

Projeto Lógico, Projeto Físico e Particionamento de Tabelas, Normalização de Dados

Neste momento o projetista descreve COMO DESENVOLVER os mecanismos de cada função e o relacionamento entre as funções. Nesta fase especificam-se integralmente os itens a serem construídos, as características, funcionalidades, regras de negócios e restrições. Neste ponto do desenvolvimento é possível ter uma visão final do projeto, embora ainda no papel. Esta visão deve ser validada junto ao cliente.

São Atividades/Sub Fases do Projeto Lógico

Revisar análise do sistema atual

Complementar e refinar.

Elaborar macro propostas

Levantar alternativas de macro solução, escolher a solução oportuna e revisar o planejamento.

Detalhar a lógica

Refinar requisitos funcionais reais das informações, elaborar os diagramas, descrever a lógica dos processos, elaborar dicionário de dados, esboçar documentos de entrada e saída, analisar as integrações com outros sistemas.

Definir estratégia de projeto físico

Definir tecnologia de base, rever impactos, elaborar planos logísticos, infra-estrutura e contingência, determinar planejamento e rever equipe, analisar custos, benefícios e viabilidades.

Aprovar o projeto lógico

Avaliar a qualidade, organizar informações, revisar fases anteriores, elaborar parecer e termo de compromisso, reunir e apresentar.

Mudanças em requisitos ocorrem em todas as fases do desenvolvimento do sistema.

As alterações nos requisitos são inevitáveis e não implicam em práticas pobres no processo de construção do sistema.

Embora a mudança seja inevitável, é usual o caso em que alguns requisitos são mais estáveis que outros.

Requisitos Estáveis

São concebidos com a essência de um sistema e seu domínio da aplicação, e mudam mais lentamente que requisitos voláteis.

Requisitos Voláteis

São específicos para a instanciação de um sistema em um ambiente particular e para um cliente particular e podem ser classificados em quatro tipos:

Requisitos Mutáveis

São os requisitos que mudam em função de mudanças no ambiente no qual o sistema opera. Por exemplo, os requisitos para um sistema que calcula taxas de dedução que evolui conforme as leis de taxação mudam.

Requisitos Emergentes

São os requisitos que não podem ser completamente definidos quando o sistema é especificado, mas que emergem quando o sistema está projetado e implementado. Por exemplo, pode não ser possível especificar de antemão os detalhes de como a informação será exibida. Conforme os stakeholders

veem exemplos de apresentações possíveis, eles podem pensar em novas maneiras de exibição da informação que seria útil para eles.

Requisitos Consequentes

São os requisitos baseados em suposições de como o sistema será utilizado. Quando o sistema é posto em uso, algumas destas suposições podem estar erradas. Usuários irão adaptar-se ao sistema e encontrar novas maneiras de usar suas funcionalidades, o que irá resultar em demandas dos usuários para mudanças no sistema.

Requisitos de Compatibilidade

São os requisitos que dependem de outro equipamento ou processo. Conforme muda esse equipamento, os requisitos também mudam.

Validação de Requisitos

Os produtos de trabalho criados como consequência da engenharia de requisitos (uma especificação dos requisitos do sistema e informações relacionadas) devem ser validados quanto à qualidade durante o passo de validação de requisitos. Esta validação examina a especificação para garantir que todos os requisitos do sistema foram estruturados de maneira não ambígua, que as inconsistências, omissões e erros foram apagados e corrigidos, e que os produtos de trabalho estão em conformidade com os padrões estabelecidos para o processo, para o projeto e para o produto.

O mecanismo primário de validação de requisitos é a revisão técnica formal. O time de revisão inclui os engenheiros de sistema, clientes, usuários e outros stakeholders que examinam a especificação do sistema à procura de erros de conteúdo ou interpretação, pontos onde pode ser necessário esclarecimento, perda de informações, inconsistências (um dos maiores problemas da engenharia de grandes produtos), requisitos conflitantes, ou requisitos irreais (de desenvolvimento impossível).

Embora a revisão para validação dos requisitos possa ser conduzida de qualquer maneira desde que possibilite a descoberta de erros nos requisitos

Projeto Físico

É a implementação física do projeto no ambiente empresarial. Execução ou realização do projeto de sistemas de informação. Contempla a confecção de programas (em software e manuais) e seus respectivos controles de testes e qualidade, bem como o layout final das entradas e saídas.

Subfases:

Especificar o modelo de dados: reestruturar os dados, eliminar redundâncias, analisar as dependências funcionais;

Especificar o modelo de processos: identificar e reestruturar os processos (tarefas) do sistema;

Definir arquitetura e plano de segurança:

Definir os arquivos físicos e métodos de acesso do software.

Definir procedimentos do plano de segurança das informações (Backup).

Construir o sistema de informação: analisar as linguagens de programação e executar o sistema ou programar o software.

