

# **Guia do Web Service WsSQL ESP**

**Equipe de documentação de Boca Raton**



## Guia do Web Service WsSQL ESP

Equipe de documentação de Boca Raton

Copyright © 2023 HPCC Systems®. All rights reserved

Sua opinião e comentários sobre este documento são muito bem-vindos e podem ser enviados por e-mail para <docfeedback@hpccsystems.com> Inclua a frase **Feedback sobre documentação** na linha de assunto e indique o nome do documento, o número das páginas e número da Versão atual no corpo da mensagem.

LexisNexis e o logotipo Knowledge Burst são marcas comerciais registradas da Reed Elsevier Properties Inc., usadas sob licença. Os demais produtos, logotipos e serviços podem ser marcas comerciais ou registradas de suas respectivas empresas. Todos os nomes e dados de exemplo usados neste manual são fictícios. Qualquer semelhança com pessoas reais, vivas ou mortas, é mera coincidência.

2023 Version 9.0.2-1

Introdução .....	4
Tipos de Arquivos Suportados .....	5
Preparação .....	6
Configuração .....	6
Usando arquivos HPCC como fonte de dados .....	9
Anotações de Índice .....	9
Métodos .....	11
Echo .....	13
GetDBSystemInfo .....	14
GetDBMetaData .....	15
ExecuteSQL .....	18
GetResults .....	21
PrepareSQL .....	22
ExecutePreparedSQL .....	24
CreateTableAndLoad .....	26
SetRelatedIndexes .....	29
GetRelatedIndexes .....	30
Estruturas Comuns .....	31
Sintaxe SQL suportada .....	32
CALL .....	32
SELECT .....	33
SELECT JOIN .....	36
CREATE / LOAD .....	38
Funções Agregadas Suportadas .....	40

# Introdução

O WsSQL é um serviço que fornece uma interface SQL para o HPCC System. Esse serviço da web aceita um subconjunto preparado e padrão de consultas SQL. Isso fornece acesso à maioria dos dados do HPCC e consultas publicadas por HTTP ou HTTPS usando interfaces SOAP ou REST.

O serviço WsSQL mapeia arquivos lógicos HPCC para tabelas RDBMS. Consultas publicadas do HPCC são expostas como Stored Procedures RDBMS.

O serviço WsSQL também fornece métodos convenientes para obter informações do sistema, metadados e resultados de consultas executadas anteriormente.

Esse serviço destina-se a ser usado de maneira programática (por exemplo, por meio de drivers de banco de dados), mas também pode ser usado de maneira interativa por usuários mais confortáveis usando o SQL do que o ECL. Isso possibilita enviar consultas ad-hoc sem aprender ECL.

O serviço expõe os arquivos lógicos do HPCC como tabelas do RDB.

- Arquivo Lógico HPCC <-> Tabela RDB
- Campos de registro de HPCC <-> Colunas da Tabela RDB
- Consulta publicada do HPCC <-> Store Procedure RDB
- Fornece metadados do sistema HPCC e metadados do RDB de dados
- Suporta subconjunto da sintaxe SQL
- Não-transacional
- Fornece meios para utilizar arquivos de índice HPCC para leituras mais rápidas.

## Tipos de Arquivos Suportados

O serviço WsSQL suporta todos os tipos de arquivos HPCC, exceto:

- XML
- Arquivos com Dataset filho aninhados
- Arquivos sem layout de registro em seus metadados.

O serviço WsSQL **suporta apenas arquivos que contêm a definição de registro nos metadados do arquivo lógico**. Arquivos sprayed não contêm esses metadados. Esses metadados existem em qualquer arquivo ou índice que é gravado no HPCC Distributed File System (DFS). Arquivos de dados sprayed normalmente passam por algum processamento e um OUTPUT dos dados transformados para o disco antes do uso, portanto, isso não deve interferir na utilidade do serviço. Você pode usar o método *CreateTableAndLoad* para produzir um arquivo utilizável a partir de um arquivo sprayed ou em uma zona de entrada de arquivo. Consulte CreateTableAndLoad para detalhes.

# Preparação

O serviço WsSQL está incluído na Plataforma HPCC.

Antes da versão 7.0, o WsSQL era um produto complementar e era instalado separadamente. Se você tiver um sistema configurado com o WsSQL anterior à versão 7.0, deverá desinstalar o WsSQL antes de atualizar a plataforma.

## Configuração

Recomendamos dedicar tempo para a leitura completa deste manual; porém, segue abaixo um guia rápido com o resumo das etapas.

### Configure Utilizando o Assistente

1. Uma vez que o Gerenciador de Configurações estiver em execução:

2. Crie um novo ambiente usando o assistente.

O serviço WsSQL é adicionado automaticamente e sua conexão é criada na configuração do servidor ESP.

3. Salve o arquivo do ambiente.

4. Copie seu arquivo environment.xml para todos os servidores.

```
# for example
sudo -u hpcc cp /etc/HPCCSystems/source/NewEnvironment.xml
/etc/HPCCSystems/environment.xml
```

5. Saia do Gerenciador de Configurações.

6. Reinicie o sistema

7. Acesse a interface do WsSQL no seu navegador (porta 8510).

8. Execute o método Echo para confirmar a conectividade. Consulte [Echo].

### Configurar manualmente

Esse método é útil ao incluir o serviço WsSQL em um sistema existente que não tenha um serviço configurado ou para incluir um serviço WsSql adicional.

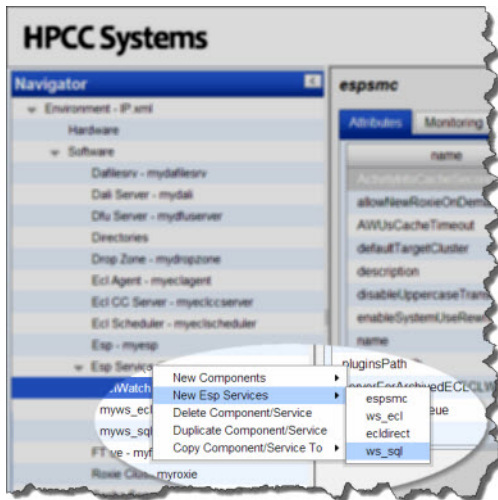
1. Uma vez que o Gerenciador de Configurações estiver em execução:

2. Abra um arquivo de ambiente (\*.xml) no Modo Avançado.

Se você estiver adicionando o WsSQL a um sistema existente, abra um arquivo de ambiente que corresponda ao environment.xml atual. Está em destaque.

3. Marque a caixa de seleção Acesso à gravação.

4. Clique com o botão direito na área **Software** da árvore no painel esquerdo e selecione **Novo Serviço ESP > ws\_sql**.



Isso adiciona uma definição para o serviço.

