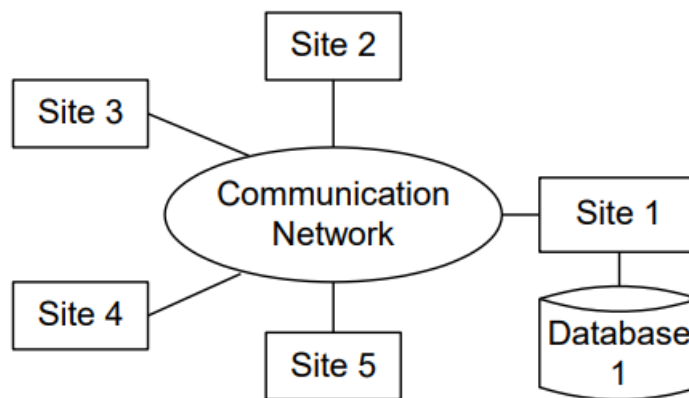


# 16. Distributed Databases

Este capítulo introduz as funções de gerenciamento de banco de dados distribuído fornecidas pelo DBMaker, incluindo bancos de dados distribuídos, a arquitetura distribuída, o acesso a dados distribuídos, o gerenciamento de objetos de banco de dados distribuídos e o gerenciamento de transações distribuídas.

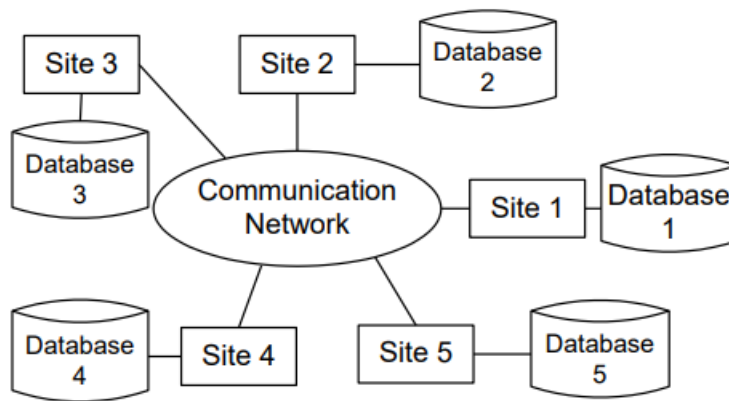
## 16.1 Introduction to Distributed Databases

Os SGBDs (Sistemas de Gerenciamento de Banco de Dados) cliente-servidor tradicionais, como mostrado na Figura 16-1, localizam o banco de dados em um computador específico na rede, e esse computador é responsável por lidar com todas as solicitações dos clientes.



Bancos de dados distribuídos, como mostrado na Figura 16-2, localizam uma cópia do banco de dados em vários computadores da rede, e cada um pode apoiar clientes de forma independente. O sistema de gerenciamento de banco de dados distribuído gerencia os bancos de dados nesses computadores, permitindo que os usuários acessem os dados de forma transparente.

O DBMaker suporta uma verdadeira arquitetura distribuída para fornecer um sistema de gerenciamento de banco de dados distribuído completo e robusto (DBMS Distribuído). Ele oferece conexões remotas de banco de dados, consultas distribuídas e gerenciamento de transações distribuídas. O DBMaker também fornece replicação de tabelas e bancos de dados para manter os dados automaticamente atualizados.



No ambiente de banco de dados distribuído do DBMaker, você pode escrever programas de aplicação usando a API compatível com ODBC 3.0 do DBMaker ou realizar consultas SQL ad-hoc que acessam dados de diferentes partes do banco de dados distribuído. O DBMaker integrará os dados de forma transparente e retornará os resultados, como se todos eles viessem de um banco de dados local.

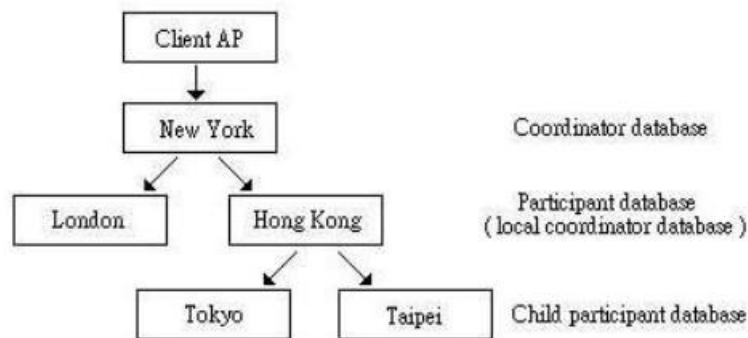
Neste capítulo, descreveremos brevemente a arquitetura do sistema e as funções básicas de gerenciamento de bancos de dados distribuídos usando o DBMaker. Isso inclui a configuração do ambiente distribuído, o gerenciamento de links de dados remotos e transações distribuídas, e a realização de consultas distribuídas. Seja você um administrador de banco de dados ou um desenvolvedor de aplicativos, este capítulo fornecerá uma visão completa da simplicidade e do poder da arquitetura distribuída do DBMaker.

