iSeries

Integração comercial utilizando o DB2 OLAP no iSeries

Relatório da **Experiência**

iSeries

Integração comercial utilizando o DB2 OLAP no iSeries

Índice

Capítulo 1. Introdução ao iCola 1
O que é o OLAP 1
Descrição geral do cenário do iCola. 2

Capítulo 2. Repositório de dados 5
Descrição geral da aplicação 5
Pontos de concepção da aplicação 7
Instalação da aplicação 10

Capítulo 3. Processo ETL 11
Descrição geral da aplicação 12
Pontos de concepção da aplicação 13
Instalação da aplicação 13
Observações chave 16

Capítulo 4. Cubo multidimensional. . . 19
Descrição geral da aplicação 20
Pontos de concepção da aplicação 22
Instalação da aplicação 24
Observações chave 33

Capítulo 5. Analyzer 35
Descrição geral da aplicação 36
Pontos de concepção da aplicação 38
Instalação da aplicação 39
Observações chave 41

Capítulo 6. Ideias para o futuro 43

Capítulo 7. Apêndice A 45

Capítulo 8. Apêndice B 47

Capítulo 9. Apêndice C 51

Capítulo 10. Exclusões 53

Capítulo 11. Referências 55

Capítulo 1. Introdução ao iCola

O cenário do iCola é um cenário de teste desenvolvido pela equipa de teste de soluções para clientes da IBM(R). A equipa de teste de soluções para clientes concebe, implementa, avalia e fornece cenários adaptados ao cliente de um modo semelhante ao utilizado por arquitectos de tecnologias de informação na indústria. O cenário do iCola testa a funcionalidade de base de dados, bem como produtos de Business Intelligence. Em particular, o actual cenário do iCola centra-se no IBM DB2(R) OLAP Server(TM). De futuro, o iCola pretende incorporar a funcionalidade de Business Intelligence de base de dados adicional e possivelmente mais ferramentas. Poderá encontrar tópicos adicionais relacionados com ideias futuras na secção Ideias para o futuro deste documento.

Esta secção fornece uma descrição geral de OLAP (Online Analytical Processing) e detalhes do cenário do iCola.

Esta secção está dividida em duas subsecções:

- O que é o OLAP
- Descrição geral do cenário do iCola

O que é o OLAP

O OLAP (Online Analytical Processing) autoriza os utilizadores a colocarem questões intuitivas e complexas ad hoc sobre os seus negócios, como "Qual o lucro para o terceiro trimestre, na região sudeste, de determinados produtos principais?" Esta questão requer várias perspectivas dos dados, como época, regiões e produtos. Todas estas perspectivas são denominadas *dimensões*.

Os dados relacionais podem ser considerados bidimensionais porque cada parte dos dados (um facto) está relacionada com uma linha e uma coluna (uma dimensão). As dimensões numa base de dados multidimensional são perspectivas mais elevadas dos dados que representam os componentes principais de um plano comercial, como Accounts, Time, Products e Markets. Numa aplicação OLAP, estas dimensões raramente são alteradas.

Cada dimensão tem componentes individuais denominados *membros*. Por exemplo, os trimestres do ano são membros da dimensão Time e os produtos individuais são membros da dimensão Products. Poderão existir hierarquias de membros nas dimensões, como meses dentro dos trimestres da dimensão Time. Os membros têm tendência para alterar com frequência, por exemplo, à medida que o negócio evolui, são adicionados novos produtos e clientes.

O OLAP beneficiou das tecnologias de informação dominantes e hoje é um componente essencial das soluções de Business Intelligence em todas as indústrias. Os pedidos cada vez mais frequentes para análise de grandes quantidades de dados e a autorização concedida a um número cada vez maior de empregados na empresa para a tomada de decisões comerciais correctas estão a exceder os limites das soluções OLAP.

O IBM(R) DB2 OLAP Server(TM) é um produto OLAP que pode ser utilizado para criar um intervalo amplo de aplicações de planeamento multidimensional, análise e comunicação. O IBM DB2 OLAP Server envia aplicações analíticas para uma análise multidimensional rápida, intuitiva, permitindo aos utilizadores colocarem questões numa linguagem comercial intuitiva. O OLAP Server processa pedidos multidimensionais que calculam, consolidam e obtêm informações de uma base de dados multidimensional, uma base de dados relacional ou ambas.