5. Selecione o componente **ESP**, em seguida, selecione a aba **ESP Service Bindings**.



6. Forneça o nome do serviço para conectar e a porta. (A porta padrão é 8510)
7. Salve o arquivo do ambiente.
8. Copie seu arquivo environment.xml para todos os servidores

```
# for example
sudo -u hpcc cp /etc/HPCCSystems/source/NewEnvironment.xml
/etc/HPCCSystems/environment.xml
```

9. Saia do Gerenciador de Configurações.

10 Reinicie o sistema.

- 11 Acesse a interface do WsSQL no seu navegador:

Utilizando *http://nnn.nnn.nnn.nnn:pppp* Usando *http://nnn.nnn.nnn.nnn:pppp* (onde nnn.nnn.nnn.nnn é o endereço IP do seu servidor ESP e pppp é a porta de serviço WsSQL. A porta padrão é 8510)

12Execute o método Echo para confirmar a conectividade. Consulte [Echo].



# Usando arquivos HPCC como fonte de dados

Uma vez instalado e configurado, o serviço WsSQL processará as instruções SQL enviadas e gerará o código ECL. O código é submetido e executado pela sua Plataforma HPCC. O conjunto de resultados é retornado para sua aplicação.

**Observação:** O serviço WsSQL **suporta apenas arquivos que contêm a definição de registro nos metadados do arquivo lógico**. Arquivos com spray não contêm esses metadados. Esses metadados existem em qualquer arquivo ou índice que é gravado no HPCC Distributed File System (DFS). Arquivos de dados sprayed normalmente passam por algum processamento e um OUTPUT dos dados transformados para o disco antes do uso, portanto, isso não deve interferir na utilidade do serviço. Você pode usar o método *CreateTableAndLoad* para produzir um arquivo utilizável a partir de um arquivo pulverizado ou em uma zona de entrada de arquivo. Consulte *CreateTableAndLoad* para detalhes.

Além disso, você pode utilizar índices no HPCC de duas maneiras:

1. Forneça instruções SQL para que o serviço use um índice específico para sua consulta.

Por exemplo:

```
USEINDEX(TutorialPersonByZipIndex)
```

2. Especifique os índices relacionados na descrição do arquivo lógico do HPCC.

## Anotações de Índice

O serviço WsSQL tenta executar leituras baseadas em índice sempre que possível. No entanto, para aproveitar as leituras de índice, os arquivos de HPCC de destino precisam ser anotados com os nomes de arquivo de índice pertinentes. Isso é feito adicionando a seguinte entrada de chave/valor na descrição do arquivo usando o ECL Watch.

Na página de detalhes de um arquivo lógico, insira as informações na caixa de entrada Description e pressione o botão **Save**.

Essas informações são usadas pelo serviço para decidir se uma busca de índice é possível para uma consulta no arquivo base.

## No arquivo de origem:

**XDBC:RelIndexes= [fullLogicalFilename1; fullLogicalFilename2]**

Exemplo:

```
XDBC:RelIndexes=[tutorial::yn::peoplebyzipindex;  
                 tutorial::yn::peoplebyzipindex2;  
                 tutorial::yn::peoplebyzipindex3]
```

Neste exemplo, o arquivo de origem possui três índices disponíveis.

Você pode adicionar anotações usando o método *SetRelatedIndexes*.

## No arquivo de índice:

**XDBC:PosField**=[*indexPositionFieldName*]

Exemplo:

```
XDBC:PosField=[fpos]
```

O campo FilePosition (fpos) pode ter qualquer nome, portanto, ele deve ser especificado nos metadados para que o serviço saiba qual campo é o fileposition.

Basta digitar as informações na caixa de entrada da descrição e pressionar o botão **Save Description** .

**Observação:** Você deve inserir essas informações ANTES de publicar qualquer consulta usando o arquivo de dados ou índices. Consultas publicadas bloqueiam o arquivo e impedem a edição dos metadados.

# Métodos

Esses métodos estão incluídos na interface de serviço do WsSQL:

- **Echo**

Este método é fornecido para testar a comunicação de ponta a ponta. A sequência de entrada é ecoada na resposta.

- **GetDBSystemInfo**

Você pode usar esse método para coletar informações de versão da Plataforma HPCC.

- **GetMetadata**

Esse método fornece metadados que você pode usar para criar uma exibição ou um modelo do sistema de HPCC de destino como um banco de dados acessível SQL. Você pode solicitar tabelas, Stored Procedures, e/ou Targetclusters.

- **ExecuteSQL**

Use este método para enviar consultas SQL padrão (não preparadas). Este método aceita texto-livre SQL (veja a sintaxe suportada abaixo).

- **GetRelatedIndexes**

Este método recupera informações da descrição de um arquivo lógico sobre índices relacionados.

- **GetResults**

Esse método fornece resultados de consultas executadas anteriormente.

- **PrepareSQL**

Esse método fornece a capacidade de enviar uma solicitação SQL free-hand para uso posterior como uma consulta parametrizada. Ele compila a consulta e retorna a ID da Workunit (WUID). Este WUID é usado posteriormente para executar a consulta com os parâmetros de entrada fornecidos usando o método ExecutePreparedSQL. Isso é ideal para consultas que são executadas muitas vezes com valores diferentes.

- **ExecutePreparedSQL**

Este método executa uma consulta SQL parametrizada previamente criada. A consulta é referenciada usando um ID de Workunit (WUID), que é retornado do método PrepareSQL. O chamador pode especificar a sequência de parâmetros de entrada como pares de valor-chave, que estão vinculados à consulta pré-compilada.

- **CreateTableAndLoad**

Esse método lê um arquivo lógico ou um arquivo em uma zona de entrada e cria uma nova tabela e carrega os dados do arquivo de origem criando uma tabela (arquivo lógico) que o WsSQL pode usar. Esta função destina-se ao uso quando você tem um arquivo lógico sem metadados ou para importar um novo arquivo para uma plataforma HPCC.

- **SetRelatedIndexes**

Esse método adiciona informações a uma descrição de arquivo lógico que o WsSQL usa como uma anotação de um índice relacionado.

## Echo

Essa função recebe uma string de entrada e "echoes" o valor em seu resultado.

Esta função destina-se ao teste de conectividade de ponta a ponta. Uma resposta bem-sucedida indica uma boa conexão com o servidor que hospeda o serviço da Web do Ws-SQL. Esta função foi construída para o teste de conectividade.

Recomendamos usar essa função como uma primeira etapa do processo de desenvolvimento de aplicativos.

## Exemplo de entrada XML

```
<soap:Envelope>  
  <soap:Body>  
    <EchoRequest>  
      <Request>StringToEcho</Request>  
    </EchoRequest>  
  </soap:Body>  
</soap:Envelope>
```

## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
EchoRequest	N	Estrutura contendo a solicitação
Request	N	String para ecoar no resultado

## Descrições da Tag de Resposta

Nome da Tag	Descrição
EchoResponse	Estrutura contendo a resposta
Response	Response

# GetDBSystemInfo

Este método permite obter informações sobre a versão do HPCC System.