Finalizar o sistema de informação: testes dos módulos, dos programas e arquivar os resultados. Aprovar o projeto físico.

Projeto de Software Lógico x Físico

- Projeto lógico (independente de implementação) é executado para produzir um projeto que poderia ser implementado em diferentes plataformas (hardware, linguagem de programação, SGBD)

- Projeto físico (implementação específica) é executado para produzir um projeto que é específico para a plataforma escolhida

- Algumas vezes se a plataforma é conhecida quando começa o projeto, não haverá o estágio de projeto lógico.

Entendemos que cada empresa possui necessidades e características específicas, por isso disponibilizamos o que há de mais moderno em tecnologia da informação e consultores especializados para desenvolver de forma ágil sistemas customizados pra cada cliente.

Contamos com uma equipe de desenvolvimento de sistemas altamente qualificada em todas as fases de um projeto. As ferramentas de desenvolvimento, bem como os sistemas elaborados, podem ser vistos em nossos escritórios e/ou nas instalações de nossos clientes.

Nosso portfólio de projetos de grande porte em diversos segmentos gerou um forte conhecimento de regras de negócio.

Desenvolvimento de Sistemas

Os projetos são iniciados através de "workshops", onde são utilizadas técnicas de JRP - Joint Requirements Planning e JAD - Joint Application Design, atingindo rapidamente os resultados esperados. Os "workshops" utilizam ainda, técnicas de modelagem de dados, análise essencial (eventos) e projeto orientado a objeto, compondo a base da Metodologia de Desenvolvimento de Projetos.

Desenvolvimento de Projetos Internet / Intranet

Elaboração de projetos de Intranet para a empresa e/ou departamentos.

Elaboração de projetos de comércio eletrônico: business to business, business to consumer.

Desenvolvimento de sites.

Alguns Projetos

Sistema Budget

Sistema GPS - Controle e gerenciamentos da solução de problemas

Sistema de Abertura de Ficha

Sistema Sales Force Automation (WEB/Mobile)

Sistema Portal do Distribuidor

Sistema Plano de Negócios

Sistema ROTULUS

Sistema Concentrador Fiscal

Metodologia de Gerenciamento de Projetos

A metodologia consiste em uma simplificação do PMBok, visando tornar o processo mais objetivo e Eficiente. As etapas componentes da metodologia são:

Elementos na Implantação

Tecnologia: Recursos tecnológicos

Pessoas: Trabalho em equipe

Objetivos: Claros e realistas

A metodologia consiste em uma simplificação do PMBok, visando tornar o processo mais objetivo e eficiente. As etapas componentes da metodologia são:

Levantamento

Planejamento do Projeto

Preparação do Projeto

Treinamento e Validação de Funcionalidades

Simulações

Disponibilização e Acompanhamento

Projeto Físico

No desenvolvimento do projeto físico, o protótipo e a especificação são transformados em código de programação. Após a transformação, são feitos testes para se avaliar a qualidade das aplicações geradas nesta fase. Para cada item construído deve ser realizada uma bateria de testes com objetivo de identificar não-conformidades face às especificações (Regras de Negócios e Funcionalidades) e face aos padrões correspondentes às tecnologias utilizadas.

O objetivo principal desta fase é, ao seu final, garantir que todos os itens foram construídos conforme definidos pelo analista de sistemas, estando a aplicação apta a ser homologada pelo cliente. O sucesso desta fase está diretamente ligado à validação correta da fase anterior. Ao final desta fase, o cliente já terá a primeira versão do sistema.

São Atividades/Sub Fases do Projeto Lógico:

Revisar projeto lógico

Complementar e refinar.

Especificar o modelo de dados

Finalizar dicionário de dados, elaborar o modelo de dados, normalizar depósitos de dados.

Definir arquitetura

Definir arquivos físicos e métodos de acesso, definir procedimentos de segurança.

Construir o sistema

Finalizar as entradas e saídas do sistema, executar o sistema ou programar o software, construir programas paralelos.

Finalizar o sistema

Elaborar testes dos programas, definir fluxos e procedimentos operacionais, complementar a documentação.

Definir estratégia de projeto de implantação

Esboçar projeto de implantação, planejar treinamento, elaborar plano de conversão.

Aprovar o projeto físico

Avaliar a qualidade, organizar informações, revisar fases anteriores, elaborar parecer e termo de compromisso, reunir e apresentar.

Você pode criar uma tabela ou um índice particionado no SQL Server 2017 usando o SQL Server Management Studio ou o Transact-SQL. Os dados das tabelas e dos índices particionados são divididos horizontalmente em unidades que podem ser disseminadas por mais de um grupo de arquivos em um banco de dados. O particionamento pode tornar as tabelas e os índices grandes mais gerenciáveis e escalonáveis.

