. FUENTE STAGE.  
        DESTINO. DIMENSION.  
    END IF;
```

```
END;
```

```
PRC_DIM_HORA_OCURRENCIA(PARAMETRO)
```

```
BEGIN  
    IF PARAMETRO = 1 THEN  
        MERGE. FUENTE STAGE.  
        DESTINO. DIMENSION.  
    ELSIF PARAMETRO = 2 THEN  
        MERGE. FUENTE STAGE. (ESTO ES LO QUE CAMBIA)  
        DESTINO. DIMENSION.  
    END IF;  
END;
```



```

PRC_DIM_HORA_OCURRENCIA(PARAMETRO)
BEGIN
    IF PARAMETRO = 1 THEN
        MERGE. FUENTE STAGE.
            DESTINO. DIMENSION.
    ELSIF PARAMETRO = 2 THEN
        MERGE. FUENTE STAGE. (ESTO ES LO QUE CAMBIA), JOIN ENTRE LA TABLA DE NO HECHOS VRS. TABLA DEL STAGE.
            DESTINO. DIMENSION.
    END IF;
END;

```

```

14
15 PRC_DIM_HORA_OCURRENCIA(PARAMETRO)
16 BEGIN
17     IF PARAMETRO = 1 THEN
18         MERGE. FUENTE STAGE.
19             DESTINO. DIMENSION.
20     ELSIF PARAMETRO = 2 THEN
21         MERGE. FUENTE STAGE. (ESTO ES LO QUE CAMBIA), JOIN ENTRE LA TABLA DE NO HECHOS VRS. TABLA DEL STAGE.
22             DESTINO. DIMENSION.
23     END IF;
24
25     REPROCESA_DIMENSIONES_QA('DIM_HORA_OCURRENCIA', PARAMETRO);
26
27 END;
28
29 -----
30 PROCEDIMIENTO REPROCESA_DIMENSIONES_QA(NOMBRE_DIMENSION, PARAMETRO)
31 BEGIN
32     INSERT INTO REPROCESA_DIMENSIONES_QA VALUES (NOMBRE_DIMENSION, SYSDATE, PARAMETRO);
33 END;
34

```

FASE DOS DEL PROYECTO

```

37
38 PROCEDIMIENTO PRC_CONSTRUYE_HECHOS(PARAMETRO)
39 CURSOR TRANSACCIONAL "LEFT JOIN" TODAS LAS DIMENSIONES.
40 CURSOR NO_HECHOS "LEFT JOIN" TODAS LAS DIMENSIONES.
41 BEGIN
42     IF PARAMETRO = 1 THEN
43     BEGIN
44         OPEN CURSOR.
45         RECORREN EL CURSOR.
46         HAY VALORES KEY NULOS ?.
47         INSERTAN EN LA TABLA DE NO_HECHOS.
48         SI NO HAY VALORES KEY NULOS.
49         INSERTAN EN LA TABLA DE HECHOS.
50         CIERRAN EL CURSOR.
51     END
52
53     ELSIF PARAMETRO = 2 THEN
54     BEGIN
55         OPEN CURSOR
56         RECORREN EL CURSOR
57         INSERTAN EN LA TABLA DE HECHOS
58         CIERRAN EL CURSOR
59     END
60     END IF
61 END
62

```



GRACIAS

ALGUIEN@EJEMPLO.COM