## Exemplo de entrada XML

```
<soap:Envelope>  
  <soap:Body>  
    <GetDBSystemInfoRequest>  
      <IncludeAll>1</IncludeAll>  
    </GetDBSystemInfoRequest>  
  </soap:Body>  
</soap:Envelope>
```

## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
GetDBSystemInfoRequest	Y	Estrutura contendo a solicitação
IncludeAll	N	Se definido como 1 ou verdadeiro, todas as informações disponíveis serão retornadas

## Descrições da Tag de Resposta

Nome da Tag	Descrição
GetDBSystemInfoResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
Name	Nome
FullVersion	Versão Completa da Plataforma HPCC
Major	Major da Plataforma HPCC
Minor	Minor da Plataforma HPCC
Point	Ponto da Plataforma HPCC
Project	Project da Plataforma HPCC
Maturity	Maturidade da Plataforma HPCC
WsSQLFullVersion	Versão Completa do WsSQL
WsSQLMajor	Major do WsSQL
WsSQLMinor	Minor do WsSQL
WsSQLPoint	Ponto do WsSQL
WsSQLProject	Projeto do WsSQL
WsSQLMaturity	Maturidade do WsSQL

## GetDBMetaData

Esse método permite que você consulte a Plataforma HPCC e obtenha metadados para criar uma exibição ou modelo do sistema de HPCC de destino como um banco de dados acessível por SQL.

Você pode solicitar um ou mais dos seguintes itens:

- Tabelas (arquivos lógicos no cluster HPCC)
- Stored Procedures (Consultas Publicadas)
- TargetClusters

### Exemplo de entrada XML

```
<soap:Envelope>
  <soap:Body>
    <GetDBMetaDataRequest>
      <IncludeTables>1</IncludeTables>
      <TableFilter/>
      <IncludeStoredProcedures>1</IncludeStoredProcedures>
      <QuerySet/>
      <IncludeTargetClusters>1</IncludeTargetClusters>
      <ClusterType/>
    </GetDBMetaDataRequest>
  </soap:Body>
</soap:Envelope>
```

### Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
GetDBMetaDataRequest	Y	Estrutura contendo a solicitação
IncludeTables	N	Se definido como 1 ou verdadeiro, as tabelas disponíveis serão incluídas na resposta
TableFilter	N	Filtrar por resultados da tabela
IncludeStoredProcedures	N	Se definido como 1 ou verdadeiro, as Stored Procedures serão incluídas na resposta
QuerySet	N	QuerySet para usar como filtro para procedimentos armazenados para retornar
IncludeTargetClusters	N	Se definido como 1 ou verdadeiro, os Clusters de Destino disponíveis serão incluídos na resposta
ClusterType	N	Tipo de cluster a ser usado como filtro

## Descrições da Tag de Resposta

Nome da Tag	Descrição
GetDBMetaDataResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
TableCount	Contagem de tabelas disponíveis
Tables	Estrutura contendo uma ou mais tabelas
Table	Estrutura contendo uma tabela
Name	Nome da tabela
Columns	Estrutura contendo uma ou mais colunas
Columns	Colunas
Name	Nome da Coluna
Type	Tipo de dados da coluna (por exemplo, unsigned8, string3)
ECL	Definição ECL para a tabela
Format	Formato da tabela (por exemplo, FLAT, KEYED, etc)
ContentType	Tipo de Conteúdo
Description	Descrição
IsKeyed	Indicador booleano: É um índice?
IsSuper	Indicador booleano: É um superfile?
CsvQuote	Caractere de cotação (válido apenas para arquivos CSV)
CsvSeparate	Caractere de separação (válido apenas para arquivos CSV)
CsvTerminate	Gravar caractere terminador (válido apenas para arquivos CSV)
Group	Grupo
MaxRecordSize	Tamanho máximo do registro
Modified	Data de modificação
NumParts	Número da partes dos arquivos
Owner	Nome do Owner
QuerySets	Estrutura contendo um ou mais conjuntos de consultas
QuerySet	Estrutura contendo um Grupo de Consultas
Name	Nome da Consulta
QuerySetQueries	Estrutura contendo uma ou mais QuerySetQueries
QuerySetQuery	Estrutura contendo um QuerySetQuery
Name	Nome da Consulta
Id	ID da consulta (somente para uso interno e fins informativos)
Wuid	ID da workunit
Suspended	Indicador booleano: A consulta está suspensa?
Signature	Assinatura da Consulta
InParams	Estrutura contendo uma ou mais parâmetros de entrada



Guia do Web Service WsSQL ESP  
Métodos

InParam	Estrutura contendo um parâmetro de entrada
Name	Nome do parâmetro
Type	Tipo de dados de parâmetro (por exemplo, string)
ResultSets	Estrutura contendo um ou mais Conjuntos de Resultados (somente o primeiro conjunto de resultados é retornado de uma solicitação de serviço WsSQL)
ResultSet	Estrutura contendo um conjunto de resultados
Name	Nome do conjunto de resultados
OutParams	Estrutura contendo uma ou mais parâmetros de saída
OutParam	Estrutura contendo um parâmetro de resultado
Name	Nome do parâmetro
Type	Tipo de dados de parâmetro (por exemplo, string)
QuerySetAliases	Estrutura contendo uma ou mais QuerySetAliases
QuerySetAlias	Estrutura contendo um QuerySetAlias
Name	Nome da Consulta
Id	ID da Consulta
ClusterNames	Estrutura contendo uma ou mais TargetClusters
ClusterName	Estrutura contendo um TargetCluster

## ExecuteSQL

Use este método para enviar consultas SQL padrão (não preparadas). A tag de entrada **SqlText** aceita texto em SQL livre (consulte a sintaxe SQL suportada abaixo).

Se você está executando SQL usando SELECT ou SELECT JOIN, você deve especificar o **TargetCluster**.

Se você estiver executando o SQL que usa um CALL em uma stored procedure, deverá qualificar totalmente o nome da procedure (por exemplo: Roxie.FindPeopleByZip) ou especifique o **TargetQuerySet** aqui. Os parâmetros devem ser passados em ordem, não por nome. Você pode recuperar o pedido usando GetDBMetaData.

Para excluir o esquema de resultados no resultado, defina a opção **SuppressXmlSchema** opção para 1 ou verdadeiro.

Para a paginação do conjunto de resultados, você pode limitar o total de resultados da consulta e a página inicial retornada (**ResultWindowStart**, **ResultWindowCount**).

## Sample Input XML

```
<soap:Envelope>
  <ExecuteSQLRequest>
    <SqlText>SELECT * from tutorial::yn::tutorialperson where lastname='JONES'</SqlText>
    <UserName>Emily</UserName>
    <!-- Use either TargetCluster or TargetQuerySet, not both -->
    <!-- If stored procedure was fully qualified, you can omit TargetQuerySet -->
    <TargetCluster>thor</TargetCluster>
    <TargetQuerySet></TargetQuerySet>
    <SuppressResults>1</SuppressResults>
    <SuppressXmlSchema>0</SuppressXmlSchema>
    <resultLimit>100</resultLimit>
    <!-- For page loading -->
    <ResultWindowStart>0</ResultWindowStart>
    <ResultWindowCount>50</ResultWindowCount>
  </ExecuteSQLRequest>
</soap:Envelope>
```

## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
ExecuteSQLRequest	Y	Estrutura contendo a solicitação
SqlText	Y	Este método aceita texto-livre SQL (veja a sintaxe suportada abaixo).
UserName	N	Nome de usuário para transmitir para o HPCC. Isso é usado como job owner no HPCC
TargetCluster	Y *	Se você estiver executando o SQL usando um SELECT ou SELECT JOIN, deverá especificar o TargetCluster.
TargetQuerySet	Y *	Se você estiver executando um SQL preparado que use um CALL em um procedimento armazenado, deverá qualificar totalmente o nome do procedimento no SQL preparado (por exemplo: Roxie.FindPeopleByZip) ou especifique o TargetCluster aqui.
SuppressResults	N	Se definido como 1 ou verdadeiro, os resultados da consulta não serão incluídos na resposta
SuppressXmlSchema	N	Se definido como 1 ou verdadeiro, o esquema do resultado da consulta não será incluído na resposta
Wait	N	Tempo limite em milissegundos. Use -1 para tempo ilimitado
resultLimit	N	Se definido, os resultados podem conter tantos registros quanto esse limite
ResultWindowStart	N	Para uso com carregamento de página, o registro inicial a ser retornado
ResultWindowCount	N	Para uso com carregamento de página, o número de registros a serem incluídos no ResultWindowStart

\* Um ou outro é obrigatório

## Descrições da Tag de Resposta

Nome da Tag	Descrição
ExecuteSQLResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
ParentWuId	O ID da Workunit para a workunit pai (aquela com o SQL preparado)
Result	O resultado da consulta
Workunit	Estrutura contendo detalhes da Workunit. Consulte Estrutura da Workunit em Estruturas Comuns
resultLimit	O resultLimit enviado
ResultWindowStart	O ResultWindowStart enviado

Guia do Web Service WsSQL ESP  
Métodos

ResultWindowCount	O ResultWindowCount enviado
-------------------	-----------------------------

## GetResults

Esse método fornece resultados de consultas executadas anteriormente. Use a **Wuid** retornada de **ExecuteSQL** ou **PrepareSQL**. Este método é ideal para paginação de resultados.

### Sample Input XML

```
<soap:Envelope>
  <soap:Body>
    <GetResultsRequest>
      <WuId>W20140529-161930</WuId>
      <SuppressXmlSchema>1</SuppressXmlSchema>
      <ResultWindowStart>0</ResultWindowStart>
      <ResultWindowCount>0</ResultWindowCount>
    </GetResultsRequest>
  </soap:Body>
</soap:Envelope>
```

### Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
GetResultsRequest	Y	Estrutura contendo a solicitação
WuId	Y	Workunit Id (WUID)
SuppressXmlSchema	N	Se definido como 1 ou verdadeiro, o esquema do resultado da consulta não será incluído na resposta
ResultWindowStart	N	Para uso com carregamento de página, o registro inicial a ser retornado
ResultWindowCount	N	Para uso com carregamento de página, o número de registros a serem incluídos no ResultWindowStart

### Descrições da Tag de Resposta

Nome da Tag	Descrição
GetResultsResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
Result	O resultado da consulta
ResultWindowStart	O registro inicial enviado
ResultWindowCount	A contagem de registros enviados
Workunit	Estrutura contendo detalhes da Workunit. Consulte Estrutura da Workunit em Estruturas Comuns

## PrepareSQL

Use este método para enviar uma solicitação free-hand SQL para uso posterior como uma consulta parametrizada. Isso compila a consulta e retorna a **Wuid**. Esta **Wuid** é usada posteriormente para executar a consulta com os parâmetros de entrada fornecidos usando o método **ExecutePreparedSQL**.

Isso é ideal para consultas que são executadas muitas vezes com valores diferentes.

O SQL preparado pode conter qualquer sintaxe SQL suportada.

Se você está executando SQL usando SELECT ou SELECT JOIN, você deve especificar o **TargetCluster**.

Se você estiver usando uma CHAMADA para uma store procedure armazenada, deverá qualificar totalmente o nome do procedimento no SQL (por exemplo: Roxie.FindPeopleByZip) ou especifique o **TargetQuerySet** aqui. Os parâmetros devem ser passados em ordem, não por nome. Você pode recuperar o pedido usando GetDBMetaData.

Crie parâmetros usando um ? como um placeholder.

### Exemplo:

```
select * from tutorial::yn::tutorialperson where lastname=?
```

Mais tarde, você enviaria uma solicitação para **ExecutePreparedSQL** fornecer um valor para usar o *lastname* conforme mostrado no exemplo a seguir:

```
<soap:Envelope >
  <soap:Body>
    <ExecutePreparedSQLRequest>
      <WuId>w20140724-135811</WuId>
      <TargetCluster>thor</TargetCluster>
      <SuppressResults>0</SuppressResults>
      <SuppressXmlSchema>1</SuppressXmlSchema>
      <resultLimit>100</resultLimit>
      <Variables>
        <NamedValue>
          <Name>lastname</Name>
          <Value>JONES</Value>
        </NamedValue>
      </Variables>
    </ExecutePreparedSQLRequest>
  </soap:Body>
</soap:Envelope>
```

## Sample Input XML

```
<soap:Envelope >
  <soap:Body>
    <PrepareSQLRequest>
      <SqlText>select * from tutorial::yn::tutorialperson where lastname=?</SqlText>
      <!-- Use either TargetCluster or TargetQuerySet, not both -->
      <!-- If stored procedure is fully qualified, you can omit TargetQuerySet -->
      <TargetCluster>thor</TargetCluster>
    </PrepareSQLRequest>
  </soap:Body>
</soap:Envelope>
```

## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
PrepareSQLRequest	N	Estrutura contendo a solicitação
SqlText	Y	Este método aceita texto-livre SQL (veja a sintaxe suportada abaixo).
TargetCluster	Y *	Se você estiver executando o SQL usando um SELECT ou SELECT JOIN, deverá especificar o TargetCluster.
TargetQuerySet	Y *	Se você estiver executando um SQL preparado que use um CALL em um procedimento armazenado, deverá qualificar totalmente o nome do procedimento no SQL preparado (por exemplo: Roxie.FindPeopleByZip) ou especifique o TargetCluster aqui.
Wait	N	Tempo limite em milissegundos. Use -1 para tempo ilimitado

\* Um ou outro é obrigatório

## Descrições da Tag de Resposta

Nome da Tag	Descrição
PrepareSQLResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
Workunit	Estrutura contendo detalhes da Workunit. Consulte Estrutura da Workunit em Estruturas Comuns
Result	Estrutura contendo resultado (se houver)

## ExecutePreparedSQL

Este método executa uma consulta SQL parametrizada previamente criada.

A consulta compilada de destino é referenciada usando um ID da Workunit (**Wuid**), que é devolvido pelo **PrepareSQL**. O chamador pode especificar a sequência de parâmetros de entrada como pares de valor-chave, que estão vinculados à consulta pré-compilada.

O SQL preparado pode conter qualquer sintaxe SQL suportada.

Se você estiver executando o SQL usando um SELECT ou SELECT JOIN, pode fornecer um **TargetCluster** para substituir o especificado ao enviar a solicitação PrepareSQL; no entanto, deve ser um cluster do mesmo tipo.

Se você quiser limitar o número de resultados, você deve usar uma cláusula LIMIT em sua consulta SQL.

Para a paginação do conjunto de resultados, você pode limitar o total de resultados da consulta e a página inicial retornada (**ResultWindowStart**, **ResultWindowCount**).