O DB2 OLAP Server é baseado na tecnologia OLAP desenvolvida pela Hyperion Solutions Corporation. As referências a Hyperion Essbase e Hyperion Integration Server existem na interface e em toda a documentação.

O DB2 OLAP Server inclui todas as funções de Hyperion Essbase. Além disso, oferece a opção de armazenamento de bases de dados multidimensionais como conjuntos de tabelas relacionais. Independentemente da opção de gestão de armazenamento escolhida, o gestor de aplicações Essbase e os comandos de Essbase podem ser utilizados para criar uma aplicação Essbase e respectivas bases de dados associadas. Além disso, existem mais de 70 ferramentas Essbase disponíveis fornecidas por fabricantes de software independentes que conseguem aceder a bases de dados multidimensionais de forma transparente.

Descrição geral do cenário do iCola

O iCola é uma empresa de bebidas mediana fictícia que funciona como uma intermediária entre fabricantes de bebidas e fornecedores de clientes. O iCola exerceu actividades comerciais durante alguns anos e recolheu dados sobre as vendas e os clientes; no entanto, a gestão não conseguiu utilizar estes dados para tomar decisões comerciais uma vez que não estava disponível um mecanismo para analisar os dados recolhidos. A gestão na empresa iCola verificou que os concorrentes utilizavam dados para tomar decisões de marketing e vendas, melhorando a qualidade do serviço, bem como aumentando as receitas com base na análise dos dados.

Depois de sessões de planeamento da gestão, a empresa iCola decidiu beneficiar das vantagens de Business Intelligence. Business intelligence é o conceito de criação de bases de dados informativas a partir de dados operacionais, utilizando esses dados para análise e tomada de decisões. Os dados são consolidados a partir de várias origens num único servidor de base de dados e são "transformados" como parte do processo de consolidação. Após a consolidação, são utilizadas ferramentas para analisar os dados. Algumas das ferramentas de análise são: Decision Support Systems (DSS), Executive Information Systems (EIS), Multidimensional Analysis Tools (MDA), Online Analytical Processing (OLAP) e Data Mining Tools. O iCola decidiu iniciar o projecto de Business Intelligence criando um repositório de dados e utilizando a solução IBM(R)'s DB2(R) OLAP para análise.

A empresa iCola possui hardware iSeries(TM), pSeries(R) e xSeries(R). Os sistemas operacionais Current(R) são executados no iSeries e as operações de comunicação são executadas nos servidores pSeries e xSeries. Durante a primeira fase de implementação da solução de Business Intelligence, foi criado um repositório de dados num sistema iSeries. Além disso, o servidor DB2 OLAP foi colocado num segundo sistema iSeries e o DB2 OLAP Analyzer foi instalado num servidor Windows(R) xSeries.

As figuras 1 e 2 ilustram os sistemas e software principais do cenário do iCola:

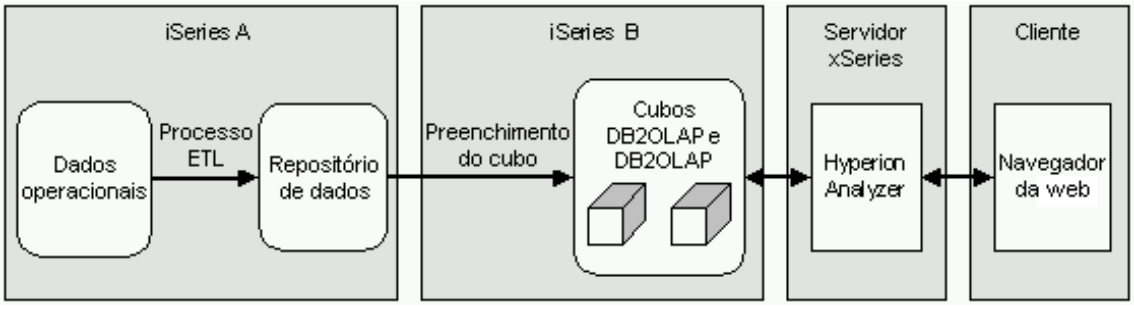

Figura 1: descrição geral do cenário do iCola

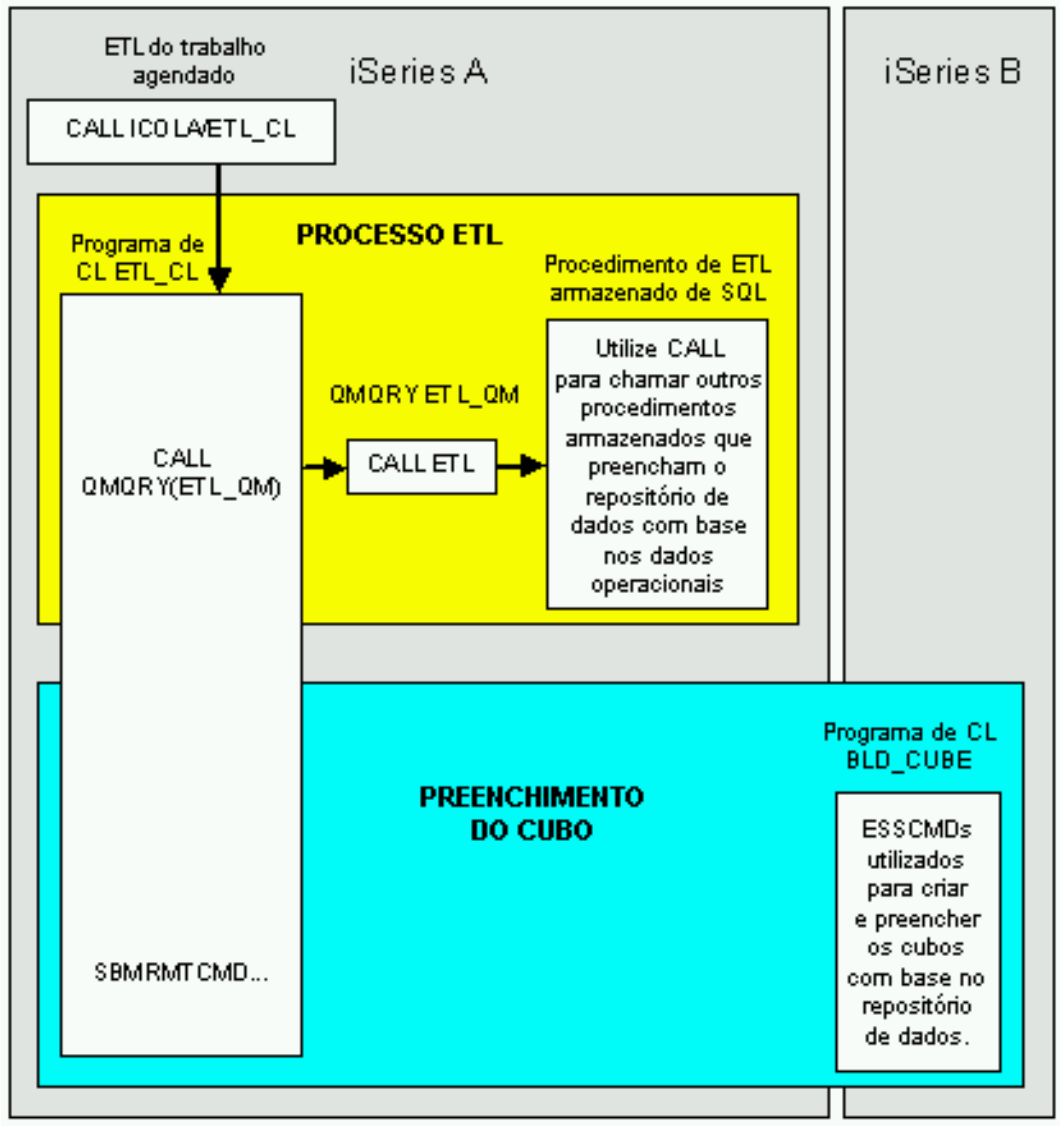

Figura 2: processo ETL e preenchimento do cubo em detalhe

O iSeries A pode ser utilizado no OS/400(R) V5R2 ou V5R3. No entanto, o iSeries B tem de ser utilizado no OS/400 V5R2. Uma vez que a versão mais recente do iSeries em execução no IBM DB2 OLAP Server(TM) 8.1 é a OS/400 V5R2, o iSeries B utiliza a versão OS/400 V5R2.

Os restantes capítulos apresentam os sistemas e software principais em detalhe:

- O repositório de dados contém dados do histórico para o cenário do iCola
- O processo ETL é o veículo que move os dados operacionais para o repositório de dados
- Os dados são então preenchidos num cubo multidimensional, onde são armazenados num formato que pode ser utilizado pelos analistas comerciais
- Por fim, o iCola utiliza o DB2 OLAP Server Analyzer para examinar os dados e criar relatórios

Capítulo 2. Repositório de dados

Esta secção apresenta as características principais da instalação do repositório de dados da DB2 Universal Database(TM) for iSeries(TM) para o cenário do iCola juntamente com o conceito de armazenamento de dados.