A criação de uma tabela ou um índice particionado ocorre normalmente em quatro partes:

Crie um grupo ou grupos de arquivos e os arquivos correspondentes que terão as partições especificadas pelo esquema de partição.

Cria uma função de partição que mapeia as linhas de uma tabela ou de um índice em partições com base nos valores de uma coluna especificada.

Cria um esquema de partição que mapeia as partições de uma tabela particionada ou índice para os novos grupos de arquivos.

Crie ou modifique uma tabela ou um índice e especifique o esquema de partição como local de armazenamento.

Antes de Começar

Limitações e Restrições

O escopo de uma função e de um esquema de partição é limitado ao banco de dados em que foram criados. No banco de dados, as funções de partição residem em um namespace separado das outras funções.

Se alguma linha dentro de uma função de partição tiver colunas de particionamento com valores nulos, essas linhas serão alocadas para a partição à extrema esquerda. No entanto, se NULL for especificado como valor de limite e RIGHT for indicado, a partição extrema esquerda permanecerá vazia e os valores NULL serão colocados na segunda partição.

Segurança

Permissões

A criação de uma tabela particionada requer a permissão CREATE TABLE no banco de dados e a permissão ALTER no esquema no qual a tabela está sendo criada. A criação de um índice particionado requer a permissão ALTER na tabela ou exibição onde o índice está sendo criado. A criação de uma tabela ou um índice particionado requer uma das permissões adicionais a seguir:

Permissão ALTER ANY DATASPACE. Essa permissão tem como padrão os membros da função de servidor fixa sysadmin e das funções de banco de dados fixas db_owner e db_ddladmin .

Permissão CONTROL ou ALTER no banco de dados no qual a função e o esquema de partição estão sendo criados.

Permissão CONTROL SERVER ou ALTER ANY DATABASE no servidor do banco de dados no qual a função e o esquema de partição estão sendo criados.

Usando o SQL Server Management Studio

Siga as etapas deste procedimento para criar um ou mais grupos de arquivos, arquivos correspondentes e uma tabela. Você fará referência a esses objetos no próximo procedimento quando criar a tabela particionada.

No Pesquisador de Objetos, clique com o botão direito do mouse no banco de dados no qual você deseja criar uma tabela particionada e selecione Propriedades.

Na caixa de diálogo Propriedades do Banco de dados – database_name, em Selecionar uma página, selecione Grupos de Arquivos.

Continue adicionando linhas até que você tenha criado todos os grupos de arquivos para a tabela particionada.

Clique em OK.

Em Selecione uma página, selecione Arquivos.

Em Linhas, clique em Adicionar. Na nova linha, digite um nome de arquivo e selecione um grupo de arquivos.

Continue adicionando linhas até que tenha criado ao menos um arquivo para cada grupo de arquivos.

Expanda a pasta Tabelas e crie uma tabela como você faz normalmente. Para obter mais informações, veja Criar tabelas (Mecanismo de Banco de Dados). Opcionalmente, você pode especificar uma tabela existente no próximo procedimento.

Para criar uma tabela particionada

Clique com o botão direito do mouse na tabela que você deseja particionar, aponte para Armazenamento e clique em Criar Partição....

No Assistente para Criar Partição, na página Bem-vindo ao Assistente para Criar Partição, clique em Avançar.

Na página Selecione uma Coluna de Particionamento, na grade Colunas de particionamento disponíveis, selecione a coluna em que deseja particionar sua tabela. Somente colunas com tipos de dados que possam ser usados para particionar dados serão exibidas na grade Colunas de particionamento disponíveis. Se você selecionar uma coluna computada como a coluna de particionamento, a coluna deverá ser designada como persistente.

As opções que você tem para as colunas de particionamento e o intervalo de valores são determinadas, basicamente, pela extensão na qual seus dados podem ser agrupados de forma lógica. Por exemplo, você pode optar por dividir seus dados em agrupamentos lógicos por meses ou trimestres de um ano. As consultas que você planeja fazer nos seus dados determinarão se o agrupamento lógico é adequado para o gerenciamento de suas partições de tabela. Todos os tipos de dados são válidos para uso como colunas de particionamento, exceto text, ntext, image, xml, timestamp, varchar(max), nvarchar(max), varbinary(max), tipos de dados de alias ou tipos de dados CLR definidos pelo usuário.

As opções adicionais a seguir estão disponíveis nessa página:

Colocar essa tabela na tabela particionada selecionada. Permite selecionar uma tabela particionada que contenha dados relacionados para unir a essa tabela na coluna de particionamento. Tabelas com partições unidas nas colunas de particionamento costumam ser consultadas de forma mais eficiente.

Alinhamento de armazenamento de índices não exclusivos e índices exclusivos com uma coluna de partição indexada

Alinha todos os índices da tabela que são particionados com o mesmo esquema de partição. Quando uma tabela e seus índices são alinhados, você pode mover partições para dentro e fora de tabelas particionadas com mais eficiência, pois seus dados são particionados com o mesmo algoritmo.