## 16.2 Distributed Database Structure

O ambiente de banco de dados distribuído do DBMaker se baseia na arquitetura tradicional cliente/servidor, conectando efetivamente múltiplas aplicações cliente e múltiplos servidores de banco de dados. As aplicações cliente processam as solicitações dos usuários e exibem os resultados, enquanto os servidores de banco de dados gerenciam os dados. Cada cliente tem uma conexão direta com um único servidor de banco de dados, conhecido como Banco de Dados Coordenador para aquele cliente. Através do Banco de Dados Coordenador, o cliente pode se conectar a outros bancos de dados remotos, conhecidos como Bancos de Dados Participantes.

O DBMaker utiliza uma estrutura de distribuição hierárquica para conectar-se a bancos de dados remotos. Isso permite que o DBMaker acesse dados de um banco de dados remoto sem uma conexão direta com o Banco de Dados Coordenador, roteando através de um dos Bancos de Dados Participantes. Quando isso ocorre, o Banco de Dados Participante torna-se um Banco de Dados Coordenador Local,

atuando como coordenador para quaisquer bancos de dados filhos acessados por meio dele.



Na Figura 16-3, o programa de aplicação cliente conecta-se ao servidor de banco de dados em Nova York, o que torna o banco de dados em Nova York o Banco de Dados Coordenador. Se você usar o banco de dados em Nova York para acessar dados de Londres e Hong Kong, ambos os bancos de dados em Londres e Hong Kong são Bancos de Dados Participantes.

Alguns dos dados que você procura em Hong Kong podem, na verdade, estar nos bancos de dados em Tóquio ou Taipei. Assim, os bancos de dados em Tóquio e Taipei são Bancos de Dados Participantes Filhos. Isso faz com que o banco de dados em Hong Kong funcione como um Banco de Dados Coordenador para os bancos de dados em Tóquio e Taipei, tornando o banco de dados em Hong Kong não apenas um Banco de Dados Participante, mas também um Banco de Dados Coordenador Local.

## 16.3 Distributed Database Environment

Configurar um ambiente de banco de dados distribuído usando o DBMaker é tão simples quanto adicionar algumas palavras-chave ao arquivo dmconfig.ini para definir as opções de configuração do banco de dados distribuído. Opcionalmente, defina esses parâmetros usando a Ferramenta de Configuração JConfiguration. Para mais informações, consulte o Guia do Usuário da Ferramenta JConfiguration.

Você deve fornecer valores para as seguintes palavras-chave ao configurar um ambiente de banco de dados distribuído no DBMaker. Palavras-chave com o prefixo DB\_ são para a conexão cliente/servidor entre o cliente e o Banco de Dados Coordenador, e palavras-chave com o prefixo DD\_ são para as conexões de banco de dados distribuído entre o Banco de Dados Coordenador e os Bancos de Dados Participantes.

- DB\_SvAdr=<endereço\_ip/nome\_do\_host> — Endereço IP ou nome do host do Banco de Dados Coordenador.
- DB\_PtNum=<número\_da\_porta> — Número da porta que o aplicativo cliente e o Banco de Dados Coordenador devem usar para se comunicar.
- DD\_DDBMd=<0/1> — Habilita o modo de banco de dados distribuído para o Banco de Dados Coordenador. O valor padrão é 0, o que significa que o modo de banco de dados distribuído está desativado.
- DD\_CTimO=<número\_de\_segundos> — Tempo em segundos que o Banco de Dados Coordenador deve esperar ao tentar estabelecer uma conexão com um Banco de Dados Participante. O valor padrão é 5 segundos.
- DD\_LTimO=<número\_de\_segundos> — Tempo em segundos que o Banco de Dados Coordenador deve esperar ao tentar estabelecer um bloqueio nos dados solicitados em um Banco de Dados Participante. O valor padrão é 5 segundos.
- DD\_GTSvr=<0/1> — Habilita o daemon de recuperação de transação global (GTRECO). O valor padrão é 1, o que significa que o daemon de recuperação de transação global está habilitado.
- DD\_GTItv=<AAAA/MM/DD hh:mm> — Especifica o intervalo de tempo que o daemon de recuperação de transação global (GTRECO) deve esperar ao processar transações globais pendentes.

