

Artículo

[Javier Lorenzo Mesa](#) · 8 jul, 2022 Lectura de 4 min

DeepSee: Bases de datos, Namespaces y Mapeos - Parte 3 de 5

Este artículo describe un diseño arquitectónico de complejidad intermedia para DeepSee. Al igual que en el [ejemplo anterior](#), esta implementación incluye bases de datos separadas para almacenar la información, la implementación y la configuración de DeepSee. También presenta dos nuevas bases de datos: la primera para almacenar los globals necesarios para la sincronización, la segunda para almacenar tablas de hechos e índices.



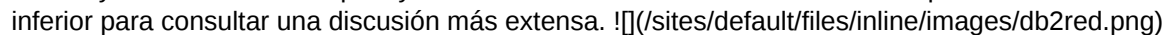
Ejemplo 2: Un diseño más flexible

Bases de datos

Además de las bases de datos APP-CACHE y APP-DEEPSEE que vimos en el ejemplo anterior, definimos las bases de datos APP-DSTIME y APP-FACT.

La base de datos APP-DSTIME contiene los globals ^OBJ.DSTIME y ^DeepSee.Update de sincronización en DeepSee. Estos globals se replican desde una base de datos (journaled) en el servidor de producción. Ten en cuenta que la base de datos APP-DSTIME debe ser de lectura-escritura en las versiones de caché que utilizan

^DeepSee.Update.

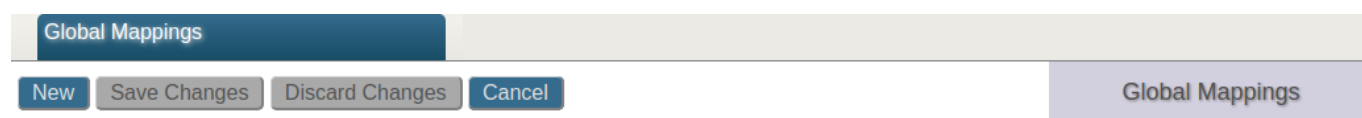
En la base de datos APP-FACT se almacenan las tablas de hechos y los índices. La razón para separar los índices de las tablas de hechos es que los índices pueden ser de gran tamaño. Al definir APP-FACT es posible tener más flexibilidad con la configuración del journal, o definir un tamaño no predeterminado para el bloque. Habilitar journaling para la base de datos APP-FACT es opcional. La elección depende principalmente de que Analytics pueda mantenerse como "no disponible" mientras se reconstruyen los modelos analíticos en caso de que ocurra un evento disruptivo. En este ejemplo, journaling en las tablas de hechos y los índices está deshabilitado, y la razón habitual para efectuar esta elección es que los modelos analíticos son de un tamaño pequeño, se construyen relativamente rápido y se someten a frecuentes reconstrucciones periódicas. Lee la nota en la parte inferior para consultar una discusión más extensa. 

Mapeo de globals

En la siguiente captura de pantalla se muestran los mapeos para el ejemplo de implementación anterior.

Los globals de sincronización ^OBJ.DSTIME y ^DeepSee.Update en DeepSee son mapeados a la base de datos APP-DSTIME. Los globals ^DeepSee.LastQuery y ^DeepSee.QueryLog establecen un registro para todas las consultas MDX que se iniciaron. En el ejemplo, estos globals son mapeados a la base de datos APP-CACHE junto con el caché de DeepSee. Estos mapeos son opcionales.

Los globals ^DeepSee.Fact* y ^DeepSee.Dimension* almacenan las tablas de hechos y dimensiones, mientras que el global ^DeepSee.Index define los índices de DeepSee. Estos globals son mapeados a la base de datos APP-FACT.



The global mappings for namespace APP are displayed below:

Global	Subscript	Database		
DeepSee.Cache*		APP-CACHE	Edit	Delete
DeepSee.Dimension*		APP-FACT	Edit	Delete
DeepSee.Fact*		APP-FACT	Edit	Delete
DeepSee.Index		APP-CACHE	Edit	Delete
DeepSee.JoinIndex*		APP-CACHE	Edit	Delete
DeepSee.LastQuery		APP-CACHE	Edit	Delete
DeepSee.QueryLog		APP-CACHE	Edit	Delete
DeepSee.Update		APP-DSTIME	Edit	Delete
DeepSee.*		APP-DEEPSEE	Edit	Delete
OBJ.DSTIME		APP-DSTIME	Edit	Delete

Comentarios

Como en el ejemplo básico, la información de DeepSee se almacena correctamente en una base de datos dedicada con journaling deshabilitado. La implementación y configuración de DeepSee son mapeadas por separado a una base de datos journaled para poder restablecer la implementación en DeepSee.

Los globals que son compatibles con la sincronización son mapeados a APP-DSTIME y journaled en el Principal (Primary).

Mapear tablas de hechos e índices a una base de datos dedicada permite que la implementación y la configuración de DeepSee se almacenen en una base de datos journaled dedicada (es decir, APP-DEEPSEE),

que puede utilizarse fácilmente para restablecer la implementación en DeepSee.

En el siguiente artículo de esta serie redefiniremos los mapeos para la base de datos APP-FACT y crearemos una base de datos para los índices de DeepSee.

Note sobre journaling y creación de modelos analíticos

Los usuarios deben tener en cuenta que la creación de modelos analíticos elimina y recrea las tablas de hechos e índices de dichos modelos. Esto significa que cuando se habilita journaling, los SETs y KILLS de globals como ^DeepSee.Fact* y ^DeepSee.Index se copian en los archivos journal. Como resultado, la reconstrucción de los modelos analíticos podría llevar a que hubiera una gran cantidad de entradas en los archivos journal y problemas de espacio en el disco.

Por esta razón, se recomienda mapear las tablas de hechos e índices a una o dos bases de datos separadas.

Para las bases de datos Fact e Índices, journaling es opcional y depende de las necesidades que tenga la empresa. Cuando los modelos analíticos tengan un tamaño relativamente pequeño, puedan generarse rápidamente o estén programados para reconstruirse periódicamente, quizás sea preferible deshabilitar journaling.

Habilitar journaling en esta base de datos cuando los modelos analíticos son relativamente grandes ocasionaría que tomara demasiado tiempo reconstruirlos. El caso ideal para mantener journaling es cuando los modelos analíticos se encuentran en un estado estable y solo se sincronizan de forma periódica, pero no se generan. Una manera de generar los modelos analíticos de forma segura es deshabilitar temporalmente el journaling en la base de datos Fact.

[#Analítica](#) [#Bases de datos](#) [#Despliegue](#) [#Mapeo](#) [#Principiante](#) [#Tutorial](#) [#InterSystems IRIS BI \(DeepSee\)](#)

URL de
fuente: <https://es.community.intersystems.com/post/deepsee-bases-de-datos-namespaces-y-mapeos-parte-3-de-5>