Depois de selecionar a coluna de particionamento e as outras opções, clique em Avançar.

Na página Selecione uma Função de Partição, em Selecione uma função de partição, clique em Nova função de partição ou em Função de partição existente. Se você escolher Nova função de partição, digite o nome da função.

Se você escolher Função de partição existente, selecione o nome da função que você gostaria de usar na lista. A opção Função de partição existente não estará disponível se não houver outra função de partição no banco de dados.

Na página Selecione um Esquema de Partição, em Selecione um esquema de partição, clique em Novo esquema de partição ou em Esquema de partição existente.

Se você escolher Novo esquema de partição, digite o nome do esquema. Se você escolher Esquema de partição existente, selecione o nome do esquema que você gostaria de usar na lista. A opção Esquema de partição existente não estará disponível se não houver outro esquema de partição no banco de dados.

Na página Mapear Partições, em Intervalo, selecione Limite esquerdo ou Limite direito para especificar se é para incluir o maior ou o menor valor de limite dentro de cada grupo de arquivos criado. Você deve sempre inserir um grupo de arquivos extra, além do número de grupos de arquivos especificado para os valores de limite durante a criação de partições.

Na grade Selecione os grupos de arquivos e especifique os valores de limite, em Grupo de arquivos, selecione um grupo de arquivos no qual deseja particionar seus dados. Em Limite, digite o valor de limite para cada grupo de arquivos. Se o valor de limite ficar em branco, a função de partição mapeará a tabela inteira ou o índice em uma única partição usando o nome da função de partição.

As opções adicionais a seguir estão disponíveis nessa página:

Definir Limites...

Abre a caixa de diálogo Definir Valores de Limite para selecionar os valores de limite e os intervalos de datas que deseja para suas partições. Essa opção estará disponível somente quando você tiver selecionado uma coluna de particionamento que contenha um dos seguintes tipos de dados: date, datetime, smalldatetime, datetime2 ou datetimeoffset.

Estimar armazenamento

Calcula o número de linhas, o espaço necessário e o espaço disponível para armazenamento para cada grupo de arquivos especificado para as partições. Esses valores são exibidos na grade como valores somente leitura.

A caixa de diálogo Definir Valores de Limite permite as opções adicionais a seguir:

Data de Início

Seleciona a data inicial para obter os valores de intervalo de suas partições.

Data de Término

Seleciona a data de término para obter os valores de intervalo de suas partições. Se você selecionar Limite esquerdo na página Mapear Partições, essa data será o último valor de cada grupo de arquivos/partição. Se você selecionar Limite direito na página Mapear Partições, essa data será o primeiro valor do próximo ao último grupo de arquivos.

Intervalo de Datas

Seleciona a granularidade de data ou incremento de valor de intervalo desejado para cada partição.

Depois de concluir essa página, clique em Avançar.

Na página Selecione uma Opção de Saída, especifique como você deseja preencher sua tabela particionada. Selecione Criar Script para criar um script SQL baseado nas páginas anteriores no assistente.

Selecione Executar imediatamente para criar a nova tabela particionada depois de concluir todas as páginas restantes no assistente. Selecione Agenda para criar uma nova tabela particionada em um momento predeterminado no futuro.

Se você selecionar Criar script, as opções a seguir estarão disponíveis em Opções de script:

Script para Arquivo

Gera o script como um arquivo. sql. Digite um nome de arquivo e o local na caixa Nome do arquivo ou clique em Procurar para abrir a caixa de diálogo Local do Arquivo de Script. Em Salvar Como, selecione Texto Unicode ou Texto ANSI.

Script para Área de Transferência

Salva o script na área de transferência.

Script para Nova Janela de Consulta

Gera o script para uma nova janela do Editor de Consultas. Essa é a seleção padrão.

Se você selecionar Agenda, clique em Alterar agenda.

Na caixa de diálogo Nova Agenda de Trabalho, na caixa Nome, digite o nome da agenda de trabalho.

Na lista Tipo de Agenda, selecione o tipo de agenda:

Iniciar automaticamente quando o SQL Server Agent for iniciado

Iniciar sempre que as CPUs estiverem ociosas

Recorrente. Selecione essa opção se sua nova tabela particionada for atualizada com novas informações regularmente.

Uma vez. Essa é a seleção padrão.

Marque ou desmarque a caixa de seleção Habilitado para habilitar ou desabilitar a agenda.

Se você selecionar Recorrente:

Em Frequência, na lista Ocorre, especifique a frequência de ocorrência:

Se você selecionar Diário, na caixa Ocorre periodicamente a cada, digite a frequência com que a agenda de trabalho se repete em dias.

