

Ao final deste assunto, você será capaz de:

- Compreender o conceito de banco de dados distribuído;
- Conhecer os critérios de distribuição de banco de dados e seus objetivos;
- Compreender o conceito e objetivo do uso de transações descentralizadas.

1.3.1. Banco de Dados Centralizado

Considere um sistema constituído dos seguintes componentes:

- Equipamento de processamento (CPU);
- Sistema operacional instalado nessa CPU;
- SGBD instalado na mesma CPU;
- Aplicação que acessa o SGBD.

Ora, todos esses componentes instalados e operando em uma única CPU constituem um sistema centralizado com o seguinte esquema:

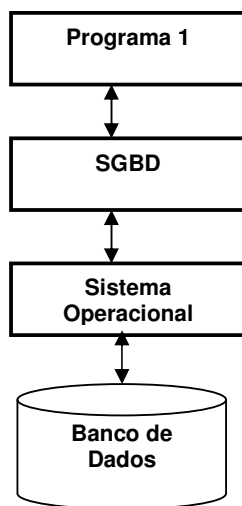


Fig. 1.3. 1 – Sistema de Banco de Dados Centralizado.

Se colocarmos essa CPU ligada em rede com outras, mantendo o SGBD, os dados e a aplicação na mesma CPU, continuaremos tendo um sistema centralizado com execução remota (teleprocessamento).

1.3.2. Aplicação Distribuída

Se, porém, mantivermos o SGBD e os dados na mesma CPU e copiarmos a aplicação para as outras CPU's da rede, acessando esses mesmos dados, passaremos a ter dados centralizados com aplicação distribuída.

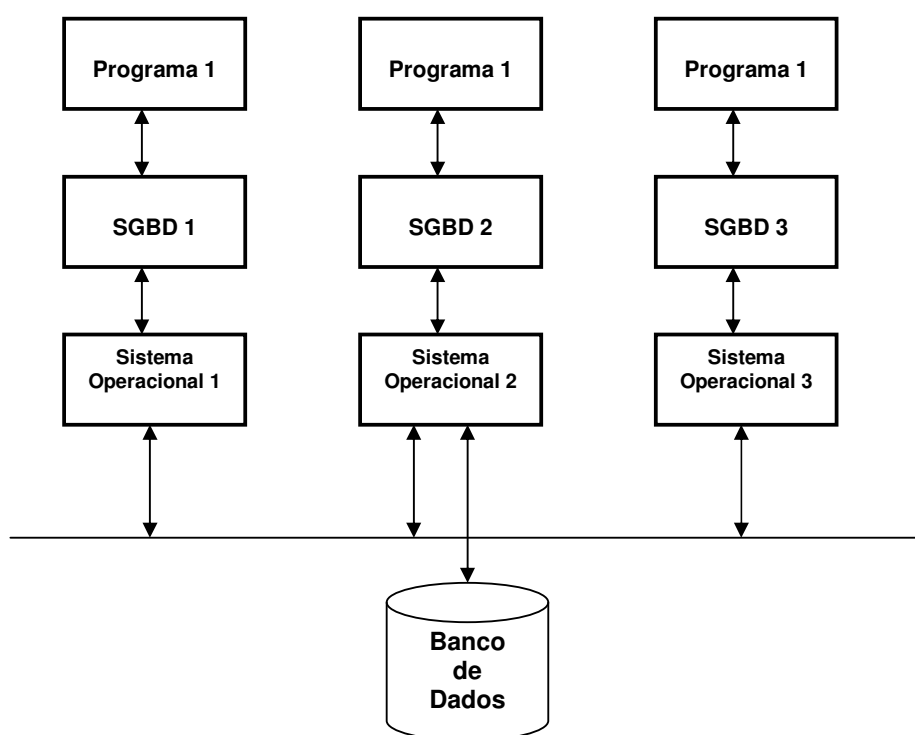


Fig. 1.3. 2 - Aplicação Distribuída.

1.3.3. Banco de Dados Distribuído

No entanto, se decidirmos que cada unidade descentralizada terá somente os seus dados na sua CPU local, então teremos um sistema de dados distribuídos.