O DBMaker suporta um mecanismo de recuperação automática para transações distribuídas que falharam devido a problemas de rede ou erros no Banco de Dados Participante. O mecanismo de recuperação automática é gerenciado pelo daemon GTRECO, que verificará se um servidor de banco de dados distribuído tem algum problema com transações globais pendentes. Se algum problema for detectado, o daemon GTRECO tentará recuperar as transações globais pendentes. O daemon GTRECO é habilitado usando a palavra-chave DD\_GTSvr no arquivo dmconfig.ini.

Para entender melhor como o DBMaker gerencia bancos de dados distribuídos, consulte o exemplo a seguir.

Exemplo:

O ABC Bank tem duas filiais; uma em Los Angeles e outra em Seattle. Cada filial mantém seus próprios dados de clientes e negócios, mas a filial de Los Angeles controla os bancos de dados governamentais e financeiros.

O arquivo dmconfig.ini do servidor de banco de dados da filial de Los Angeles:

```
[BankTranx] ;LA branch business database
DB_DbDir = c:\\database
DB_SvAdr = 192.168.0.1
DB_PtNum = 21000
```

```
DD_DDBMd = 1
[BankMIS] ;government and financial database
DB_DbDir = c:\\database
DB_SvAdr = 192.168.0.1
DB_PtNum = 30000
DD_DDBMd = 1
[BankTranx@Seattle] ;Seattle branch business database
DB_SvAdr = 192.168.0.2
DB_PtNum = 21000
DD_CTimO = 20
DD_LTimO = 10
```

O arquivo dmconfig.ini do servidor de banco de dados da filial de Seattle:

```
[BankTranx] ;Seattle branch business database
DB_DbDir = c:\\database
DB_SvAdr = 192.168.0.2
DB_PtNum = 21000
DD_DDBMd = 1
[BankMIS] ;government and financial database
DB_SvAdr = 192.168.0.1
DB_PtNum = 30000
DD_CTimO = 20
[BankTranx@La] ;LA branch business database
DB_SvAdr = 192.168.0.1
DB_PtNum = 21000
DD_CTimO = 20
DD_LTimO = 10
```

O arquivo dmconfig.ini do servidor de aplicação cliente da filial de Los Angeles:

```
[BankTranx] ;LA branch business database
DB_SvAdr = 192.168.0.1
DB_PtNum = 21000
```

O arquivo dmconfig.ini do servidor de aplicação cliente da filial de Seattle:

```
[BankTranx] ;Seattle branch business database
DB_SvAdr = 192.168.0.2
DB_PtNum = 21000
```

Para habilitar o suporte a banco de dados distribuído, defina DD\_DDBMd = 1 na seção de configuração para o banco de dados local. Nos exemplos a seguir, coloque as palavras-chave na seção de configuração BankTranx dos arquivos dmconfig.ini das filiais de Los Angeles e Seattle.

Além disso, inclua uma seção de configuração do banco de dados para o Banco de Dados Participante no arquivo de configuração do Banco de Dados Coordenador, e para o Banco de Dados Coordenador no arquivo de configuração do Banco de Dados Participante. Neste caso, ambos os bancos de dados das filiais de Los Angeles e Seattle usam o mesmo nome de banco de dados. Se você usar o nome do banco de dados remoto como o nome da seção de configuração do banco de dados, isso causará um conflito com o nome do banco de dados local no arquivo dmconfig.ini.

Para evitar esse tipo de problema ao usar bancos de dados distribuídos, o DBMaker pode distinguir o nome do banco de dados remoto do nome do banco de dados local ao adicionar uma descrição do host do servidor ao nome do banco de dados remoto no arquivo dmconfig.ini local.

O nome do banco de dados remoto ficaria assim:

```
database_name@server_host_description
```

A descrição do host do servidor pode ser qualquer nome identificador, como o endereço IP ou o nome do host do servidor de banco de dados, o nome de domínio ou quase qualquer outro texto descritivo. Neste exemplo, o aplicativo cliente da filial de Los Angeles usaria BankTranx@Seattle quando quisesse acessar dados no banco de dados da filial de Seattle, e o aplicativo cliente da filial de Seattle usaria BankTranx@La quando quisesse acessar dados no banco de dados da filial de Los Angeles.

Além disso, configure o endereço do servidor e o nome da porta tanto para o banco de dados local quanto para o banco de dados remoto em suas respectivas seções de configuração nos arquivos de configuração das filiais de Los Angeles e Seattle. Neste exemplo, o arquivo de configuração da filial de Los Angeles conteria o endereço do servidor local na seção de configuração BankTranx, e conteria o endereço do servidor da filial de Seattle na seção de configuração BankTranx@Seattle. Da mesma forma, o arquivo de configuração da filial de Seattle conteria o endereço do servidor da filial de Los Angeles na seção de configuração BankTranx, e conteria o endereço do servidor da filial de Seattle na seção de configuração BankTranx@La.