Se você selecionar Semanal, na caixa Ocorre periodicamente a cada, digite a frequência com que a agenda de trabalho se repete em semanas. Selecione o dia ou os dias da semana em que a agenda de trabalho é executada.

Se você selecionar Mensalmente, selecione Dia ou O.

Se você selecionar Dia, digite o dia do mês que você deseja que a agenda de trabalho seja executada e a frequência com que a agenda de trabalho se repete em meses. Por exemplo, se você desejar que a agenda de trabalho seja executada no 15º dia do mês a cada dois meses, selecione Dia e digite "15" na primeira caixa e "2" na segunda caixa. Observe que o maior número permitido na segunda caixa é "99".

Se você selecionar O, selecione o dia específico da semana no mês que você deseja que a agenda de trabalho seja executada e a frequência com que a agenda de trabalho se repete em meses. Por exemplo, se você desejar que a agenda de trabalho seja executada no último dia da semana do mês a cada dois meses, selecione Dia, selecione último na primeira lista e dia da semana na segunda lista e depois digite "2" na última caixa. Você também pode selecionar primeiro, segundo, terceiro ou quarto, bem como dias específicos da semana (por exemplo: domingo ou quarta-feira) nas primeiras duas listas. Observe que o maior número permitido na última caixa é "99".

Em Frequência diária, especifique a frequência com que a agenda de trabalho se repete no dia da execução da agenda de trabalho:

Se você selecionar Ocorre uma vez às, digite a hora específica do dia em que a agenda de trabalho deve ser executada na caixa Ocorre uma vez às. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Se você selecionar Ocorre a cada, especifique a frequência com que a agenda de trabalho é executada durante o dia escolhido em Frequência. Por exemplo, se você desejar que o agendamento de trabalho se repita a cada 2 horas durante o dia em que é executado, selecione Ocorre a cada, digite "2" na primeira caixa e selecione hora(s) na lista. Nessa lista, você pode selecionar também minuto(s) e segundo(s). Observe que o maior número permitido na primeira caixa é "100".

Na caixa Iniciando às, digite a hora em que a agenda de trabalho deve começar a ser executada. Na caixa Terminando às, digite a hora em que a agenda de trabalho deve parar de se repetir. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Em Duração, em Data de início, digite a data que você deseja que a agenda de trabalho inicie a execução. Selecione Data de término ou Nenhuma data de término para indicar quando a execução da agenda de trabalho deve parar. Se você selecionar Data de término, digite a data em que você deseja que a execução da agenda de trabalho pare.

Se você selecionar Uma Vez, em Ocorrência única, na caixa Data, insira a data em que o agendamento de trabalho será executado. Na caixa Hora, digite a hora em que a agenda de trabalho será executada. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Em Resumo, em Descrição, verifique se todas as configurações da agenda de trabalho estão corretas.

Clique em OK.

Depois de concluir essa página, clique em Avançar.

Na página Resumo da Revisão, em Examinar as seleções, expanda todas as opções disponíveis para verificar se todas as configurações de partição estão corretas. Se tudo estiver como esperado, clique em Concluir.

Na página Progresso do Assistente para Criar Partição, monitore as informações de status das ações do Assistente para Criar Partição. Dependendo das opções selecionadas no assistente, a página de progresso pode conter uma ou várias ações. A caixa superior exibe o status geral do assistente e o número de mensagens de status, erro e aviso que ele recebeu.

As opções a seguir estão disponíveis na página Progresso do Assistente para Criar Partição :

Detalhes

Fornece a ação, status e qualquer mensagem retornada pela ação executada pelo assistente.

Ação

Especifica o tipo e o nome de cada ação.

Status

Indica se a ação do assistente retornou como um todo o valor de Êxito ou de Falha.

Mensagem

Fornece qualquer mensagem de aviso ou erro retornada pelo processo.

Relatório

Cria um relatório contendo os resultados do Assistente para Criar Partição. As opções são Exibir Relatório, Salvar Relatório no Arquivo, Copiar Relatório na Área de Transferência e Enviar Relatório como Email.

Exibir Relatório

Abre a caixa de diálogo Exibir Relatório, que contém um relatório de texto do progresso do Assistente para Criar Partições.

Salvar Relatório no Arquivo

Abre a caixa de diálogo Salvar Relatório Como .

Copiar Relatório na Área de Transferência

Copia os resultados do relatório de progresso do assistente na área de transferência.

Enviar Relatório como Email

Copia os resultados do relatório de progresso do assistente em uma mensagem de email.

Quando terminar, clique em Fechar.

O Assistente para Criar Partição cria a função e o esquema de partição e aplica o particionamento à tabela especificada. Para verificar o particionamento da tabela, no Pesquisador de Objetos, clique com o botão direito do mouse na tabela e selecione Propriedades. Clique na página Armazenamento. A página exibe informações como o nome da função e do esquema de partição e o número de partições.

