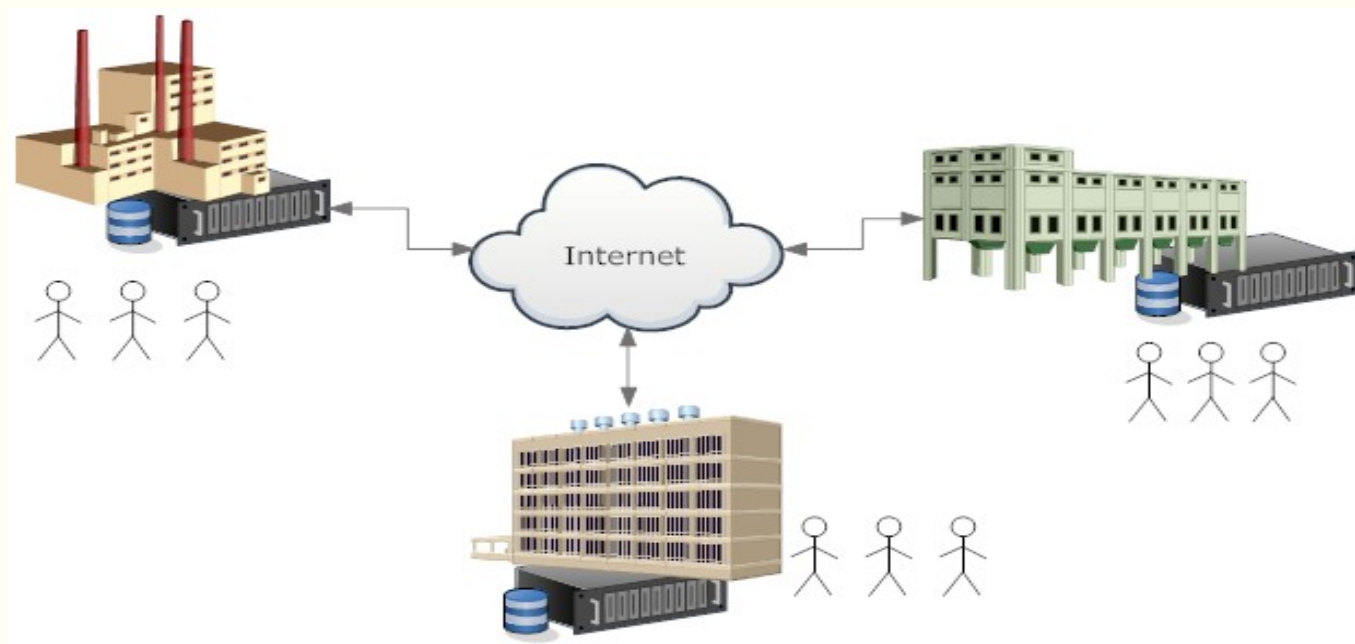




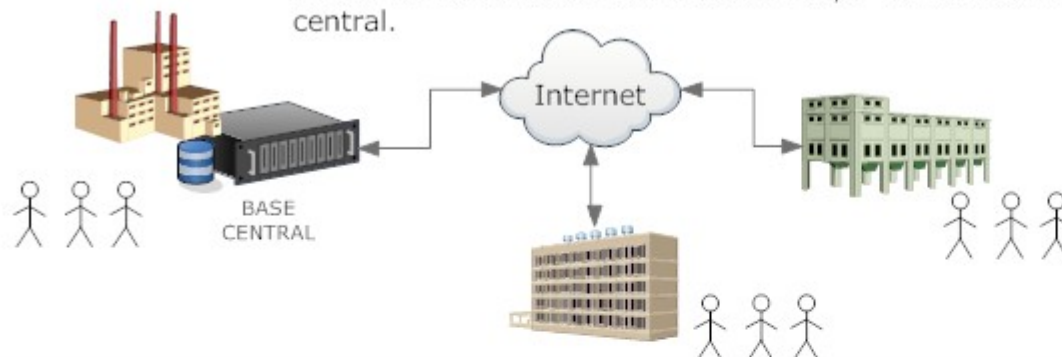
Replicação na Prática

Como transformar seu banco de dados Firebird para poder trabalhar de forma distribuída e multi-master em N servidores