Você também deve definir os parâmetros de conexão remota DD\_CTimO e DD\_LTimO. Esses parâmetros devem ser colocados na seção de configuração do

Banco de Dados Participante no arquivo de configuração do Banco de Dados Coordenador, e no Banco de Dados Coordenador no arquivo de configuração do Banco de Dados Participante.

Cada servidor de banco de dados na rede pode operar em objetos de banco de dados distribuídos. Qualquer um desses servidores de banco de dados pode ser acessado através do Banco de Dados Coordenador, de maneira semelhante à arquitetura normal cliente/servidor. Os comandos SQL que fazem referência a um banco de dados remoto serão passados para o servidor de banco de dados remoto através do Banco de Dados Coordenador. O Banco de Dados Coordenador irá decompor este comando SQL nas partes local e remota, e enviar os comandos apropriados para o servidor de banco de dados remoto. O Banco de Dados Coordenador aguardará o banco de dados remoto retornar os resultados, e então mesclará todos os dados locais e remotos e retornará os resultados combinados.

## 16.4 Distributed Database Objects

A DBMaker fornece vários métodos diferentes para acessar um Banco de Dados Participante:

- Especificar diretamente o nome do Banco de Dados Participante
- Utilizar links de banco de dados definidos no Banco de Dados Coordenador
- Através de mapeamento de objetos remotos, como visualizações ou sinônimos.

A diferença entre as duas primeiras abordagens é que os links de banco de dados contêm informações de segurança além do nome do banco de dados remoto. Isso permite que você especifique o nome de usuário e a senha que deseja usar no link de banco de dados ao acessar o banco de dados remoto.

Não há diferença óbvia entre as declarações de uma consulta distribuída e uma consulta normal, exceto na maneira como os objetos de banco de dados são especificados. No entanto, ao usar um banco de dados remoto, os únicos objetos de banco de dados remoto que podem ser acessados são tabelas, visualizações, sinônimos e comandos armazenados. Para acessar um objeto de banco de dados remoto, forneça o nome do banco de dados remoto ou link de banco de dados ao especificar o nome do objeto de banco de dados. Isso fornece duas maneiras de identificar um objeto de banco de dados remoto:

- `remote_database_name.object_name`
- `database_link.object_name`

Exemplo 1:

Para especificar um objeto de banco de dados remoto em uma consulta:

```
dmSQL> SELECT * FROM Bank:EmpTable;  
dmSQL> DELETE FROM Bank:EmpTable WHERE id = 101;  
dmSQL> INSERT INTO Link1:mis.account VALUES (2003,'Kevin Liu','2327-0021');
```

Exemplo 2:

Para acessar objetos de banco de dados remotos em dois Bancos de Dados Participantes diferentes:

```
dmSQL> SELECT * FROM ABCBank@La:account a,  
ABCBankMIS@Seattle:account b  
WHERE a.name = b.name;
```

Exemplo 3:

Para criar um comando armazenado chamado cmd1 no banco de dados remoto DB1 e, em seguida, executá-lo no banco de dados local DB2, siga os passos abaixo:

```
dmSQL> EXECUTE COMMAND DB1:cmd1(value);
```

## Remote Database Connections-Using Names

Os usuários podem se conectar a bancos de dados remotos com o nome do banco de dados do Servidor do Banco de Dados Coordenador. Os usuários devem conhecer o nome do banco de dados remoto, que é definido no arquivo dmconfig.ini no Servidor do Banco de Dados Coordenador. Exemplo 1:

Uma aplicação cliente na filial do banco ABC em Los Angeles acessa o banco de dados localizado na filial do banco ABC em Taipei. Parece que o usuário está se conectando à filial de Taipei com o nome de usuário SYSADM e a senha aa. Na verdade, o usuário está se conectando ao Banco de Dados Coordenador, ou seja, o banco de dados da filial de Los Angeles. O Banco de Dados Coordenador, então, conecta-se ao banco de dados remoto com a conta e a senha fornecidas.

```
dmSQL> CONNECT TO BankTranx SYSADM aa;  
dmSQL> SELECT * FROM BankTranx@Taipei:SYSADM.Account ORDER BY AccID;
```