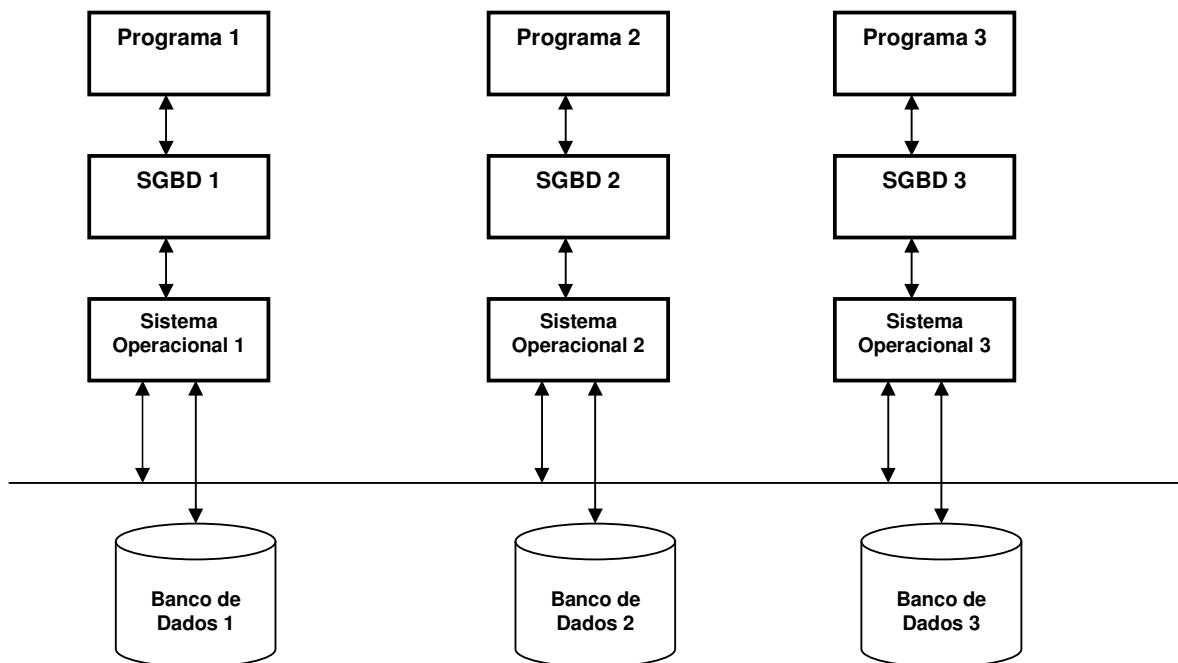


Fig. 1.3. 3 – Sistema de Banco de Dados Distribuído.

O problema começa quando alguém determina que é necessário que se tenha a consolidação dos dados dos diversos bancos de dados. Ou seja, se for necessário agregar numa única aplicação dados relativos a todas as unidades descentralizadas, como no caso de um banco. Neste caso, será necessário que a aplicação acesse cada banco de dados e vá consolidando gradativamente os dados até alcançar o último banco de dados. Se houver interrupção da comunicação com qualquer unidade, os dados dessa não farão parte do conjunto consolidado e, a cada nova consolidação, sempre haverá a possibilidade de ocorrer interrupção de comunicação com outra e mais outra unidade e assim indefinidamente. Que dizer, pode haver a possibilidade de nunca se conseguir fazer toda a consolidação, numa luta constante contra as “leis de Murphy”.

O que fazer, então?

Uma alternativa poderia ser a definição de uma máquina que contenha uma cópia de cada banco de dados e fazer a consolidação de dados nessa máquina.