## Sample Input XML

```
<soap:Envelope>
  <soap:Body>
    <ExecutePreparedSQLRequest>
      <Wuid>W20140529-161930</Wuid>
      <UserName>EmilyKate</UserName>
      <!-- You can override the TargetCluster used in original PrepareSQL query, -->
      <!-- but it must be of the same type -->
      <TargetCluster>Thor</TargetCluster>
      <SuppressResults>0</SuppressResults>
      <SuppressXmlSchema>1</SuppressXmlSchema>
      <Wait>-1</Wait>
      <!-- For page loading -->
      <ResultWindowStart>0</ResultWindowStart>
      <ResultWindowCount>50</ResultWindowCount>
      <!-- Paramaters using name/value pairs -->
      <Variables>
        <NamedValue>
          <Name>firstname</Name>
          <Value>Jim</Value>
        </NamedValue>
        <NamedValue>
          <Name>lastname</Name>
          <Value>JONES</Value>
        </NamedValue>
      </Variables>
    </ExecutePreparedSQLRequest>
  </soap:Body>
</soap:Envelope>
```



## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
ExecutePreparedSQLRequest	N	Estrutura contendo a solicitação
WuId	Y	A ID da Workunit (WUID)
UserName	N	O nome de usuário para usar como nome próprio do trabalho na plataforma HPCC
TargetCluster	Y	Se você estiver executando o SQL preparado usando um SELECT ou SELECT JOIN, poderá especificar o TargetCluster, mas ele deve ser do mesmo tipo daquele em que foi preparado
SuppressResults	N	Se definido como 1 ou verdadeiro, os resultados da consulta não serão incluídos na resposta
SuppressXmlSchema	N	Se definido como 1 ou verdadeiro, o esquema do resultado da consulta não será incluído na resposta
Wait	N	Tempo limite em milissegundos. Use -1 para tempo ilimitado
ResultWindowStart	N	Para uso com carregamento de página, o registro inicial a ser retornado
ResultWindowCount	N	Para uso com carregamento de página, o número de registros a serem incluídos no ResultWindowStart
Variables	N	Se seu SQL preparado tiver parâmetros, forneça-os como pares nome/valor dentro dessa estrutura
NamedValue	N	Uma estrutura contendo um par Nome/Valor
Name	N	Name
Value	N	Valor

## Descrições da Tag de Resposta

Nome da Tag	Descrição
ExecutePreparedSQLResponse	Estrutura contendo a resposta
Exceptions	Estrutura contendo o pedido. Consulte Estrutura de Exceções em Estruturas Comuns
ParentWuId	O ID da Workunit para a workunit pai (aquela com o SQL preparado)
Result	O resultado da consulta
ResultWindowStart	O registro inicial enviado
ResultWindowCount	A contagem de registros enviados
Workunit	Estrutura contendo detalhes da Workunit. Consulte Estrutura da Workunit em Estruturas Comuns

## CreateTableAndLoad

Esse método cria uma tabela (arquivo lógico do HPCC) que pode ser acessada subsequentemente usando o WsSQL. A tabela recebe o layout conforme especificado na seção ECLFields da solicitação. A tabela é então preenchida com dados de um arquivo existente. O arquivo de dados de origem pode residir na Zona de Entrada de Arquivo do HPCC após ser carregado, ou pode já ter sofrido o spray no sistema.

### Sample Input XML

```
<CreateTableAndLoadRequest>
  <TableName>JIM:MyNewTable</TableName>
  <TableDescription>My Description of the new table</TableDescription>
  <Overwrite>1</Overwrite>
  <EclFields>
    <EclField>
      <FieldName>PersonName</FieldName>
      <EclFieldType>
        <!-- Valid types are: BOOLEAN, INTEGER, UNSIGNED, REAL, DECIMAL, STRING, -->
        <!-- QSTRING, UNICODE, DATA, VARSTRING, or VARUNICODE -->
        <Type>STRING</Type>
        <Locale></Locale>
        <Length>20</Length>
        <Precision></Precision>
      </EclFieldType>
    </EclField>
    <EclField>
      <FieldName>PersonID</FieldName>
      <EclFieldType>
        <Type>INTEGER</Type>
        <Locale></Locale>
        <Length>2</Length>
        <Precision></Precision>
      </EclFieldType>
    </EclField>
  </EclFields>
  <TargetCluster>thor</TargetCluster>
  <Owner>Jimmy</Owner>
  <DataSource>
    <!-- Use either SprayedFileName -->
    <SprayedFileName>Tutorial::JD:OriginalPerson</SprayedFileName>
    <!-- or full Landing Zone details, not both -->
    <LandingZoneIP>127.0.0.1</LandingZoneIP>
    <LandingZonePath>/var/lib/HPCCSystems/mydropzone</LandingZonePath>
    <LandingZoneFileName>OriginalPerson</LandingZoneFileName>
  </DataSource>
  <DataSourceType>
    <!-- Valid types are: FLAT, CSV, JSON, or XML -->
    <Type>FLAT</Type>
    <Params>
      <Param>
        <Name></Name>
        <Values>
          <Value></Value>
          <Value></Value>
        </Values>
      </Param>
    </Params>
  </DataSourceType>
  <Wait>-1</Wait>
</CreateTableAndLoadRequest>
```

## Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
TableName	Y	O nome da tabela a ser criada no sistema de arquivos distribuídos da Plataforma HPCC.
TableDescription	N	Sua descrição do arquivo.
Overwrite	N	Indicador booleano para especificar se deve permitir que o novo arquivo sobrescreva um arquivo existente com o mesmo nome.
EclFields	Y	Estrutura contendo detalhes para o layout do registro do arquivo. Isso deve corresponder ao layout do arquivo de dados de origem
EclField	Y	Estrutura contendo detalhes para um campo
FieldName	Y	Nome do Campo
EclFieldType	Y	Estrutura contendo detalhes do tipo de dados
Type	Y	Field type. Os tipos válidos são: BOOLEAN, INTEGER, UNSIGNED, REAL, DECIMAL, STRING, QSTRING, UNICODE, DATA, VARSTRING, ou VARUNICODE
Locale	N	Locale
Length	Y*	Comprimento do campo
Precision	N	Precisão de campo (casas decimais)
TargetCluster	Y	O cluster de destino em que o job será executado.
Owner	N	Nome do owner para o arquivo
DataSource	Y	Estrutura contendo detalhes do Data Source Os detalhes de SprayedFileName ou LandingZone File devem ser fornecidos, não ambos.
SprayedFileName	Y**	Nome de arquivo lógico do arquivo de dados pulverizado
LandingZoneIP	Y**	Endereço IP ou nome de host da Zona de Entrada de Arquivo do HPCC
LandingZonePath	Y**	Caminho para o arquivo de dados na Zona de Entrada de Arquivo
LandingZoneFileName	Y**	Nome do arquivo físico
DataSourceType	N	Estrutura contendo detalhes do tipo DataSource.
Type	Y	Os tipos válidos são: FLAT, CSV, JSON, or XML
Params	N	Estrutura contendo um ou mais parâmetros de tipo que descrevem o DataSource. Por exemplo, CSV Terminator = \n
Param	N	Estrutura contendo um tipo de parâmetro.
Name	N	Nome
Values	N	Estrutura contendo um ou mais valores.
Value	N	Valor

Wait	N	Tempo limite em milissegundos. Use -1 para tempo ilimitado
------	---	--

\* O comprimento do campo é necessário apenas para campos de largura fixa (arquivos FLAT)

\*\* Os detalhes do SprayedFileName ou do LandingZone File devem ser fornecidos, não ambos

## Descrições da Tag de Resposta

A estrutura de resposta contém muitas tags que fornecem informações úteis para desenvolvimento e depuração do código ECL. A maioria deles está fora do escopo deste documento. Apenas aqueles que são úteis ao usar o serviço WsSQL estão listados abaixo.