Usando Transact-SQL

Para criar uma tabela particionada

No Pesquisador de Objetos, conecte-se a uma instância do Mecanismo de Banco de Dados.

Na barra Padrão, clique em Nova Consulta.

Copie e cole o exemplo a seguir na janela de consulta e clique em Executar. O exemplo cria novos grupos de arquivos, uma função e um esquema de partição. Uma nova tabela é criada com o esquema de partição especificado no local de armazenamento

O uso de particionamento de tabelas se torna útil em grandes ambientes de banco de dados, onde existem a necessidade de grande acesso aos dados, não somente para esta causa, mas também para inúmeras funcionalidades dentro do banco, como: backup, rápido acesso a dados e dados históricos.

Para que seja necessário implementar particionamento de tabelas é aconselhável ter conhecimento sobre filegroups

O que é?

O particionamento de tabelas (Oracle Partitioning) é a capacidade de dividir fisicamente tabelas de seu banco dados, sem a necessidade de reescrever o código fonte de sua aplicação, pois logicamente a tabela particionada é como uma tabela normal de seu banco de dados e é capaz de receber as mesmas instruções SQL (DDL, DML e DCL) que qualquer outra tabela, a separação dos dados acontece somente no âmbito físico, de acordo com o que for especificado no momento do particionamento pelo desenvolvedor.

Para que Serve?

O particionamento é de grande utilidade quando estamos trabalhando com tabelas que tem uma massa de dados muito grande, pois a partir do momento que a tabela está particionada, os dados estão divididos fisicamente e na hora da busca das informações, um comando SELECT, por exemplo, você sempre irá buscar as informações em uma porção fisicamente menor.

Outro ponto está relacionado à segurança, se você perder partição por alguma falha (seja ela qual for) você pode fazer o restore e recover apenas da partição danificada sem afetar as demais diminuindo então o tempo de restauração e recuperação considerando também que os dados das outras partições ficam disponíveis para a aplicação durante todo esse processo, outra vantagem que se pode ter com relação à performance de seu banco de dados, é você poder colocar as partições em discos distintos colocando os dados mais acessados em discos de maior velocidade (para que seus usuários tenham uma boa performance) e os dados históricos em discos de menor velocidade visto que esse tipo de informação não é comumente acessada. Com o passar do tempo, as informações que vão se tornando “velhas” você pode mover dos discos de alta performance para os discos de menor performance sem haja a necessidade de parar a aplicação (isso faz parte das atividades do DBA).

Como Se Faz?

Para particionar uma tabela, podemos fazer isso em dois momentos, primeiro no momento de criação da tabela, ou então em um segundo momento, é quando a tabela já está criada e você mesmo assim

ainda pode particioná-la utilizando uma package do Oracle chamada DBMS_REDEFINITION (em breve escreverei um artigo sobre este tema), mas agora vamos ver como particionar uma tabela do “zero”. Neste artigo vamos considerar que teremos uma tabela que armazenará informações sobre notas fiscais (É uma tabela bem básica, apenas para entendermos o objetivo deste artigo) abaixo especificação da tabela.

Como Funciona

O particionamento descrito neste artigo é aquele que afeta individualmente tabelas e/ou índices, também conhecido como particionamento horizontal de tabelas. É um processo para segregar dados de tabelas específicas em várias unidades físicas menores.

Ele é diferente da segmentação de dados em datafiles diferentes, pois neste caso não temos controle de quais datafiles serão usados para armazenar dados específicos. O particionamento nos permite definir regras para que os dados sejam gravados em filegroups diferentes, com base nos valores de um ou mais campos da tabela.

Este recurso é bastante útil, mas não deve ser confundido com o sharding, comum em bancos que funcionam na nuvem, como o próprio AZURE SQL. Sharding envolve a criação de uma infraestrutura para que o banco todo funcione de forma “horizontal” e divida a carga de processamento entre máquinas diferentes.

Benefícios

Particionamento permite, por exemplo:

Melhorar performance de consultas que acessem dados de uma única partição;

Melhorar a performance da importação de grandes volumes de dados;

Reduzir tempo de execução de tarefas administrativas como reindexação, backup e restore, que são executadas na partição ao invés da tabela toda;

Facilitar operações de arquivamento de dados.

Uma vez definidas as regras de particionamento de cada tabela, a administração das partições é um processo simples. Portanto, é uma opção interessante a se considerar quando trabalhar com tabelas muito grandes (eu diria da ordem de 100Gb ou mais).

Criando Partições

Vejamos um exemplo prático. Temos uma tabela com dados históricos que vem apresentando problemas, particularmente em razão do seu tamanho.

A tabela contém histórico de transações de muitos anos, mas a grande maioria das pesquisas envolvendo esta tabela trata das transações do ano corrente. Mesmo assim, espera-se que esta tabela continue a ser usada pelo menos nos próximos 10 anos.