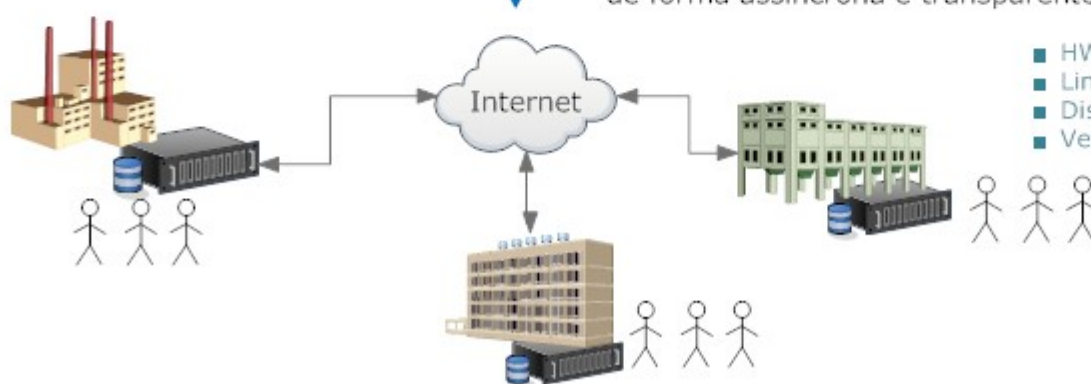


Aplicação Centralizada, com usuários remotos usando o sistema ou via Terminal Server ou conexão TCP/IP remota com o banco central.



- HW Servidor Central de alto custo
- Link Internet de alto custo
- Indisponibilidade frequente
- Lentidão

Aplicação Descentralizada, com usuários remotos usando o sistema no servidor local, e os dados replicados de forma assíncrona e transparente para o usuário.



- HW Servidores Baixo Custo
- Link Internet de baixo custo
- Disponibilidade independe de rede
- Velocidade de rede local





- Pontos a serem observados na fase de planejamento, independente da ferramenta de replicação de banco de dados a ser utilizada
 - Qual Topologia de replicação adotar ?
 - Estrela: Uma base central e N bases remotas, boa opção quando se quer uma base central consolidando os dados de todas as filiais.
 - Ponto-a-Ponto: N bases onde cada uma envia dados para todas as outras, sem uma base central.
 - Hierárquico: Quando se deseja uma hierarquia de servidores, por exemplo, na camada mais externa servidores por filiais, na intermediária servidores regionais, e no topo da hierarquia um servidor principal.
 - **Dica:** Use topologia estrela com no máximo 250 bases remotas, acima disso recomendamos partir para uma topologia hierárquica.





- Pontos a serem observados na fase de planejamento, independente da ferramenta de replicação de banco de dados a ser utilizada
 - Replicação Multi-Master ou Master-Slave ?
 - Multi-Master (bidirecional) é quando se deseja que os usuários possam alterar dados em todos os servidores
 - Master-Slave (unidirecional) é quando se deseja que as atualizações sejam feitas apenas em uma das bases e que as outras fiquem disponíveis remotamente apenas para consultas
 - **Dica:** Em algumas situações você pode adotar Multi-Master em uma parte do sistema e Master-Slave em outra parte (devido a problemas de concorrência, etc)





- Pontos a serem observados na fase de planejamento, independente da ferramenta de replicação de banco de dados a ser utilizada
 - Quais Tabelas replicar e em qual direção ?
 - Exemplo: Tabelas de setup de sistema e/ou produtos, preços, etc, na direção apenas matriz para filiais e não no sentido contrário. Tabelas de movimento de vendas, etc, apenas no sentido filiais para matriz, ou replicar tudo para todas as bases.
 - **Dica:** Se o volume de dados não for muito grande e a banda de rede permitir replique todas as tabelas e em todas as direções, pois assim você "ganha" um backup full em cada site da empresa.