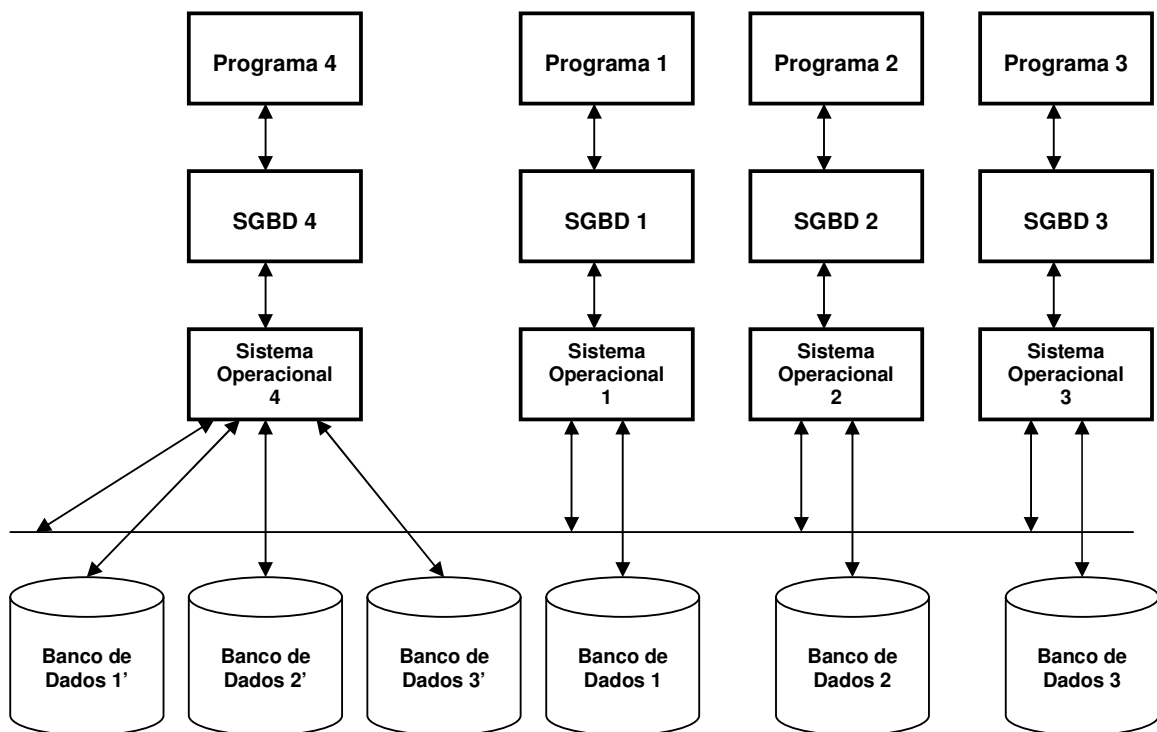


Fig. 1.3. 4 - Banco de Dados Consolidado.

É claro que, mesmo essa solução, não é muito simples, pois implica na necessidade de manter o sincronismo entre todos os sistemas de modo que o sistema consolidado possa sempre representar o conjunto, ou seja, estar atualizado. Neste caso, temos distribuição de dados, ou ainda, temos diversos sistemas de bancos de dados espalhados pela rede e consolidados em um só local.

O sincronismo entre os dados do ambiente 4 e os demais pode ser feito de diversas formas:

- Síncrona: sempre que uma aplicação atualizar os dados no seu banco de dados, esta atualização será entregue ao banco de dados do ambiente 4. Por exemplo, se a aplicação 1 entregar uma atualização para o SGBD1, ela terá que entregar a mesma atualização para o SGBD4.

- Assíncrona: quando uma aplicação fizer uma atualização no seu banco de dados, esta atualização será armazenada em algum lugar e cada grupo de atualizações, estas serão entregues ao banco de dados do ambiente 4. Por exemplo, a aplicação 1 entrega diversas atualizações para o SGBD1 e, a cada 10 atualizações ou de 10 em 10 minutos, essas atualizações são enviadas ao SGBD4. Ou seja, o banco de dados do ambiente 4 pode ser atualizado assincronamente de duas formas:
 - o Em blocos: a cada conjunto de atualizações, atualiza-se o banco de dados central;
 - o Por tempo: a cada intervalo de tempo as atualizações são efetuadas no banco de dados central.

Se a consolidação dos dados no ambiente 4 tiver que ser feita por uma aplicação que rode no ambiente 4, teremos distribuição de dados mas não teremos um sistema de banco de dados distribuídos. Caso o SGBD4 reconheça os demais SGBDs (SGBD1, SGBD2 e SGBD3) e, mediante um acesso a ele, consiga determinar se a resposta está em um dos demais SGBDs, neste caso teremos um sistema de banco de dados distribuídos.

Concluindo

A tecnologia dos bancos de dados distribuídos é um desenvolvimento comparativamente recente no campo global dos bancos de dados.