## Exemplo 2:

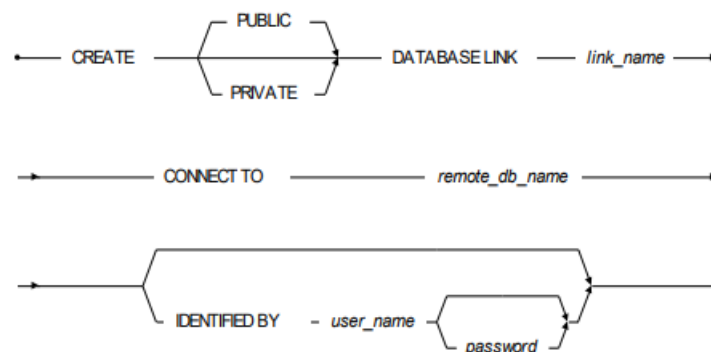
Usando junções para acessar objetos de banco de dados remoto:

```
dmSQL> SELECT * FROM BankMIS:SYSADM.Personnel ORDER BY PID;  
dmSQL> SELECT Personnel.* FROM BankTranx@Taipei:Account A,  
BankMIS:Personnel B  
WHERE A.CustID = B.CustID;
```

## Remote Database Connections-Using Links

Um link de banco de dados cria uma conexão com um banco de dados remoto e contém as informações de login e senha necessárias para a conexão. O link permite que os usuários se conectem a um banco de dados remoto com um nome de usuário diferente do Banco de Dados Coordenador ou se conectem a um banco de dados remoto sem uma conta. Ele também torna os dados em um ambiente de banco de dados distribuído transparentes em relação à localização. A definição do link, que também contém as informações de login e senha, é armazenada no Banco de Dados Coordenador.

### CREATING DATABASE LINKS



Somente administradores de banco de dados podem criar links públicos para serem utilizados por todos os usuários em um banco de dados. Qualquer usuário pode criar links privados para si mesmo. Os usuários podem criar links privados usando o mesmo nome. Um link privado substituirá um link público com um nome idêntico.

O DBMaker cria um link privado por padrão se o usuário não especificar o tipo. Se o usuário não especificar a conta de login e a senha na cláusula IDENTIFY BY, o nome e a senha de login atuais do usuário se tornam os valores padrão.

## REMOTE DATABASE OBJECTS & DATABASE LINKS

### Exemplo 1:

O seguinte mostra como acessar objetos de banco de dados remoto usando links de banco de dados. Neste exemplo, o SYSADM se conecta ao banco de dados e cria um link público chamado Bank\_Seattle, que se conecta ao banco de dados da filial de Seattle usando a conta SYSADM. O SYSADM atualiza alguns valores e se desconecta. Em seguida, o usuário1 se conecta e realiza uma consulta na tabela Account.

```
dmSQL> CONNECT TO BankTranx SYSADM;
dmSQL> CREATE PUBLIC DATABASE LINK Bank_Seattle CONNECT TO
BankTranx@Seattle
  2> IDENTIFIED BY SYSADM;
dmSQL> UPDATE Bank_Seattle:Account SET balance = balance + 100
  2> WHERE id = 1001;
dmSQL> DISCONNECT;
dmSQL> CONNECT TO BankTranx user1 pwd1;
dmSQL> SELECT * FROM Bank_Seattle:Account;
```

### Exemplo 2:

O SYSADM não especifica a conta a ser usada ao se conectar ao link público. Isso significa que, quando o usuário1 utiliza o link público para se conectar ao banco de dados Bank\_Seattle, deve haver uma conta para o usuário1 no banco de dados remoto, e a conta do usuário1 deve ter autoridade para consultar a tabela SYSADM.Account. Caso contrário, ocorrerá um erro.

```
dmSQL> CONNECT TO BankTranx SYSADM;
dmSQL> CREATE PUBLIC DATABASE LINK Bank_Seattle CONNECT TO
BankTranx@Seattle;
dmSQL> SELECT * FROM Bank_Seattle:Account;
dmSQL> DISCONNECT;
dmSQL> CONNECT TO BankTranx user1 pwd1;
dmSQL> SELECT * FROM Bank_Seattle:SYSADM.Account;
```

Se o nome de um link de banco de dados for o mesmo que o nome do banco de dados remoto, o DBMaker usará o nome do link de banco de dados em preferência ao nome do banco de dados remoto. Se você deseja acessar o banco de dados remoto diretamente, deve especificar o nome do banco de dados remoto na forma

dbname@''' para forçar o DBMaker a acessar o banco de dados remoto diretamente em vez de através do link de banco de dados.

Os seguintes 2 exemplos mostram diferentes maneiras de acessar um banco de dados remoto, uma através de um link de banco de dados e a outra especificando o nome do banco de dados remoto na forma dbname@''' .

