**Consejo de diseño 142 Construyendo puentes**

Este consejo de diseño continúa nuestras series de cómo implementar patrones comunes de diseño dimensional en tu sistema ETL. La relación entre una tabla de hechos y sus dimensiones es normalmente muchos a uno. Eso significa, una fila en una dimensión, como un cliente, puede tener muchas filas en la tabla de hechos, pero una fila en la tabla de hechos debe pertenecer a sólo un cliente. Sin embargo, hay veces que una fila de la tabla de hechos puede estar asociada con más de un valor en una dimensión. Utilizamos un puente para recoger esta relación muchos a muchos.

Hay dos clases principales de tablas puente. La primera, y la más fácil de modelar, recoge conjuntos simples de valores asociados con una única fila de hechos. Por ejemplo, un registro de ingreso en la sala de emergencias puede tener uno o más diagnósticos iniciales de la enfermedad asociados. No hay variación de tiempo en esta tabla puente porque recoge el conjunto inicial de valores que estaban en vigor cuando la transacción ocurrió.

El segundo tipo de relación muchos a muchos existe independientemente de las transacciones medidas. La relación entre cliente y cuenta es un buen ejemplo. Un cliente puede tener una o más cuentas, y una cuenta puede pertenecer a uno o más clientes, y esta relación puede variar en el tiempo.

Este consejo de diseño indica los pasos para crear una tabla puente simple y estática, este enfoque puede extenderse para dar apoyo a una tabla puente más compleja que incluya la variable tiempo.

**Carga histórica**

Los pasos involucrados en crear la tabla puente histórica dependen en cómo se recogen los datos del sistema fuente. Suponemos que la fuente recoge los datos que asocian el conjunto de dimensión IDs multivalorado con cada transacción ID. En el ejemplo de diagnóstico, esto sería una tabla con una fila para cada registro de admisión ID y diagnóstico ID. Observa que esta tabla tendrá más filas que el número de admisiones individuales.

**Crear la lista de grupos inicial**

Desde que la fuente se normalice y tiene una fila por valor dimensional, el primer paso es crear una lista única del grupo de diagnóstico que tiene lugar en la tabla de transacciones. Esto implica agrupar los conjuntos de diagnósticos, decuplicar la lista de grupo, y asignar una clave única a cada grupo. Normalmente esto es más fácil de hacer en SQL creando una nueva tabla que soporte la lista de grupos. Utilizando el ejemplo de diagnostico, la Figura 1 muestra como las primeras filas del sistema transaccional se agruparán y deduplicarán en la tabla Grupo de diagnóstico.

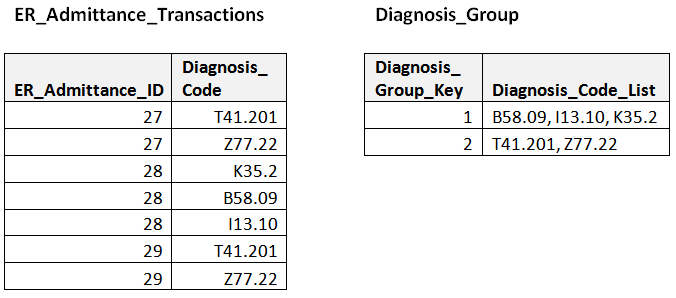


Figura 1 – Datos de transacción de origen y tabla asociada de grupo de diagnóstico.

Hay muchos modos de hacer esto en los diversos dialectos del SQL. La siguiente versión utiliza un enfoque de agregación de cadena basado en el servidor SQL, utilizando la función de la cadena STUFF() para hacer la concatenación, FOR XML PATH para colapsar las filas en cada conjunto de códigos de la transacción, SELECT DISTINCT para crear la lista de grupos únicos, y la función Row\_Number() para asignar la clave de grupo.

SELECT Row\_Number() OVER ( ORDER BY Diagnosis\_Code\_List) AS  
Diagnosis\_Group\_Key, Diagnosis\_Code\_List  
INTO Diagnosis\_Group  
FROM(  
SELECT DISTINCT Diagnosis\_Code\_List  
FROM  
(SELECT DISTINCT OuterTrans.ER\_Admittance\_ID,  
STUFF((SELECT ‘, ‘ + CAST(Diagnosis\_Code AS VARCHAR(1024))  
FROM ER\_Admittance\_Transactions InnerTrans  
WHERE InnerTrans.ER\_Admittance\_ID = OuterTrans.ER\_Admittance\_ID  
ORDER BY InnerTrans.Diagnosis\_Code  
FOR XML PATH(”)),1,2,”) AS Diagnosis\_Code\_List  
FROM ER\_Admittance\_Transactions OuterTrans  
) OuterList  
) FinalList;

Asegúrate de que tu version de este código mantiene la lista de grupos en un orden consistente; en algunos casos el grupo ‘T41.201, Z77.22’ es la misma que el grupo ‘Z77.22, T41.20’. En sanidad, el orden normalmente importante y ambos grupos serán creados, el sistema fuente proporcionará un número de secuencia que deberá ser incluido en el orden interno y pasado a través de la tabla puente. Observa que la tabla de grupos de diagnóstico es más corta que la tabla fuente original. Esto se debe en parte a que hemos reducido el conjunto de filas, pero también porque el número de combinaciones de valores utilizado en la vida real es normalmente menor que las combinaciones teóricas posibles.