Um banco de dados distribuído é tipicamente aquele que não é inteiramente armazenado em uma única localização física, estando disperso através de uma rede de computadores geograficamente afastados e conectados por elos de comunicações. Como um exemplo (muito simplificado), consideremos o sistema de um banco no qual o banco de dados das contas dos clientes esteja distribuído pelas agências desse banco, de tal forma que cada registro individual de conta de cliente se encontre armazenado na agência local do cliente. Em outras palavras, o dado esteja armazenado no local no qual é mais freqüentemente usado, mas ainda assim

disponível via rede de comunicações, a usuários de outros locais (por exemplo, usuários da agência central do banco).

As vantagens dessa distribuição são claras: combinam a eficiência do processamento local (sem sobrecarga de comunicações) na maioria das operações, com todas as vantagens discutidas anteriormente (em particular, o compartilhamento dos dados) apresentadas por um sistema centralizado. Mas, naturalmente, também há desvantagens: podem ocorrer elevadas sobrecargas nas comunicações, além de dificuldades técnicas significativas para se implementar esse sistema.

Um objetivo primordial em um sistema distribuído é o de que ele *pareça ser, ao usuário, um sistema centralizado*. Isto é, normalmente o usuário não precisará saber onde se encontra fisicamente armazenada determinada porção dos dados (ficando as aplicações independentes da maneira pela qual os dados são distribuídos, tornando possível uma mudança de distribuição sem que as aplicações sejam afetadas - outro aspecto da independência de dados). Portanto o fato de ser o banco de dados distribuído só deve ser relevante ao nível interno, e não aos níveis externo e conceitual. Até bem pouco tempo, poucos sistemas haviam avançado na direção deste objetivo, porém hoje os SGBDs mais famosos no mercado já disponibilizam versões que implementam essa filosofia de trabalho.

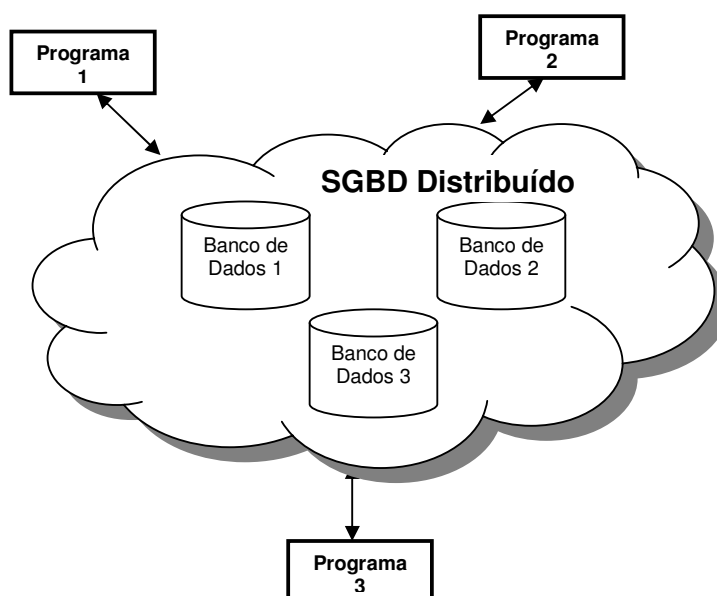


Fig. 1.3. 5 - SGBD Distribuído.

1.3.4. Formas ou Critérios de distribuição

Considerando o seguinte exemplo de um banco de dados:

Banco de Dados: Conta-Corrente

Arquivo: Correntista

Conta	Agência	Nome	Saldo
123	12	FULANO	100,00
142	12	SICRANO	200,00
154	13	BELTRANO	150,00
160	14	PEDRO	200,00
165	13	ANTONIO	1000,00

A distribuição de banco de dados pode ser feita utilizando os seguintes critérios:

- ***Fragmentação:*** O banco de dados é fatiado de acordo com características dos dados necessários nos locais. Pode ser:
 - ***Fragmentação horizontal*** – registros (linhas) são distribuídos (separados) por locais, segundo critério definido. Essa regra de distribuição é definida no banco de dados via uma DDL.

Ex.: O banco distribui os seus dados por agência de modo que cada agência tem seus dados localizados no banco de dados local.