- Pontos a serem observados na fase de planejamento, independente da ferramenta de replicação de banco de dados a ser utilizada
 - Planejar testes das operações mais críticas
 - Pedidos de compra, venda, emissões de NF, etc.
 - Realizar testes com no mínimo 3 servidores (ou 3 bases no mesmo servidor). Com 2 servidores muitos problemas não são detectados.
 - Os testes devem incluir tabelas com todos os tipos de dados usados no sistema, principalmente dados como datas, blobs, etc.
 - Faça um "script" formal de toda sequência de testes e o conteúdo esperado nas tabelas em todos os servidores
 - Ao executar o "script" simule queda de um servidor (desligue-o da tomada), simule queda de rede (tire o cabo de rede), veja se após ligado novamente os dados são replicados corretamente
 - **Dica:** Os testes realmente são necessários, e não testes aleatórios mas sim testes bem planejados – um dos grandes problemas que encontramos em algumas instalações foi descobrir problemas depois que a replicação já está em uso com o sistema em produção.





- Quais os pré-requisitos a nível de sistema ?
 - Todas as tabelas a serem replicadas devem ter chave primária ou algum índice único.
 - Analisar as tabelas do sistema e verificar se existe tabela sem chave primária. Veja se é possível determinar uma chave primária com as colunas existentes na tabela, e no caso de não ser possível criar uma coluna do tipo ID e associar com uma GENERATOR.
 - A maioria dos replicadores assíncronos vai exigir este pré-requisito.





- Quais os pré-requisitos a nível de sistema ?
 - Se tiver tabelas com chaves primárias formadas por IDs gerados por generators, max(), tabela de contadores.
 - Cada servidor de banco de dados deve usar uma faixa de valores que não se sobreponham. (Exemplo: Matriz com ID entre 1 e 999999, Filial 1 com ID entre 1000001 e 1999999, Filial 2 com ID entre 2000001 e 2999999, etc.
 - Ou então mude a chave primária para incluir um ID do servidor / filial evitando assim possibilidade de conflitos de INSERTs.
 - Se tiver tabelas com chaves primárias formadas por dados informados pelo cliente.
 - Neste caso na grande maioria das vezes não fazemos tratamento nenhum porque a probabilidade de conflito é pequena. Exemplo: Cadastro de pessoa física com o CPF como chave primária, Cadastro de usuários com o "login name" como chave primária.





- Quais os pré-requisitos a nível de sistema ?
 - Se tiver tabelas com chaves primárias formadas por numeradores ou códigos gerados pelo sistema. Exemplo: Números de pedido formado pela concatenação de ano+sequência, número de NF, etc.
 - Mudar a chave primária para incluir um ID do servidor / filial evitando assim possibilidade de conflito de chave primária em operações de INSERT.





- Quais os pré-requisitos a nível de sistema ?
 - Analisar as triggers existentes na aplicação e fazer os ajustes necessários
 - Se a trigger faz INSERT/UPDATE/DELETE em outra tabela:
 - Se esta tabela que recebe os INSERTs/UPDATEs/DELETEs for "manuseada" apenas pela trigger então a sugestão é NÃO replicar a tabela. Se tiver que replicar a tabela a qual a trigger efetua operações então deve criar uma maneira de evitarmos a "duplicidade" da replicação. Nossa sugestão é colocar uma condição na trigger para que ela seja executada apenas se a atualização tiver sido originado no servidor local.
 - **Dica:** Cada replicador pode oferecer uma forma de contornar isso, no OBJECTMMRS isso é feito através da comparação da variável USER que nos retorna o usuário que está efetuando a operação. Se USER for usuário usado pelo replicador ignoramos as operações feitas pela trigger.