Os dados que são recolhidos das transacções comerciais diárias são conhecidos como dados operacionais. Estes dados podem conter informações úteis para os analistas comerciais. Por exemplo, os analistas podem utilizar as informações sobre os produtos que são vendidos, em que regiões e época do ano para verificar a existência de anomalias ou planear futuras vendas. No entanto, caso os analistas acedam directamente aos dados operacionais poderão ocorrer vários problemas:

- Os analistas poderão não ter conhecimentos para consultar a base de dados operacional. Por exemplo, a consulta de bases de dados do sistema de gestão de informações requer uma aplicação que utilize uma linguagem de manipulação de dados especializada. De um modo geral, os programadores com conhecimentos de consulta de bases de dados operacionais têm que assegurar permanentemente a manutenção da base de dados e respectivas aplicações, sem analisar o respectivo conteúdo.
- O desempenho é crítico para muitas bases de dados operacionais, como as bases de dados de um banco. O sistema não consegue gerir as consultas ad hoc efectuadas pelos analistas.
- Os dados operacionais geralmente não se encontram no formato mais adequado para que possam ser utilizados por analistas comerciais. Por exemplo, os dados de vendas resumidos por produto, região e época são muito mais úteis para os analistas do que os dados não processados. Do mesmo modo, ferramentas como o DB2(R) OLAP podem criar mais facilmente cubos de dados com um repositório de dados do que com os dados operacionais.
- As informações podem ser distribuídas para fora da organização. Os analistas necessitam de aceder aos dados centralizados em vez de a origens diferentes de dados.

O armazenamento de dados resolve estes problemas. O armazenamento de dados consiste na concepção e implementação de processos, ferramentas e serviços para gerir e distribuir informações completas, atempadas, precisas e legíveis para a tomada de decisões. Com o armazenamento de dados, são criados arquivos de dados informativos. Estes dados são extraídos dos dados operacionais e, em seguida, convertidos num formato que possa ser utilizado pelos analistas comerciais. Um repositório de dados é uma versão mais pequena do armazém de dados e é utilizado no cenário do iCola. Existem várias ferramentas de armazenamento que podem copiar todos os dados de vendas a partir da base de dados operacional, efectuar cálculos para resumir os dados e escrever os dados resumidos numa base de dados independente a partir dos dados operacionais. Assim, os analistas podem consultar a base de dados independente (o repositório de dados) sem causar qualquer impacto no desempenho das bases de dados operacionais.

Descrição geral da aplicação

A empresa iCola tem uma base de dados operacional para operações diárias e um repositório de dados para análise. A base de dados operacional e o repositório de dados estão localizados na DB2 Universal Database(TM) for iSeries(TM).

As tarefas mais comuns requeridas para a instalação de um repositório de dados incluem:

- Exploração dos dados operacionais para definir as tabelas e campos a incluir no repositório de dados. Estes são os campos que mais interessam aos analistas comerciais.
- Definição de um modelo de esquema em estrela para os dados existentes no repositório de dados. Um esquema em estrela é uma estrutura especializada que consiste em tabelas de várias dimensões, que

descrevem aspectos de um negócio (exemplos de dimensões incluem clientes, produtos e tempo) e uma tabela de factos, que agrega campos das tabelas de dimensões. A tabela de factos normalmente contém também os volumes de vendas.

Um excelente recurso para a criação de armazéns de dados e repositórios de dados é o IBM(R) DB2 Universal Database Business Intelligence Tutorial, que pode encontrar em http://www.ibm.com/software/data/db2/db2olap/docs/V71docs/db2tu/frame3.htm#db2tussw.

Base de dados operacional

A base de dados operacional contém dados das transacções diárias da empresa iCola.

Exemplos destes dados incluem:

- Produtos vendidos pela iCola
- Clientes que compram à iCola
- Fornecedores a quem a iCola compra os produtos
- Tipos de transporte utilizados pela iCola

Os dados operacionais da iCola existiam muito antes da iCola decidir melhorar a empresa com Business Intelligence. O repositório de dados foi adicionado ao cenário sem causar qualquer impacto nos dados operacionais nem nas transacções da empresa.