Nome da Tag	Descrição
CreateTableAndLoadResponse	Estrutura contendo a resposta
TableName	Nome da tabela criada
Success	Indicador booleano de sucesso [1 = sucesso]
EclRecordDefinition	Definição de registro fornecida no formato ECL
Workunit	Estrutura contendo detalhes da Workunit. Consulte Estrutura da Workunit em Estruturas Comuns

## SetRelatedIndexes

Essa função adiciona uma descrição a um arquivo lógico a ser usado como uma anotação que indica um arquivo de índice relacionado a um arquivo de dados. Isso o torna disponível para o WsSQL para uso em uma busca indexada.

### Sample Input XML

```
<SetRelatedIndexesRequest>
  <RelatedIndexSets>
    <RelatedIndexSet>
      <FileName>Tutorial::JD::TutorialPerson</FileName>
      <Indexes>
        <Index>Tutorial::JD::TutorialPersonByName</Index>
        <Index>Tutorial::JD::TutorialPersonByZIP</Index>
      </Indexes>
    </RelatedIndexSet>
  </RelatedIndexSets>
</SetRelatedIndexesRequest>
```

### Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
SetRelatedIndexesRequest		Estrutura contendo a solicitação
RelatedIndexSets	Y	Estrutura contendo uma ou mais relacionadas índice estabelece
RelatedIndexSet	Y	Estrutura contendo um conjunto de índice relacionados
FileName	Y	Nome de arquivo lógico ao qual a anotação é adicionada
Indexes	Y	Estrutura contendo um ou mais índices para adicionar à anotação
Index	Y	Índice para adicionar à anotação

### Descrições da Tag de Resposta

Nome da Tag	Descrição
SetRelatedIndexesResponse	Estrutura contendo a resposta
RelatedIndexSets	Estrutura contendo uma ou mais relacionadas índice estabelece
RelatedIndexSet	Estrutura contendo um conjunto de índice relacionados
FileName	Nome de arquivo lógico ao qual a anotação foi adicionada
Indexes	Estrutura contendo um ou mais índices adicionados à anotação
Index	Índice adicionados à anotação

## GetRelatedIndexes

Essa função recupera informações de descrições de arquivos lógicos sobre anotações que indicam um arquivo de índice relacionado a um arquivo de dados.

### Sample Input XML

```
<GetRelatedIndexesRequest>
  <FileNames>
    <FileName>Tutorial::JD::TutorialPerson</FileName>
  </FileNames>
</GetRelatedIndexesRequest>
```

### Descrições das Tags de Solicitação

Nome da Tag	Req?	Descrição
SetRelatedIndexesRequest		Estrutura contendo a solicitação
FileNames	Y	Estrutura contendo uma ou mais relacionadas índice estabelece
FileName	Y	Nome de arquivo lógico ao qual a anotação é adicionada

### Descrições da Tag de Resposta

Nome da Tag	Descrição
GetRelatedIndexesResponse	Estrutura contendo a resposta
RelatedIndexSets	Estrutura contendo uma ou mais relacionadas índice estabelece
RelatedIndexSet	Estrutura contendo um conjunto de índice relacionados
FileName	Nome de arquivo lógico ao qual a anotação foi adicionada
Indexes	Estrutura contendo um ou mais índices adicionados à anotação
Index	Índice adicionados à anotação

## Estruturas Comuns

Essas estruturas são retornadas em vários métodos.

### Estrutura de Exceções

Nome da Tag	Descrição
Exceptions	Estrutura contendo uma ou mais exceções
Exceção	Estrutura contendo uma exceção
Code	Code
Audience	Audience
Source	Componente de origem
Mensagem	Mensagem de Erro

### Estrutura da Workunit

A estrutura da workunit contém muitas tags que fornecem informações úteis para desenvolvimento e depuração do código ECL. A maioria deles está fora do escopo deste documento. Somente aqueles que são úteis ao usar o serviço WsSQL são listados aqui.

Nome da Tag	Descrição
Workunit	Estrutura contendo detalhes da Workunit.
Wuid	Workunit Id (WUID)
Owner	Owner do Job (se houver)
Cluster	Cluster de Destino
Jobname	Nome do Job (se houver)
StateID	State ID
State	Estado (compilado, falho, etc)
Protected	Indicador booleano: A Workunit está protegida?
Snapshots	Snapshots
IsPausing	Indicador booleano de estado de pausa
ThorLCR	Indicador booleano
EventSchedule	Indicador booleano
TotalClusterTime	Tempo gasto para processar no cluster
Query	Query
Result	Estrutura contendo resultado (se houver)

# Sintaxe SQL suportada

## CALL

**Call** *queryName* ([*paramList*])

queryName	O nome da consulta publicada ou o alias
paramList	Os parâmetros expostos pela consulta publicada (separados por vírgulas)

Call executa uma consulta ECL publicada como se fosse uma stored procedure.

Exemplo:

```
Call SearchPeopleByZipService ('33024')
```



# SELECT

**select** [distinct] *columnList* **de** *tableList* [USE INDEX(*indexFileName* | NONE )]

[**where** *logicalExpression*] [**group by** *columnList*<sup>1</sup>] [**having** *logicalExpression*<sup>2</sup>]

[**order by** *columnList*<sup>1</sup> [asc | desc] ] [**LIMIT** *limitNumber*]

**OBSERVAÇÃO:** Os identificadores podem estar sem aspas ou entre aspas duplas; os valores das sequências literais devem ter aspas simples.