- Quais os pré-requisitos a nível de sistema ?
 - Analisar as triggers existentes na aplicação e fazer os ajustes necessários
 - Se a trigger faz algum tipo de "controle de saldo", por exemplo em um sistema de controle de estoque o saldo em estoque de um produto, ou em um sistema contábil o saldo de uma conta (neste caso mais complexo ainda porque pode cascatear a atualização até chegar na conta nível 1).
 - No caso do estoque é relativamente simples se a chave primária da tabela de saldos de estoque contiver a chave do produto mais a chave do local/filial. Com isso teremos o saldo do produto em cada local da empresa. Se este é o caso o que temos de fazer é só recalcular o saldo de um produto se a movimentação ocorrida (entrada/saída) foi feita para o local onde está a base de dados. Na prática vamos colocar IFs para checar a chave do local/filial com a chave que estiver em alguma tabela de parâmetro do sistema indicando qual é o local/filial daquela base.
 - Se o estoque não for controlado por local, e tivermos apenas um saldo total, então neste caso apenas o servidor central deverá replicar esta tabela de saldos, os servidores locais podem recalcular mas o resultado não será replicado ao servidor central.





- Quais os pré-requisitos a nível de sistema ?
 - Analisar as triggers existentes na aplicação e fazer os ajustes necessários
 - No caso de trigger totalizadora de forma total (sem ser por local/filial), como no exemplo da conta contábil, neste caso novamente apenas o servidor central deverá replicar esta tabela de saldos, os servidores locais podem recalcular mas o resultado não será replicado ao servidor central.



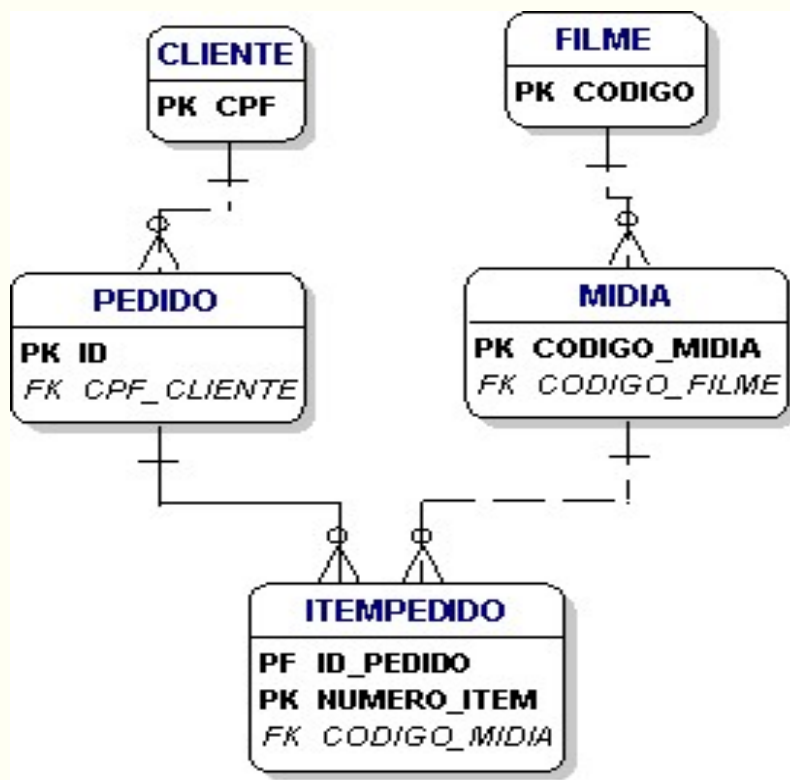


- Preparando um sistema para trabalhar descentralizado
 - Sistema original: Controle de Videolocadora. Sistema desenvolvido para a gestão de uma pequena videolocadora, funciona perfeitamente e atende às funcionalidades necessárias (desenvolvido para trabalho centralizado)
 - Modelo de dados: Cliente, Pedido, Item, Filme, Mídia
 - Algumas funcionalidades
 - Cadastro de clientes
 - Cadastro de filmes e mídias dos filmes (DVDs,fitas,etc)
 - Consulta estoque de filmes
 - Venda = Registro da locação de filmes
 - Registro da devolução de filmes





- Preparando um sistema para trabalhar descentralizado
 - Sistema original: Controle de Videolocadora. Sistema desenvolvido para a gestão de uma pequena videolocadora, funciona perfeitamente e atende às funcionalidades necessárias (desenvolvido para trabalho centralizado)