Exemplo 1:

O SYSADM se conecta a um banco de dados remoto através de um link.

```
dmSQL> CONNECT TO BankTranx SYSADM;  
dmSQL> CREATE PUBLIC DATABASE LINK BankMIS CONNECT TO BankMIS  
2> IDENTIFIED BY SYSADM;  
dmSQL> DISCONNECT;
```

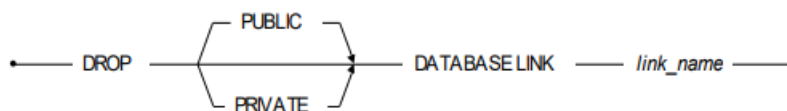
Exemplo 2:

O user1 se conecta a um banco de dados remoto usando a forma BankMIS@''':SYSADM.Personnel:

```
dmSQL> CONNECT TO BankTranx user1 pwd1;  
dmSQL> SELECT * FROM BankMIS:Personnel; //using database link  
dmSQL> SELECT * FROM BankMIS@''':SYSADM.Personnel; //using remote db  
name
```

**NOTA:** Ao acessar objetos de banco de dados remoto usando links de banco de dados e realizando algumas ações de UPDATE ou DELETE, não deve haver subconsultas. No momento, isso não é suportado pelo DBMaker.

## DELETING DATABASE LINKS



Somente administradores de banco de dados podem excluir links públicos, e apenas o proprietário de um link privado pode excluí-lo. Certifique-se de especificar o link público a ser excluído quando ele tiver o mesmo nome que um link privado, ou o DBMaker excluirá o link privado por padrão.

Exemplo:

Para excluir um link de banco de dados público chamado BankMIS:

```
dmSQL> DROP PUBLIC DATABASE LINK BankMIS;
```

## Database Object Mapping

O Mapeamento de Objetos de Banco de Dados proporciona melhor transparência de localização em um ambiente de banco de dados distribuído. Não há diferença entre a forma como um usuário acessa objetos de banco de dados remoto com o Mapeamento de Objetos de Banco de Dados e objetos de banco de dados locais. Esse tipo de Mapeamento de Objetos de Banco de Dados inclui o uso de visões e sinônimos.

## SYNONYMS

Usar um sinônimo para definir um objeto de banco de dados remoto é feito atribuindo ao objeto remoto um nome alternativo. Os privilégios que você possui no banco de dados remoto ao usar um sinônimo são os mesmos que em um banco de dados local.

Exemplo:

Para acessar um objeto de banco de dados remoto usando um sinônimo:

```
dmSQL> CONNECT TO BankTranx user1;
dmSQL> CREATE DATABASE LINK LK1 CONNECT TO BankMIS IDENTIFIED BY
user2;
dmSQL> CREATE SYNONYM s1 FOR BankTranx:Account;
dmSQL> CREATE SYNONYM s2 FOR LK1:user2.Personnel;
dmSQL> SELECT * FROM s1;
// SELECT * FROM BankTranx:user1.Account; (BankTranx, user1)
dmSQL> SELECT * FROM s2;
// SELECT * FROM LK1:user2.Personnel; (BankMIS, user2)
dmSQL> DISCONNECT;
dmSQL> CONNECT TO BankTranx user3;
dmSQL> CREATE DATABASE LINK LK1 CONNECT TO BankMIS IDENTIFIED BY
user4;
dmSQL> SELECT * FROM s1;
// SELECT * FROM BankTranx:user3.Account; (BankTranx, user3)
dmSQL> SELECT * FROM s2;
```

```
// SELECT * FROM LK1:user2.Personnel; (BankMIS, user4)
```

Os comentários indicam a expressão SQL equivalente, o banco de dados ao qual está sendo feita a conexão e a conta usada para se conectar.

## VIEWS

Usar uma visão para definir um objeto de banco de dados remoto é um pouco diferente de usar um sinônimo. A visão não é apenas um alias, mas inclui o nome do banco de dados, a conta de usuário, a senha, o proprietário do objeto e o nome do objeto como parte da definição. Os privilégios que um usuário possui no banco de dados remoto dependem dos privilégios do usuário que a criou.

Exemplo:

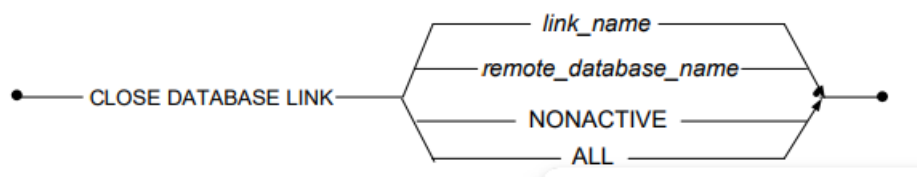
Para acessar um objeto de banco de dados remoto usando uma visão:

```
dmSQL> CONNECT TO BankTranx user1;
dmSQL> CREATE DATABASE LINK LK1 CONNECT TO BankMIS IDENTIFIED BY
user2;
dmSQL> CREATE VIEW v1 AS SELECT * FROM BankTranx:Account;
dmSQL> CREATE VIEW v2 AS SELECT * FROM LK1:user3.Personnel;
dmSQL> SELECT * FROM v1;
// SELECT * FROM BankTranx:user1.Account; (BankTranx, user1)
dmSQL> SELECT * FROM v2;
// SELECT * FROM BankMIS:user3.Personnel; (BankMIS, user2)
dmSQL> DISCONNECT;
dmSQL> CONNECT TO BankTranx user3;
dmSQL> CREATE DATABASE LINK LK1 CONNECT TO BankMIS IDENTIFIED BY
user4;
dmSQL> SELECT * FROM v1;
// SELECT * FROM BankTranx:user1.Account; (BankTranx, user1)
dmSQL> SELECT * FROM v2;
// SELECT * FROM LK1:user3.Personnel; (BankMIS, user2)
```

Os comentários indicam a expressão SQL equivalente, o banco de dados ao qual está sendo feita a conexão e a conta usada para se conectar.

## Closing Links

Uma vez que um usuário acesse um banco de dados remoto com um comando SQL, o Banco de Dados Coordenador irá estabelecer uma conexão remota com o Banco de Dados Participante. A conexão remota permanecerá aberta até que todos os usuários se desconectem do Banco de Dados Coordenador ou até que o link seja fechado com o comando `CLOSE DATABASE LINK`. O DBMaker fornece até 256 conexões remotas para cada banco de dados; é uma boa prática fechar conexões remotas que não estão mais em uso para liberar as conexões para outros usuários.



### Exemplo 1:

Para fechar um link de banco de dados usando o nome do banco de dados remoto BankMIS:

```
dmSQL> CLOSE DATABASE LINK BankMIS;
```

### Exemplo 2:

Para fechar um link de banco de dados usando o link de banco de dados remoto BankLink1:

```
dmSQL> CLOSE DATABASE LINK BankLink1;
```

Quando um usuário emite um comando `CLOSE DATABASE LINK`, o DBMaker diminuirá o contador de conexões remotas em um. Quando o contador atingir zero, a conexão será totalmente fechada e os recursos ocupados serão liberados. Caso contrário, a conexão permanecerá aberta.

### Exemplo 3:

Para fechar todos os links de banco de dados e liberar as conexões e recursos:

```
dmSQL> CLOSE DATABASE LINK ALL;
```

### Exemplo 4:

Para fechar todas as conexões remotas NÃO ATIVAS que não estão mais sendo utilizadas:

```
dmSQL>          CLOSE          DATABASE          LINK          NONACTIVE;
```

## Link System Catalog Tables

Duas tabelas de catálogo do sistema estão relacionadas a links de banco de dados: SYSDBLINK e SYSOPENLINK. A SYSDBLINK registra todos os nomes de links de banco de dados e suas definições, enquanto a SYSOPENLINK registra as conexões abertas entre os bancos de dados.

## 16.5 Distributed Transaction Control

O DBMaker suporta um mecanismo de transação distribuída que é transparente para os usuários. Os Bancos de Dados Participantes não realizam o compromisso apenas de uma parte de uma transação distribuída; o DBMaker gerencia isso.

Exemplo:

O seguinte mostra como o controle de transações distribuídas funciona.

```
dmSQL> CONNECT TO BankTranx user1; // ABC Bank in Taipei
dmSQL> SET AUTOCOMMIT OFF;
dmSQL> UPDATE BankTranx:Customer SET money=money-1000 WHERE id=123;
dmSQL> UPDATE BankTranx@"Bank_in_Seattle":Customer SET
money=money+1000
  2> WHERE id=123;
dmSQL> COMMIT;
```

Como o Banco de Dados Coordenador gerencia todas as operações de banco de dados da aplicação cliente, ele conhece o escopo das instruções por meio do Gerenciador de Catálogo de Sistema Distribuído. O Banco de Dados Coordenador lidará com transações pertencentes ao site local da mesma forma que uma transação regular cliente/servidor. As transações pertencentes a sites remotos precisam fazer referência ao banco de dados remoto apropriado. O Banco de Dados Coordenador trocará informações com cada Banco de Dados Participante e coordenará toda a transação até que ela seja revertida ou comprometida.