<i>columnList</i>	columnreference1[,columnreference2,columnreference3,...,columnreference <sub>n</sub> ]
	A(s) coluna(s) a retornar(em) (lista separada por vírgula). Além disso, essas funções agregadas são suportadas: COUNT, SUM, MIN, MAX, and AVG. Estes funcionam de uma maneira similar às suas contrapartes ECL
columnreference	[tablename.]columnname[ [AS] alias ]
<i>distinct</i>	[distinct] col1, col2,... col <sub>n</sub>
	O conjunto de resultados conterá apenas valores distintos (exclusivos)
<i>tableList</i>	tableref1[,tableref2,tableref3,...,tableref <sub>n</sub> ]
	Uma ou mais tabelas, separadas por vírgulas.
	<b>OBSERVAÇÃO:</b> Uma lista de tabelas com múltiplas tabelas criadas com (uma ou mais) inner join implícito usando a expressão lógica da cláusula where como a condição de join que deve conter uma condição de igualdade
tableref	tableName[ [AS] alias ]
	O nome da tabela como referenciado, definindo opcionalmente seu alias
<i>alias</i>	O alias usado para se referir à tabela ou referência de campo correspondente.
<i>logicalExpression</i>	Expressão lógica baseada na sintaxe padrão de filtragem SQL.
	BOOLEAN suporta apenas <i>Verdade</i> ou <i>Falso</i> , não use Y, N, 0 ou 1
	<b>Operadores válido:</b>
	= Equal (e.g., age=33)
	<> Not equal (e.g., age <>33)
	> Greater than (e.g., age >55)
	< Less than (e.g., age < 18)
	>= Greater than or equal (e.g., age >=21)
	<= Less than or equal (e.g., age <=21)
	IN(value1,value2,...,value <sub>n</sub> ) where values are comma separated homogeneous types
	NOT IN(value1,value2,...,value <sub>n</sub> ) where values are comma separated homogeneous types
	LIKE <i>pattern</i> where the pattern uses SQL LIKE operators with % and _ wildcards.
	NOT LIKE <i>pattern</i> where the pattern uses SQL LIKE operators with % and _ wildcards.

<i>limitNumber</i>	O número de linhas para retornar. Isso substitui o atributo de configuração padrão (EclResultLimit), mas não pode ser definido como ALL
--------------------	---

<sup>1</sup>Aliasing não suportado

<sup>2</sup>Pode conter apenas referências a funções agregadasse usado com a cláusula *HAVING*.

Funções agregadas só podem ser expressas em logicalExpressions usando *Group by* *HAVING*

Exemplos:

```
Select * from tableList where Sum(F1 > 100) /* is NOT SUPPORTED */
Select * from tableList Group by F1 Having Sum (F1 > 100) /* IS SUPPORTED */
```

Exemplo:

```
Select fname, lname, state from TutorialPerson where
    state='FL' OR (lname='Smith' and fname='Joe')
//returns data that looks like this:
John Doe FL
Jim Smith FL
Jane Row FL
Joe Smith CA

Select fname, lname, state from TutorialPerson where state='FL' AND lname <> 'Smith'
//returns data that looks like this:
John Doe FL
Jane Row FL

Select fname, lname, state from TutorialPerson where state='FL' AND lname like 'Smi%'
//returns data that looks like this:
Jim Smith FL

Select fname, lname, state from TutorialPerson where
    state='FL' OR (lname='Smith' and fname='Joe') AND fname NOT LIKE 'Ji%'
//returns data that looks like this:
John Doe FL
Jane Row FL
Joe Smith CA
```

A interface suporta sugestões de índice SQL, o que dá ao usuário SQL a opção de especificar o índice HPCC mais apropriado para a consulta SQL atual. Isso também permite desativar o uso de um índice.

**select columnList de tableName USE INDEX(hpcc::index::file::name) where logicalExprssions**

**USE INDEX(none)** força o sistema a evitar a busca por um índice na consulta atual.

Exemplo:

```
Select fname, lname, zip, state from TutorialPerson
USEINDEX(TutorialPersonByZipIndex)where zip='33024'

//returns data that looks like this:
John Doe FL 33024
Jim Smith FL 33024
Jane Row FL 33024
```

Uma consulta Select retorna duas saídas de dataset por solicitação. O segundo dataset (chamado WsSQL-Count) fornece a contagem total de resultados. Isso é útil ao paginar resultados.

Exemplo:

```
<Dataset name='WsSQLResult'>
```

```
<Row>
  <yearbuilt>1203</yearbuilt>
  <id>4</id></Row>
<Row>
  <yearbuilt>2003</yearbuilt>
  <id>5</id></Row>
</Dataset>
<Dataset name='WsSQLCount'>
  <Row>
    <WSSQLSelectQueryResultCount>2</WSSQLSelectQueryResultCount>
  </Row>
</Dataset>
```

# SELECT JOIN

**select** *columnList* **from** *tableName* [**as** *alias*]

[<outer | inner > **JOIN** *join* *TableName* [ **as** *alias*] **on** *joinCondition*]

[**USE INDEX**(*indexFileName* | **NONE** )]

[**where** *logicalExpression*][**group by** *fieldName*]

[**order by** *columnNames* [asc | desc] ] [**LIMIT** *limitNumber*]

<i>columnList</i>	columnreference1[,columnreference2,columnreference3,...,columnreference]
	A(s) coluna(s) a retornar(em) (lista separada por vírgula). Além disso, essas funções agregadas são suportadas: COUNT, SUM, MIN, MAX, and AVG. Estes funcionam de uma maneira similar às suas contrapartes ECL
columnreference	[tablename.]columnname[ [AS] alias ]
<i>distinct</i>	[distinct] col1, col2,... coln
	O conjunto de resultados conterá apenas valores distintos (exclusivos)
<i>alias</i>	O alias usado para se referir à tabela ou referência de campo correspondente.
<b>outer   inner</b>	O tipo de JOIN a usar. Observação: O serviço WsSQL atualmente suporta INNER JOIN ou OUTER Joins. Um OUTER JOIN é convertido em um FULL OUTER JOIN internamente.
<i>joinTableName</i>	O arquivo JOIN a ser utilizado.
<i>joinCondition</i>	Especifica o relacionamento entre colunas nas tabelas associadas usando expressão lógica.
<i>logicalExpression</i>	Expressão lógica baseada na sintaxe padrão de filtragem SQL.
	BOOLEAN suporta apenas <i>Verdade</i> ou <i>Falso</i> , não use Y, N, 0 ou 1
	<b>Operadores válido:</b>
	= Equal (e.g., age=33)
	<> Not equal (e.g., age <>33)
	> Greater than (e.g., age >55)
	< Less than (e.g., age < 18)
	>= Greater than or equal (e.g., age >=21)
	<= Less than or equal (e.g., age <=21)
	IN(value1,value2,...,valuen) where values are comma separated homogeneous types.
	NOT IN(value1,value2,...,valuen) where values are comma separated homogeneous types.
	LIKE <i>pattern</i> where the pattern uses SQL LIKE operators with % and _ wildcards.
	NOT LIKE <i>pattern</i> where the pattern uses SQL LIKE operators with % and _ wildcards.

<i>limitNumber</i>	Opcional. O número de linhas para retornar. Isso substitui o atributo de configuração padrão ( <i>EclResultLimit</i> ) mas não pode ser definido para ALL.
--------------------	--

<sup>1</sup> Aliasing não suportado

<sup>2</sup> Pode conter apenas referências a funções agregadasse usado com a cláusula *HAVING*.

Funções agregadas só podem ser expressas em logicalExpressions usando *Group by HAVING*

Exemplos:

```
Select * from tableList where Sum(F1 > 100) /* is NOT SUPPORTED */
Select * from tableList Group by F1 Having Sum (F1 > 100) /* IS SUPPORTED */
```

Exemplo:

```
Select t1.personname, t2.address
  from persontable as t1 inner join adresstable as t2
    on (t1.personid = t2.personid AND
        (t1.firstname = 'jim' AND
         t1.lastname = 'smith' ))
```

A interface não converte os valores da lista de parâmetros ou da lista de colunas em literais de string.