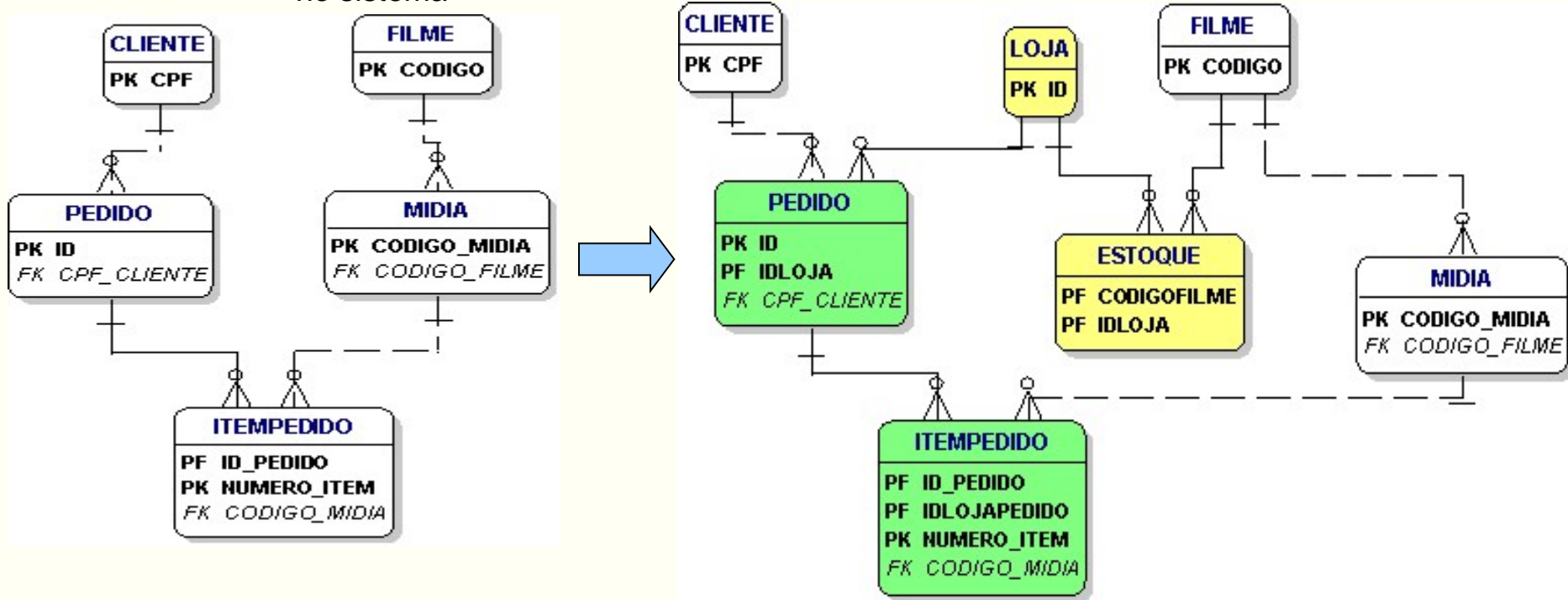


- Preparando um sistema para trabalhar descentralizado
 - Problema: Proprietário da Videolocadora abriu outra loja de locação de filmes e quer usar o mesmo sistema, mas quer compartilhar os clientes cadastrados entre as 2 lojas e ter uma visão do estoque das 2 lojas, assim como faturamento, etc.
 - Primeira opção: A ideal mas que pode ocasionar uma quantidade grande de alterações no sistema:
 - Modelo de dados: Cliente, Pedido, Item, Filme, Mídia, Loja, Estoque
 - O sistema continuaria com as mesmas funcionalidades mas em todas as consultas de saldos, faturamento, etc haveria a possibilidade do usuário informar a Loja ou Todas, e nas entradas de dados ele informaria ou o sistema através de algum parâmetro gravaria o ID da loja.
 - Cadastro de clientes
 - Cadastro de filmes e mídias dos filmes (DVDs,fitas,etc)
 - Consulta estoque de filmes
 - Venda = Registro da locação de filmes
 - Registro da devolução de filmes





