

Artigo

[Alessandro Marin](#) · Nov. 26, 2020 3min de leitura

DeepSee - Banco de Dados, Namespaces, e Mapeamento - Parte 1 de 5

Estou planejando implementar a Inteligência de Negócio (BI) com base nos dados de minhas instâncias. Qual é a melhor maneira de configurar meus bancos de dados e ambiente para usar o DeepSee?



Este tutorial aborda essa questão mostrando três exemplos de arquitetura para DeepSee. Começaremos com um modelo de arquitetura básico e destacaremos suas limitações. O modelo subsequente é recomendado para aplicações de Inteligência de Negócio (BI) de complexidade intermediária e deve ser suficiente para a maioria dos casos de uso. Terminaremos este tutorial descrevendo como aumentar a flexibilidade da arquitetura para gerenciar implementações avançadas.

Cada exemplo neste tutorial apresenta novos bancos de dados e mapeamentos globais, junto com uma discussão sobre por que e quando eles devem ser configurados. Ao construir a arquitetura, os benefícios fornecidos pelos exemplos mais flexíveis serão destacados.

Antes de começar

Servidores primários e analíticos

Para tornar os dados altamente disponíveis, a InterSystems geralmente recomenda usar as soluções de espelhamento ou sombreamento e então basear a implementação DeepSee no servidor espelho/sombra. A máquina que hospeda a cópia original dos dados é chamada de servidor Primário, enquanto as máquinas que hospedam cópias dos dados e as aplicações de Inteligência de Negócio (BI) costumam ser chamados de servidores Analíticos (ou, às vezes, de Relatórios)

Ter servidores primários e analíticos é muito importante, o principal motivo é evitar problemas de desempenho em qualquer um dos servidores. Verifique a documentação sobre a [Arquitetura Recomendada](#).

Dados e código da aplicação

Armazenar dados de origem e código no mesmo banco de dados geralmente funciona bem apenas para aplicações de pequena escala. Para aplicações mais extensas, é recomendado armazenar os dados de origem e código em dois bancos de dados dedicados, o que permite compartilhar o código com todos os namespaces onde o DeepSee é executado, mantendo os dados separados. O banco de dados de dados de origem deve ser espelhado no servidor de produção. Este banco de dados pode ser somente leitura ou leitura-gravação. É recomendável manter o registro do diário habilitado para este banco de dados.

As classes de origem e as aplicações personalizadas devem ser armazenados em um banco de dados dedicado nos servidores de produção e analítico. Observe que esses dois bancos de dados para o código-fonte não precisam estar sincronizados ou mesmo rodar a mesma versão do Caché. Normalmente, o registro no diário não é necessário, desde que o backup do código seja feito regularmente em outro lugar.

Neste tutorial teremos a seguinte configuração. O namespace do APP no servidor analítico tem o APP-DATA e o APP-CODE como bancos de dados padrão. O banco de dados APP-DATA tem acesso aos dados (a classe da tabela de origem e seus fatos) no banco de dados de dados de origem no Primário. O banco de dados APP-CODE armazena o código Caché (arquivos .cls e .INT) e outros códigos personalizados. Essa separação de dados e código é uma arquitetura típica e permite ao usuário, por exemplo, implantar com eficiência o código DeepSee e a aplicação personalizada.

Executar DeepSee em diferentes namespaces

As implementações de Inteligência de Negócio (BI) usando DeepSee geralmente são executadas a partir de namespaces diferentes. Nesta postagem, mostraremos como configurar um único namespace de APP, mas o mesmo procedimento se aplica a todos os namespaces onde a aplicação de inteligência de negócio é executada.

Documentação

Recomenda-se familiarizar-se com a página de documentação [Executando a Configuração Inicial](#). Esta página inclui a configuração de aplicações web, como colocar DeepSee globais em bancos de dados separados e uma lista de mapeamentos alternativos para DeepSee globais.

Na [segunda parte](#) desta série mostraremos com a implementação de um modelo básico de arquitetura

[#Analytics](#) [#Bancos de dados](#) [#Implantação](#) [#Iniciante](#) [#Mapeamento](#) [#Tutorial](#) [#InterSystems IRIS BI \(DeepSee\)](#)

URL de
origem: <https://pt.community.intersystems.com/post/deepsee-banco-de-dados-namespaces-e-mapeamento-parte-1-de-5>