Tendo isso em mente, decidiu-se criar um particionamento da tabela com base na data da transação. Implementaremos estas ações do modo mais didático possível, desconsiderando o fato de que este processo eventualmente envolveria um tempo maior de parada da aplicação.

Em primeiro lugar, vamos renomear a tabela existente, visto que a tabela particionada que criaremos precisa ter o mesmo nome da tabela antiga, assim não será necessária nenhuma alteração na aplicação.

A normalização de um banco de dados é o tema do artigo de hoje. Este é um tema super importante para a criação do banco de dados, vamos colocarmos a mão-na-massa assim como fizemos no primeiro artigo de Modelagem de Dados e vamos continuar a passar os fundamentos da linguagem e os conceitos básicos de como funciona um banco de dados.

O Oracle é um SGBD relacional e isso quer dizer que ele aplica as regras definidas por Edgar Frank Codd, ele foi quem desenvolveu o modelo de banco de dados relacional. Ao todo são 12 regras, porém vou passar à vocês apenas as 3 primeiras que são as essenciais para o seu dia-a-dia.

Normalização Banco de Dados

Para começarmos a normalização é necessário recapitularmos o modelo da primeira parte da aula de modelagem, abaixo vocês vão ver o modelo relacional de entidades que fizemos na semana passada vamos seguir a ideia de sempre darmos exemplos em cima da teoria

Criando Tabelas na Normalização de Dados

Toda entidade vira uma tabela;

Então seguindo esta regra teremos as seguintes tabelas:

TB_PRODUTO

TB_COMANDA

TB_ESTOQUE

TB_CLIENTE

TB_CAIXA

Se você clicar no nome da entidade aparecerá o nome da tabela que vai ser criada.

Relacionamentos que possuem atributos viram tabelas (existe a possibilidade em relacionamentos 1:n dos atributos irem para uma das tabelas, ao invés de se criar uma nova. Entretanto, relacionamentos com atributos são mais comuns em relacionamentos n:n, gerando assim uma nova tabela);

Agora vamos criar as tabelas desses relacionamentos.

TB_PRODUTO_COMANDA

TB_PRODUTO_ESTOQUE

TB_CAIXA_ESTOQUE

TB_CLIENTE_PAGAMENTO

Toda tabela possui um ou mais campos que são os campos únicos, onde cada entidade se diferencia, por exemplo, um cliente possui um CPF único que pode ser o que diferencia todos os clientes, estes campos únicos são chamados de chaves primárias.

Criando as Chaves Primárias na Normalização de Dados

Abaixo seguem as chaves primárias de todas as tabelas criadas.

ID_PRODUTO

ID_COMANDA, DT_INICIO, DT_FIM

ID_ESTOQUE

ID_CLIENTE (Que nesse caso vai ser o CPF)

ID_PAGAMENTO

ID_PRODUTO, ID_COMANDA, DT_INICIO, DT_FIM

ID_PRODUTO, ID_ESTOQUE

ID_PAGAMENTO, ID_ESTOQUE

ID_CLIENTE, ID_PAGAMENTO

Criando Chaves Estrangeiras na Normalização de Dados

Relacionamentos são representados por chaves estrangeiras (ou Foreign Key – atributo correspondente à chave primária de outra relação, base para a integridade referencial);

Temos as tabelas que possuem chaves estrangeiras que compõem os relacionamentos das tabelas do nosso banco de dados.

ID_PRODUTO, ID_COMANDA

ID_PRODUTO, ID_ESTOQUE

ID_PAGAMENTO, ID_ESTOQUE

ID_CLIENTE, ID_PAGAMENTO

Perceberam algo bem interessante, as chaves estrangeiras são os mesmos campos que formam as chaves primárias compostas dos relacionamentos, então tentem sempre achar esta conexão entre os relacionamentos, chaves estrangeiras e chaves primárias.

Criando Relacionamentos na Normalização

Relacionamentos 1:1 podem ser mapeados numa única tabela (quando possuem a mesma chave primária), em duas tabelas (quando as chaves primárias são diferentes e um dos lados do relacionamento é obrigatório) ou em três tabelas (quando o relacionamento é opcional em ambos os lados)

No nosso caso existe apenas um relacionamento 1:1 na nossa modelagem que é o relacionamento da entidade Comanda x Cliente, porque um cliente pode apenas ter uma comanda para efetuar compras e uma comanda pode pertencer apenas a um cliente enquanto ela estiver ativada. Por esse motivo temos esse relacionamento 1:1.

Mas elas possuem chaves primárias diferentes então por esse motivo temos duas tabelas, porém com a obrigatoriedade do ID_CLIENTE (Chave Primária da TB_CLIENTE, na TB_COMANDA).