Os valores das strings devem estar com aspas simples encapsuladas. O identificador de campo pode ser deixado sem aspas ou com aspas duplas.

Por exemplo, a tabela **pessoa** tem colunas lastname (String) e Zip (numérica)

```
Select Firstname from persons where lastname = 'Jones' and zip > 33445      /* works */
Select Firstname from persons where lastname = 'Jones' and "zip" > 33445    /* also works */
Select Firstname from persons where lastname = Jones and zip > 33445        /* doesn't work */
Select Firstname from persons where lastname = 'Jones' and zip > '33445'    /* doesn't work */
```

## CREATE / LOAD

**create table** [if not exists] *newtablename* '('(fieldname(fieldtype[fieldlen[,precision]]][UNSIGNED][,])+'

[comment 'commenttext'][:]

**load data infile** 'sourcetablename' [connection 'landingzoneIP' directory 'landingzonepath']

**into table** *newtablename*

[(fields | columns) [ terminated by 'fieldDelimiter'] [ enclosed by 'quoteChar'] [ escaped by 'escapeChar']]]

[lines [ terminated by 'recordDelimiter']][:]

As instruções CREATE TABLE e LOAD DATA SQL permitem criar um arquivo em um cluster HPCC que é acessível via WsSQL. Você pode preencher o novo arquivo a partir de dados em um arquivo lógico que já esteja no cluster HPCC ou em um que esteja em uma zona de entrada de arquivo.

Todos as solicitações de criação DEVEM ser acompanhados por uma solicitação de LOAD. Você não pode criar uma tabela com a intenção de carregar ou inserir dados posteriormente.

<i>newtablename</i>	O nome de arquivo lógico a ser criado no cluster HPCC. <b>Observação:</b> Create e LOAD devem segmentar o mesmo arquivo.
<i>commenttext</i>	Um comentário de texto para adicionar à descrição do arquivo lógico
<i>sourcetablename</i>	O arquivo de origem do qual extrair dados para carregar na nova tabela. Isso pode ser um arquivo lógico em um cluster HPCC ou em um arquivo físico em uma zona de entrada de arquivo. Para um arquivo da Zona de Entrada de Arquivos, você DEVE fornecer <b>uma conexão</b> 'landingzoneIP' e um <b>diretório</b> 'landingzonepath' (o caminho para o arquivo). Para um arquivo de zona de entrada de arquivo, use o caso correspondente para o nome do arquivo.
<i>landingzoneIP</i>	Endereço IP ou nome do host da Zona de Entrada de Arquivos do HPCC
<i>landingzonepath</i>	O caminho para o datafile no Landing Zonebut
<i>fieldDelimiter</i>	Valor string para o delimitador do campo.
<i>quoteChar</i>	Valor string para aspas.
<i>escapeChar</i>	Valor string para o ausência de caractere.
<i>recordDelimiter</i>	Valor string para o delimitador do registro.

Exemplos:

```
CREATE TABLE newCustomerFile (id DECIMAL( 30, 5), mytint INT(9),
                                mydouble DOUBLE (5,3) UNSIGNED)
COMMENT 'this file created via WsSQL and populated from file on lz';
LOAD DATA INFILE 'CustomerData' CONNECTION '127.0.0.1'
        DIRECTORY '/var/lib/HPCCSystems/mydropzone'
INTO TABLE newCustomerFile;

CREATE TABLE newCustomerFile2 (id DECIMAL( 30, 5), mytint INT(9),
                                mydouble DOUBLE (5,3) UNSIGNED)
COMMENT 'this file created via WsSQL and populated from sprayed file';
LOAD DATA INFILE 'thor::customerdata::customers' INTO TABLE newCustomerFile2

CREATE TABLE IF NOT EXISTS newCustomerFile3 (id DECIMAL( 30, 5), mytint INT(9),
```

```
mydouble DOUBLE (5,3) UNSIGNED)
COMMENT 'won't overwrite';
LOAD DATA INFILE 'thor::customerdata::customers' INTO TABLE newCustomerFile3

CREATE TABLE newCustomerFile4CSV (id DECIMAL( 30, 5), mytint INT(9),
mydouble DOUBLE (5,3) UNSIGNED)
COMMENT 'Loading CSV data';
LOAD DATA INFILE 'somecsvfile.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY 'n'
INTO TABLE newCustomerFile4CSV
```

## Funções Agregadas Suportadas

**COUNT**(*[DISTINCT]*columnName)

**DISTINCT**(columnName)

**SUM**(columnName)

**MIN**(columnName)

**MAX**(columnName)

**AVG**(columnName)

Essas funções agregadas são suportadas. Estas funcionam de uma maneira similar às suas contrapartes ECL Consulte a seção **Referência a Linguagem ECL** para detalhes.

COUNT	Conta as ocorrências de columnName no resultado, sempre um inteiro.
DISTINCT	Retorna somente valores distintos de columnName no resultado, o tipo de saída depende do tipo de entrada.
SUM	Retorna a soma dos valores de columnName no resultado, o tipo de saída depende do tipo de entrada.
MIN	Retorna o valor mínimo de columnName no resultado, o tipo de saída depende do tipo de entrada.
MAX	Retorna o valor máximo de columnName no resultado, o tipo de saída depende do tipo de entrada.
AVG	Retorna a média dos valores de columnName no resultado, sempre um número real.
columnName	A coluna para agregar.

Exemplo:

```
Select fname, lname, state, COUNT(zip) from TutorialPerson where zip='33024'
```

## Modificadores de strings suportados

**UPPER**(columnName)

**LOWER**(columnName)

UPPER	Retorna com todos os caracteres minúsculos convertidos em maiúsculos.
LOWER	Retorna com todos os caracteres maiúsculos convertidos em minúsculas.
columnName	A coluna para agregar.



## Considerações Especiais

Como esse serviço usa ECL e SQL, há algumas considerações especiais ao projetar os bancos de dados de back-end e stored procedures.

### Palavras-chave SQL reservadas

Seus identificadores de HPCC (nomes de campo, nomes de Stored Procedure, etc.) não podem usar palavras reservadas de SQL, como:

```
ADD ALL ANY AS ASC AT AVG BETWEEN BOOL BOOLEAN BY CALL COLUMN CONTAINS COUNT  
DESC DISTINCT FALSE FOR FROM GROUP HAVING IN INDEX INNER IS JOIN KEY KEYS LAST  
LEFT LIKE LIMIT LOWER MAX MIN MOD NOT NULL OFFSET ON ORDER OUT OUTER POWER SELECT  
SUM TABLE TRUE UPPER USE WHERE XOR DIV MOD OR AND
```

### Caracteres especiais.

Os nomes de arquivo do seu HPCC devem evitar caracteres especiais que não sejam:

```
( 'A'..'Z' | 'a'..'z' | ' ' | '$' ) ( 'A'..'Z' | 'a'..'z' | ' ' | '$' | '0'..'9' )  
Os nomes dos arquivos podem ser prefixados com um ~ (til) ou .::
```

Se um nome de arquivo não for suportado, ele não será retornado ao interrogar o sistema usando GetDB-MetaData.