## **Two-Phase Commit**

Os sistemas de gerenciamento de banco de dados precisam manter a integridade dos dados, e isso requer que todas as transações sejam atômicas. Todas as operações na transação devem ser comprometidas ou revertidas juntas. Na arquitetura tradicional cliente/servidor, um diário é utilizado para garantir que as alterações sejam ou revertidas ou comprometidas. Na arquitetura de banco de dados distribuído, um protocolo de commit em duas fases com abortamento presumido é usado como mecanismo para controlar transações distribuídas que abrangem vários servidores de banco de dados. Uma transação que modifica dados em dois ou mais bancos de dados deve completar o protocolo de commit em duas fases antes de ser comprometida. O mecanismo de commit em duas fases garante que todos os sites comprometam ou revertam globalmente. Ele também protege as operações de manipulação de dados realizadas por sinônimos remotos, restrições de integridade e gatilhos. Para comprometer uma transação, um usuário deve garantir que cada sub-transação tenha sido concluída; caso contrário, a transação será abortada. Pela mesma razão, se alguma sub-transação não puder ser comprometida, as outras sub-transações também devem ser abortadas.

## **Distributed Transaction Recovery**

O DBMaker utiliza o protocolo de commit em duas fases para informar a todos os Bancos de Dados Participantes para comprometer uma transação global. Antes de entrar na fase de commit, o Banco de Dados Coordenador verifica o status dos Bancos de Dados Participantes para garantir que não há problemas de servidor ou de rede. Se o Banco de Dados Coordenador encontrar um problema com qualquer um dos Bancos de Dados Participantes, ele informa os outros Bancos de Dados Participantes para reverter sua parte da transação e retorna um erro indicando que a transação global falhou. Se o protocolo de commit em duas fases concluir a fase de preparação, mas houver um problema de servidor ou de rede com um Banco de Dados Participante, a transação global é considerada um sucesso. O DBMaker registra qual servidor de Banco de Dados Participante não conseguiu comprometer sua parte da transação na tabela de catálogo de sistema SYSGLBTRANX e registra qual banco de dados contém uma transação pendente na tabela de catálogo de sistema SYSPENDTRANX para o banco de dados que falhou.

O DBMaker também fornece um mecanismo de recuperação automática para lidar com falhas de rede ou de site durante a execução de uma transação distribuída. No Banco de Dados Coordenador, você inicia o daemon de recuperação de transações globais (GTRECO). Este daemon escaneia a tabela de catálogo de sistema SYSGLBTRANX e periodicamente recupera quaisquer transações globais pendentes.



Em seguida, ele tenta se conectar ao Banco de Dados Participante que falhou e o informa para comprometer ou reverter sua parte da transação global.

## **Heuristic End Global Transaction**

Após uma falha de rede ou de site ocorrer durante o commit em duas fases, as transações pendentes continuam a manter alguns recursos, como bloqueios ou blocos de diário. A transação pendente ocupará esses recursos até que o problema seja resolvido pelo daemon de recuperação global (GTRECO). Se a falha de rede ou de site não puder ser resolvida imediatamente, os recursos retidos podem bloquear alguns usuários no site do Participante. Para resolver esse problema, o DBMaker suporta o encerramento heurístico de transações globais. Um encerramento heurístico de transação é uma ação independente tomada pelo administrador do banco de dados para forçar uma transação pendente em um Banco de Dados Participante a comprometer ou reverter. O administrador do banco de dados pode usar o JDBATool para resolver esse problema. Consulte o Guia do Usuário da Ferramenta JDBA para mais informações.

Para resolver uma transação pendente manualmente, execute o seguinte:

1. No Banco de Dados Participante, navegue até a tabela SYSPENDTRANX para descobrir se há transações que estão pendentes há muito tempo.
2. No Banco de Dados Coordenador, determine o status de commit da transação pendente na SYSGLBTRANX. Também determine o status de commit da transação pendente no Banco de Dados Participante. Por exemplo, se houver duas transações pendentes "DB\_1-3376aafd" e "DB\_2-3376aafd#1", o administrador do banco de dados coordenador deve pedir ao administrador do DB\_1 para determinar o status de "DB\_1-3376aafd" e ao administrador do DB\_3 para determinar o status de "DB\_2-3376aafd#1".
3. No Banco de Dados Coordenador, quando o administrador receber a consulta sobre o status da transação, ele pode examinar a SYSGLBTRANX para determinar o status da transação. Se o STATE for 2 (COMMIT) ou 3 (PENDCOM), responda "commit". Se o STATE for 4 (PENDABO), responda "abort". Mas se o STATE for 1 (PREPARE), este ramo da transação está pendente neste site também, e peça ao administrador do site pai para determinar seu status.

No Banco de Dados Participante, o administrador usa o JServer Manager para realizar um commit heurístico ou abortar a transação pendente com base na resposta.

Se o administrador iniciar um encerramento heurístico de transação em uma transação pendente que for diferente da ação tomada no Banco de Dados Coordenador, os dados distribuídos ficarão inconsistentes.