Banco de Dados: Conta-Corrente (Agência 12)

Arquivo: Correntista

Conta	Agência	Nome	Saldo
123	12	FULANO	100,00
142	12	SICRANO	200,00

Banco de Dados: Conta-Corrente (Agência 13)

Arquivo: Correntista

Conta	Agência	Nome	Saldo
154	13	BELTRANO	150,00
165	13	ANTONIO	1000,00

Banco de Dados: Conta-Corrente (Agência 14)

Arquivo: Correntista

Conta	Agência	Nome	Saldo
160	14	PEDRO	200,00

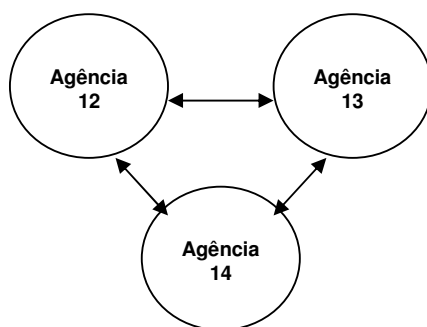


Fig.2. 11 - Banco de Dados Distribuído com Fragmentação Horizontal.

- **Fragmentação vertical** – campos (colunas) de todos os registros (linhas) são distribuídos (separados) por locais e reconectados entre os diversos bancos.

Ex.: O banco distribui os seus dados separando os campos por dois bancos de dados, mantendo um elo de ligação entre eles (a conta).

Banco de Dados: Conta-Corrente (Banco 1)

Arquivo: Correntista

Conta	Agência	Nome
123	12	FULANO
142	12	SICRANO
154	13	BELTRANO
160	14	PEDRO
165	13	ANTONIO

Banco de Dados: Conta-Corrente (Banco 2)

Arquivo: Correntista

Conta	Saldo
123	100,00
142	200,00
154	150,00
160	200,00
165	1000,00

- Fragmentação mista – registros e campos são fragmentados segundo algum critério. É feita fragmentação horizontal e vertical no mesmo banco de dados.

Ex.:

Banco de Dados: Conta-Corrente (Banco 1)

Arquivo: Correntista

Conta	Agência	Nome
123	12	FULANO
142	12	SICRANO

Banco de Dados: Conta-Corrente (Banco 2)

Arquivo: Correntista

Conta	Agência	Nome
154	13	BELTRANO
165	13	ANTONIO
160	14	PEDRO

Banco de Dados: Conta-Corrente (Banco 3)

Arquivo: Correntista

Conta	Agência	Saldo
123	12	100,00
142	12	200,00

Banco de Dados: Conta-Corrente (Banco 4)

Arquivo: Correntista

Conta	Agência	Saldo
154	13	150,00
165	13	1000,00
160	14	200,00

- **Duplicação ou Réplica:** cópias locais de arquivos locais. O banco todo é duplicado ou os arquivos são duplicados e distribuídos. As réplicas não são alteráveis e tem que ser re-sincronizadas periodicamente.

Ex.: O banco distribui réplicas do todo em alguns pontos de consulta.

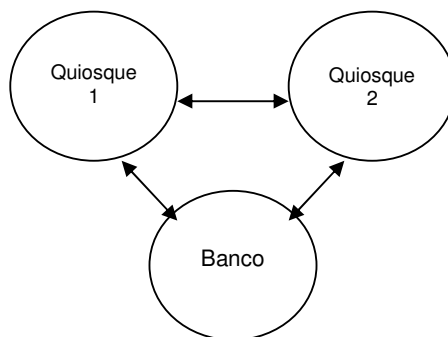


Fig. 1.3. 6 - Distribuição em réplicas.

- **Fragmentação com Duplicação:** neste caso, tanto são efetuadas tanto fragmentações como réplicas.

Ex.: O banco distribui seus dados pelas suas 3 agências, mantendo dois quiosques de consultas para uma das agências. Essa agência (2) terá terminais para consultas nos dois quiosques, que precisarão ser atualizados periodicamente.

