Desarrollo Squeesar – Observaciones y Consideraciones

# Introducción.

EL siguiente documento explica brevemente como se desarrolló la solución para el procesamiento de la data cargada de Squeesar en la Base de Datos.

# Procesos.

En la solución se pueden identificar 5 procesos.

**I Carga de Información – Nueva tablas**

La nueva información de Squeesar deben ser cargada en el Esquema “squeesar”, y la tabla debe tener un formato específico:

squeesar\_[num\_squessar]\_[direccion]\_ [numero\_de\_parte]

Donde:

* [num\_squessar]: obligatorio. Parte numérica, indica el numero del squeesar (ejemplo: 6)
* [direccion]: obligatorio. Debe indicar la dirección (asc para ascendente; desc para descendente) de los datos
* [numero \_de\_parte]: opcional. En caso de que la información total se haya dividido en 2 o más parte, este segmento se utilizara para identificar cada una de las partes; es solo referencial, por lo que se puede utilizar como el usuario estime conveniente.

Ejemplo

1. squeesar\_6\_asc
2. squeesar\_7\_asc\_parte1
3. squeesar\_7\_desc\_parte1
4. squeesar\_10\_asc\_uno\_de\_dos

**II Registro de nuevas Tablas**

Se ha creado la función registrar\_tabla\_squeesar() que tiene por finalidad llevar un registro de todas las tablas cargadas en el sistema, en la tabla “registro\_squeesar”.

El proceso es manual, y puede ejecutarse una vez se ha terminado de cargar toda la nueva información.

Esta funcionalidad consulta en el esquema “squeesar” todas las tablas que existan, que no hayan sido registradas con anterioridad en la tabla “registro squeesar” (para no duplicar información) y que comiencen con la palabra “squeesar” según formato anterior descrito. Por esto último, es que como política interna no se pueden crear nuevas tablas que comiencen con la palabra squeesar en este esquema, a menos que sea estrictamente necesario, recordando si es así, incluirla en una lista de excepciones en la misma función.

Con la nueva información obtenida, se comienza a poblar la tabla “registro\_squeesar” donde además de los campos llave, y composición del nombre de la tabla, se registra con estado “NUEVO”, estado que será utilizado por el siguiente proceso.

**III Procesamiento de información masiva**

Se ha creado la función procesar\_squeesar(). Esta función comienza consultando de la tabla “registro\_squeesar” todos los registros que se encuentren en estado “NUEVO”, es decir, aquellos que aún no han sido procesados, además de obtener el squeesar que actualmente está marcado como vigente (es decir, está siendo utilizado para las intersecciones con polígonos) desde la misma tabla.

Con la información obtenida, se crean nuevas tablas “historicas” a partir de 2 tablas vacías en el sistema: “template\_squeesar\_consolidado” y “template\_squeesar\_consolidado\_fecha”; a partir de estas tablas se creara una nueva tabla consolidada de manera dinámica; el nombre de la tabla será con el siguiente formato:

historico\_squeesar\_[num\_squessar] - historico\_squeesar\_[num\_squessar]\_fecha

1. **Función preparar\_tabla**

Una vez creada las nuevas tablas, y dado que podría un squeesar estar dividido en más de una tabla (partes) lo primero es modificar las tablas originales, agregando un identificador único; una nueva columna con un correlativo único para cada registro de todas las partes del mismo squeesar.

1. **Función procesar\_tabla**

La finalidad de esta función es separar la tabla original, en las dos tablas nuevas recién creadas, donde las columnas con datos de fecha pasan a ser filas en la tabla “historio\_squeesar[numsqueesar]\_fecha”. Por cada tabla ya modificada con el correlativo único, se toman cada una de las columnas de la tabla a procesar (ejem: squeesar\_7\_asc\_parte1), y se verifica si esta columna existe en la nueva tabla (ejem: historico\_squeesar\_7); si la columna existe, se prepara un string para luego ejecutarlo e insertar en la tabla correspondiente (en nuestro ejemplo, historico\_squeesar\_7). Si la columna no existe, se asume por defecto que es una columna de tipo fecha que se convertirá en una fila en la tabla de fecha (historico\_squeesar\_7\_fecha), por lo tanto se le da formato de tipo fecha, generando también el string que será ejecutado para insertar en la tabla correspondiente; si no es posible darle formato fecha, arrojará error, pudiendo ser por algunos de los siguientes motivos:

* Nombre de columna no está en el formato correcto de fecha esperado Se espera: “dYYYYMMDD” donde

d: variable fija, indica que es un date.

YYYY: año

MM: mes

DD dia

Ejemplo: d20150125

* Columna nueva en tabla, no existente en el template.

1. **Función intersecta\_squeesar\_poligonos**

Una vez finalizado el procesamiento de cada una de las tablas (cuyo procesamiento fue en orden ascendente), se verifica cual es el último squeesar procesado, y se compara con el que se encuentra marcado como VIGENTE en la tabla registro squeesar; esto es porque, eventualmente, puede llegar información de squeesar anteriores, los que se necesitan procesar pero no intersectar, por lo tanto si el número del último squeesar procesado es menor al marcado como vigente, la función intersecta\_squeesar\_poligonos no es invocada; en cambio , si es mayor, debe intersectarse poligonos con squeesar. El resultado de esta intersección se inserta en la tabla “poligono\_squeesar” del esquema “poligonos”.

**IV Procesamiento data final**

Con la información de intersección ya procesada, es necesario ejecutar el proceso “agrupar\_squeesar()”; este proceso toma la información de las tablas vigentes (marcadas en la tabla “registro\_squeesar”) agrupando por polígono la información necesaria, siendo insertada en las tablas finales “poligono\_resumen” y “poligono\_fecha”.

**V Creación, modificación y eliminación de Poligonos**

Existen trigger en las tablas “polígono” y “points” del esquema “poligonos” para el caso de modificaciones de estas tablas, para procesar o eliminar datos, según corresponda, a partir de la intersección y la data final, pero solo para ese polígono.