- Preparando um sistema para trabalhar descentralizado
 - Problema: Proprietário da Videolocadora abriu outra loja de locação de filmes e quer usar o mesmo sistema, mas quer compartilhar os clientes cadastrados entre as 2 lojas e ter uma visão do estoque das 2 lojas, assim como faturamento, etc.
 - Primeira opção: A ideal mas que pode ocasionar uma quantidade grande de alterações no sistema



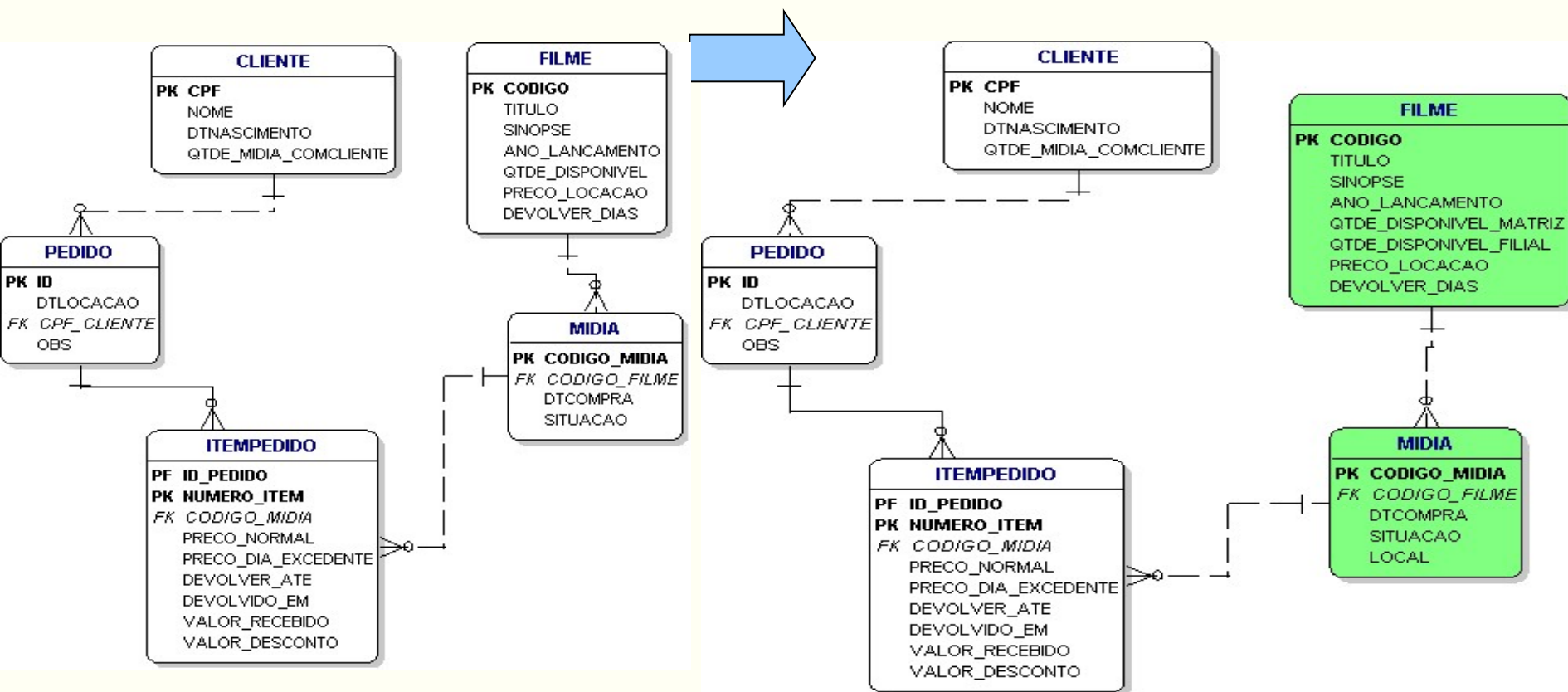


- Preparando um sistema para trabalhar descentralizado
 - Problema: Proprietário da Videolocadora abriu outra loja de locação de filmes e quer usar o mesmo sistema, mas quer compartilhar os clientes cadastrados entre as 2 lojas e ter uma visão do estoque das 2 lojas, assim como faturamento, etc.
 - Segunda opção: Não tão boa do ponto de vista do usuário mas a mais viável do ponto de vista de poder ser feita com o mínimo de alterações no sistema:
 - Modelo de dados: Cliente, Pedido, Item, Filme(*), Mídia(*)
 - O sistema continuaria com mesmas funcionalidades, apenas Filme sofreria a adição de colunas a mais de saldo de cada local e Mídia para indicar a qual local ela pertence.
 - Cadastro de clientes
 - Cadastro de filmes e mídias dos filmes (DVDs,fitas,etc)
 - Consulta estoque de filmes
 - Venda = Registro da locação de filmes
 - Registro da devolução de filmes
 - Compra = Registro da entrada de mídias de filmes





- Preparando um sistema para trabalhar descentralizado
 - Problema: Proprietário da Videolocadora abriu outra loja de locação de filmes e quer usar o mesmo sistema, mas quer compartilhar os clientes cadastrados entre as 2 lojas e ter uma visão do estoque das 2 lojas, assim como faturamento, etc.
 - Segunda opção: Não tão boa do ponto de vista do usuário mas a mais viável do ponto de vista de poder ser feita com o mínimo de alterações no sistema:





- Vamos mostrar na prática como fazer a Segunda Opção, que é a que requer a **menor** alteração na aplicação
 - Antes de começar, backup da base principal
 - Ajuste no modelo de dados