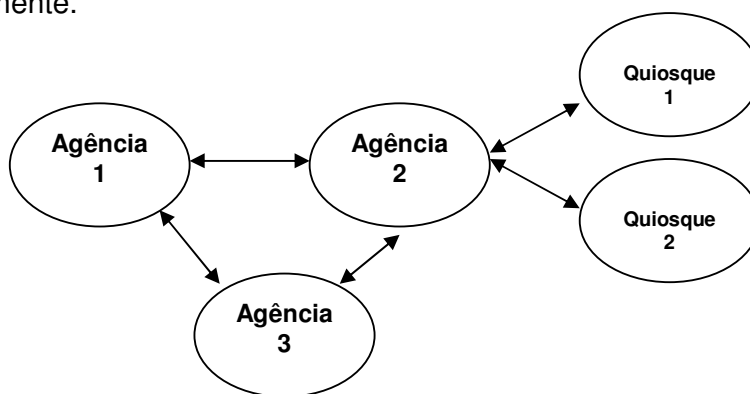


Fig. 1.3. 7 - Distribuição com Fragmentação e Réplica.

1.3.5. Transações descentralizadas

O conjunto de atualizações aplicado ocorre entre os diversos bancos que compõem o sistema distribuído.

Requisitos:

- Necessidade de um coordenador de transações (o nó que iniciou a transação)
- Necessidade de um protocolo de compromissamento que permita que os diversos bancos de dados mantenham sua integridade ao fim da transação.

Exemplos:

- Two-phase commit, também chamado de 2PC ou 2-phase commit;
- Three-phase commit, também chamado de 3PC ou 3-phase commit.

Algoritmo do Two-phase commit

Considere que:

- A transação distribuída envolve 3 nós (N1, N2 e N3);

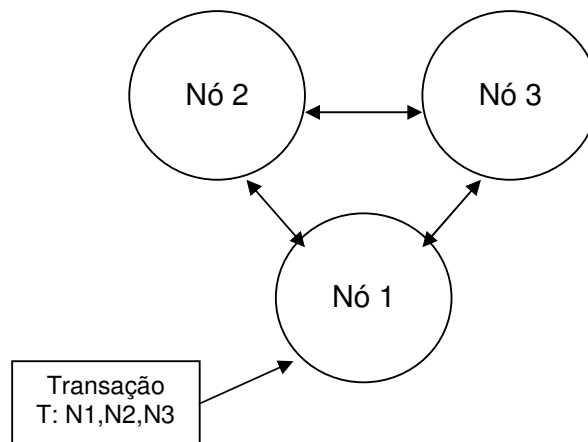


Fig. 1.3. 8 - Two-phase commit.

- N1 começa a transação T, que envolve o nó N2 e N3. N1 passa a ser o coordenador de T (C1, dentro das funções descentralizadas do SGBD)
- Todos os nós (N1, N2 e N3) informam a C1 que efetuaram a sua parte (atualização) e C1 inicia o protocolo de compromissamento

Fase 1:

- C1 gera no seu log <Prepare T>.
- Envia <Prepare T> para N1, N2 e N3.
- Cada SGBD Local recebe a mensagem e se compromete <ready T> ou não <abort T>. O SGBD gera <ready T> ou <abort T> no seu log. Envia essa mensagem para C1.

Fase 2:

- C1 recebe as respostas.
- Se todos responderam <ready T>, C1 manda <commit T> para todos participantes.
- Se algum nó respondeu <abort T>, C1 envia <abort T> para todos os participantes.
- Tanto o coordenador como os participantes esperam um tempo determinado pelas mensagens. Passado esse tempo, enviam um <abort T>.
- Se C1 demorar a responder para o participante, o participante gera <abort T>.
- Se o participante demora a concluir uma fase, C1 gera <abort T> para todos.
- Se ocorrer queda de comunicação na rede pode ser que nunca uma transação se complete.

ATIVIDADE

Responda as questões a seguir:

1. Conceitue Sistema de Banco de Dados Distribuído.
2. Cite uma vantagem de usar SGBDDs ao invés de fazer o controle de distribuição pela aplicação.
3. Qual critério de distribuição de banco distribuído busca distribuir as tuplas de uma tabela mantendo o esquema dessa tabela da mesma forma que se o banco fosse centralizado?
4. Qual critério de distribuição faz com que se tenha todas as tuplas mas somente alguns atributos em alguma localidade?
5. Qual o objetivo do protocolo 2-phase commit?