**Crear la tabla puente**

Una vez que hemos hecho el trabajo de crear la tabla de Grupos de diagnóstico y asignado las claves de grupo, necesitamos no pivotar para crear la tabla puente de diagnóstico actual. Esta es la tabla que mapea cada grupo en las filas de dimensiones individuales desde las cuáles se define. La figura 2 muestra la tabla puente de diagnóstico y la tabla de dimensión de diagnóstico asociada ICD10\_Diagnosis basada en nuestros datos de ejemplo.

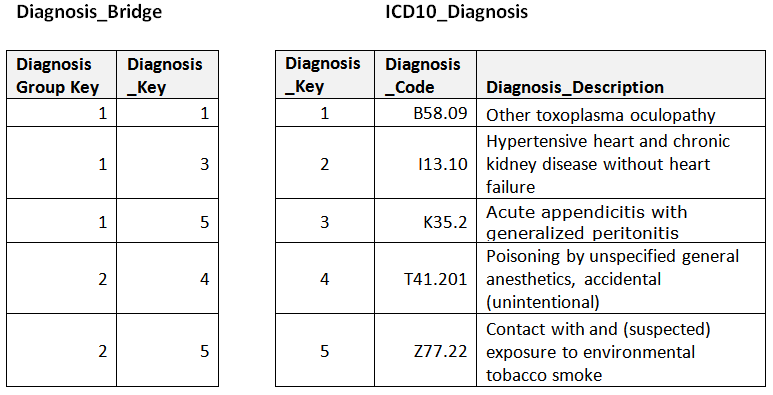


Figura 2 – Tabla de diagnostico puente y dimensión de diagnóstico asociada ICD10\_Diagnosis

Otra vez, hay muchos modo de realizar esto en SQL. La siguiente versión en servidor SQL T-SQL utiliza un proceso de dos pasos para no pivotar la tabla de grupo. El primer paso concatena las etiquetas XML (<I> </I>) en el Diagnosis\_Code\_List y lo convierte en un tipo de datos XML como parte de una tabla de expresión común. El segundo paso utiliza el comando CROSS APPLY para analizar las etiquetas XML y enumerar los valores en líneas separadas.

WITH XMLTaggedList AS (  
SELECT Diagnosis\_Group\_Key,  
CAST(‘<I>’ + REPLACE(Diagnosis\_Code\_List, ‘, ‘, ‘</I><I>’) + ‘</I>’ AS XML)  
AS Diagnosis\_Code\_List  
FROM Diagnosis\_Group  
)  
SELECT Diagnosis\_Group\_Key,  
ExtractedDiagnosisList.X.value(‘.’, ‘VARCHAR(MAX)’) AS Diagnosis\_Code\_List  
FROM XMLTaggedList  
CROSS APPLY Diagnosis\_Code\_List.nodes(‘//I’) AS ExtractedDiagnosisList(X);

**Procesamiento incremental**

El proceso incremental de carga aplica esencialmente el mismo código para las filas de hechos entrantes. El primer paso genera una lista Diagnosis\_Code\_List para cada grupo de filas de hechos entrantes en el granulado de la tabla de hechos. Este valor de la Diagnosis\_Code\_List puede ser utilizado para unir la tabla de Diagnosis\_Group para mapear de filas de hechos entrantes con la clave de grupo apropiada. Si un nuevo grupo surge, el proceso ETL necesitará añadirlo a la tabla Diagnosis\_Group, como lo hace para las mini-dimensiones y las dimensiones superfluas.

Hemos utilizado SQL para ilustrar el patrón de diseño utilizado para crear una tabla puente. La misma lógica puede ser implementada en la sintaxis nativa y controlar estructuras de la mayor parte de las herramientas ETL. Por supuesto, sería ideal si los desarrolladores de herramientas ETL incluyesen tablas puente de procesamiento en sus conjuntos de herramientas más que forzarnos a reinventar la rueda. Tal vez, lo incluyan en el próximo lanzamiento. Si te gustaría tener el script SQL de este ejemplo, mándame un mail.

Observación: Este consejo de diseño se ha actualizado del original para incluir una clave subrogada en la tabla ICD10\_Diagnosis

<http://www.kimballgroup.com/2012/02/design-tip-142-building-bridges/>