 - Inclusão das colunas QTDE_DISPONIVEL_MATRIZ e QTDE_DISPONIVEL_FILIAL em FILME
 - Inclusão da coluna LOCAL em MIDIA
 - Ajuste nas triggers ITEM_PEDIDO_AI0, ITEM_PEDIDO_AU0, MIDIA_AI0, MIDIA_AU0
 - Cópia da base de dados para o servidor "filial"
 - Ajuste da sequence / generator GEN_PEDIDO_ID na base filial
 - Instalação e configuração do software de replicação de banco de dados OBJECTMMRS
 - Testes





- Instalação do OBJECTMMRS para replicar na direção Matriz para Filial
 - OBS.: Estaremos simulando uma base remota da Filial, fisicamente estará no mesmo notebook da base Matriz (o processo de instalação é praticamente o mesmo, com a diferença que estaremos usando "localhost" como hostname do servidor FB tanto para MATRIZ como para FILIAL e em uma instalação real colocaríamos o IP ou algum DDNS como hostname da base da FILIAL.
 - Base Matriz: c:\fdd2011\bases\FDD2011_MATRIZ.FDB
 - Base Filial: c:\fdd2011\bases\FDD2011_FILIAL.FDB
 - OBS.1: Vídeo com instruções de como instalar pode ser visto em <http://www.object.com.br/files/Film03/InstalacaoBasica.html>
 - OBS.2: Manual do usuário do OBJECTMMRS em <http://www.object.com.br/wiki/ObjectMMRS>





- Instalação do OBJECTMMRS para replicar na direção Filial para Matriz
 - OBS.: Como estamos simulando 2 servidores na mesma máquina (notebook), faremos 2 instalações do software OBJECTMMRS em diretórios distintos.
 - Replicador da Base Matriz: c:\FDD2011\Matriz
 - Replicador da Base Filial: c:\FDD2011\Filial





- Testes
 - Inclusão de cliente CARLOS CANTU na base MATRIZ
 - Inclusão de cliente PAULO SHIBATA na base FILIAL
 - Inclusão de pedido na base MATRIZ
 - Inclusão de pedido na base FILIAL
 - Consultas para demonstrar o estado final das bases





- Questões ?
- Comentários ?
- Contato:
 - <http://www.object.com.br>
 - wagner@object.com.br

Agradeço a atenção de todos e estou à disposição para questões envolvendo projetos de replicação de banco de dados

Parabéns novamente ao grande evento FDD !!!