Relacionamentos 1:n são mapeados de forma que a chave primária do lado “1” seja representada do lado “n” como chave estrangeira;

Relacionamentos n:n devem ser transformados em dois relacionamentos 1:n, resultando numa nova tabela;

Estes dois últimos passos ficarão mais legais nos desenhos que vou mostrar para vocês logo abaixo.

Ferramenta para Normalização de Dados

Após passar essas regras começamos a desenhar o nosso banco de dados, bom como nesse caso ficaria muito trabalhoso usar o Paint para desenhar tudo na mão, então eu vou utilizar o MySQL Workbench que é uma ferramenta utilizada para modelar o MySQL, que é um banco de dados muito utilizado para aplicações web e que foi comprado pela Oracle e melhor de tudo, esta ferramenta é totalmente grátis.

Formas Normais

Um pouco mais de teoria com as 3 formas normais de Codd, que vou apresentar a vocês.

1ª Forma Normal (1FN)

Toda relação deve ter uma chave primária e deve-se garantir que todo atributo seja atômico. Atributos compostos devem ser separados.

Por exemplo, um atributo Endereço deve ser subdividido em seus componentes: Logradouro, Número, Complemento, Bairro, Cidade, Estado e CEP. Além disso, atributos multivalorados devem ser discriminados separadamente ou separados em uma outra relação.

Por exemplo, um atributo multivalorado Telefones poderia ser separado em Telefone Residencial, Telefone

Comercial e Telefone Celular ou, ainda, ser convertido em outra relação que pudesse representar um número indeterminado de telefones.

Já passei para vocês como transformar um Modelo de Relacionamentos de Entidades em um modelo de relacionamento de tabelas. E se alguém ficar com alguma dúvida por favor comentem para que eu possa ajudá-los a resolvê-las e claro fazer com que o meu curso evolua sanando as principais dúvidas de todos.

Para exemplificar vou pegar um dos casos de relacionamento, quando criamos a tabela TB_PRODUTO_COMANDA, que contém o Identificador da Comanda e o Identificador do Produto estamos transformando um relacionamento de n:n entre o produto e a comanda em um relacionamento 1:n, pois podem existir vários produtos, mas na tabela TB_PRODUTO_COMANDA vai existir apenas um ID_PRODUTO e campo quantidade para fazer os cálculos na hora da compra e o mesmo serve para a Comanda, onde podem existir várias comandas, mas apenas um identificador de comanda poderá estar atrelado a tabela TB_PRODUTO_COMANDA.

2ª Forma Normal (2FN)

Toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária. Como exemplo, uma relação que contenha os atributos Código da Obra, Código do Fornecedor, Nome do Fornecedor e Preço de Venda, Considerando que a chave primária é composta pelos atributos Código da Obra e Código do Fornecedor, não está na Segunda Forma Normal, uma vez que o Nome do Fornecedor depende apenas do Código do Fornecedor, e não do Código da Obra. Uma nova relação (Fornecedor) deve ser criada contendo os campos Código do Fornecedor (como chave) e Nome do Fornecedor. Na relação original, ficariam os atributos Código da Obra e o Código do Fornecedor, ambos formando a chave primária composta, e o atributo Preço de Venda. Além disso, o atributo Código do Fornecedor também seria uma chave estrangeira para a nova relação criada. Esta forma normal ajuda a diminuir redundâncias de informações criadas indevidamente.

Essa regra é bem interessante, porque eliminamos informações duplicados e conseguimos conservar a integridade das informações, por exemplo, na tabela TB_PRODUTO colocamos o nome do tipo do produto, bom vamos dizer que o produto seja uma bebida, e que um funcionário cadastrou o banco de dados Bebida, mas veio outro funcionário e ao cadastrar o produto cadastrou Bebidas, isso faria com que tivéssemos dois tipos de produto cadastrados no banco, que na verdade seriam os mesmos, por esse motivo é necessário criar essas tabelas.

3ª Forma Normal (3FN)

Toda relação deve estar na 2FN e devem-se eliminar dependências funcionais transitivas, ou seja, todo atributo não chave deve ser mutuamente independente. Como exemplo, uma relação que contenha os atributos Matrícula do Funcionário (atributo chave), Nome do Funcionário, Código do Departamento e Nome do Departamento não está na Terceira Forma Normal.

O Nome do Departamento é dependente do Código do Departamento, e não da Matrícula do Funcionário. Uma mudança no nome do departamento, por exemplo, levaria a modificações em todos os funcionários daquele departamento.

Para eliminar este problema, cria-se uma nova relação (Departamento) contendo Código do Departamento e Nome do Departamento. Na relação original, retira-se o Nome de Departamento, mantendo-se o Código do Departamento, agora como chave estrangeira. Esta forma normal também ajuda a diminuir redundâncias e aumentar a independência das relações.