Repositório de dados

Os dados operacionais geralmente não se encontram no formato mais adequado para as ferramentas de análise comercial. O formato mais adequado para os dados é o de um esquema em estrela. A principal diferença entre os dados operacionais e o esquema em estrela do iCola consiste nas relações simplificadas entre as tabelas de base de dados. O esquema em estrela está configurado de um modo que os dados que não são úteis para os analistas não são incluídos no repositório de dados. Os dados que são úteis são dispostos numa estrutura hierárquica.

Pontos de concepção da aplicação

Concepção do repositório de dados

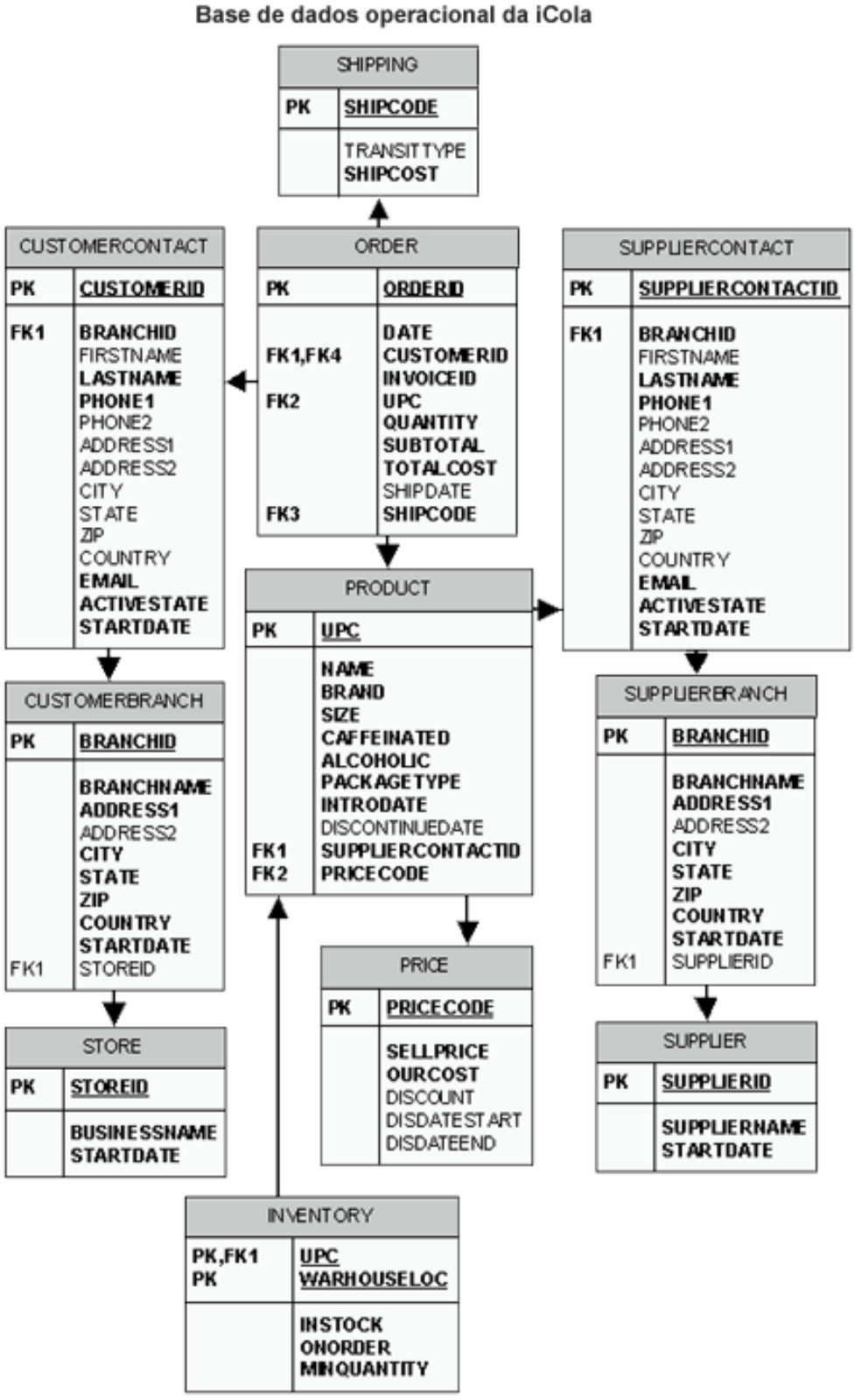

Figura 3. Base de dados operacional do iCola

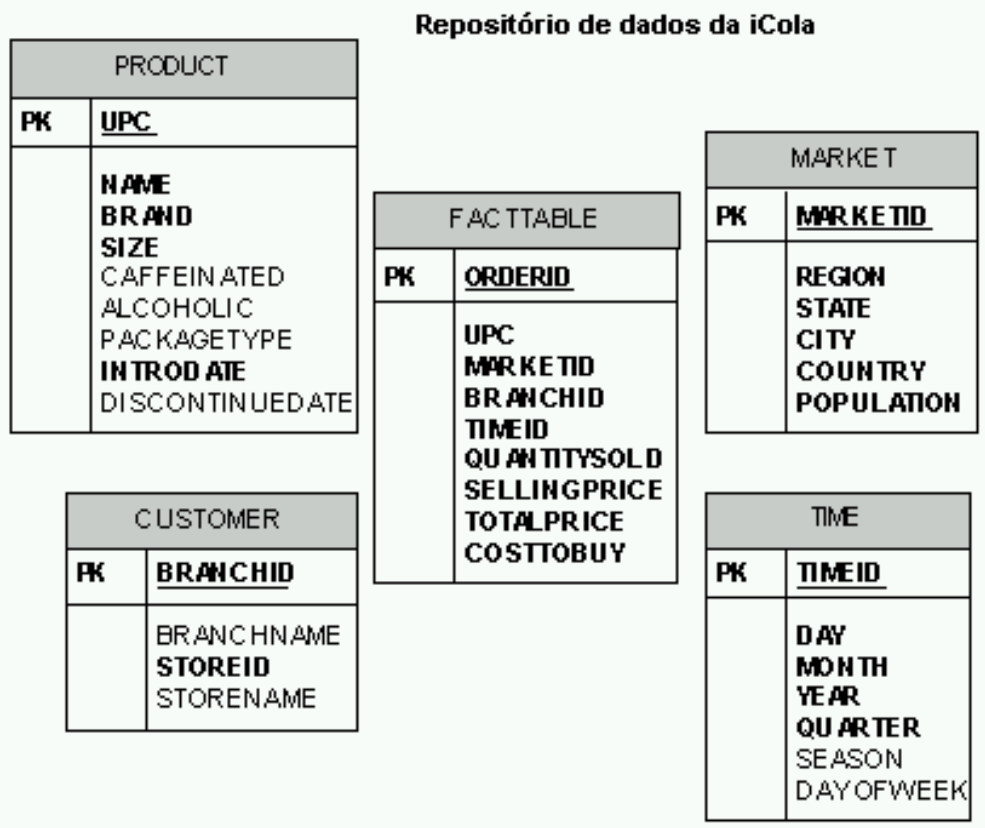

Figura 4. Repositório de dados do iCola

A concepção do repositório de dados começa com a análise de dados operacionais, a procura de relações entre os dados e a determinação das informações mais interessantes para os utilizadores.

O guia de iniciação *Designing the Star Schema* refere a importância da determinação de medidas e dimensões. As medidas são valores numéricos mensuráveis e cumulativos. Exemplos de medidas incluem:

- Dados de vendas - gerados por todas as encomendas
  - Valor de custo das mercadorias vendidas (COGS)
  - Valor das vendas
- O número de clientes do iCola num dia específico
- A média de produtos comprados num dia específico.

As dimensões afectam o modo de visualização das medidas. Por exemplo, quando observar o volume de vendas, existe uma dimensão temporal envolvida que indica quando foram efectuadas as vendas. Também é útil para saber as vendas por produto ou por cliente.

Foram considerados os seguintes elementos na concepção do repositório de dados do iCola:

- Medidas
  - Custos das mercadorias vendidas (COGS)
  - Vendas (a quantidade de produto vendido)
- Dimensões

– Tempo
– Produto
– Mercado
– Cliente

A tabela de factos contém as medidas, ou factos, juntamente com as referências a cada dimensão no repositório de dados. As medidas não fornecem quaisquer informações úteis sem primeiro considerarem as dimensões.

A tabela de factos do iCola (denominada FACTTABLE) tem como base a tabela ORDER na base de dados do ICOLA. As linhas na tabela ORDER contêm medidas e dimensões. As dimensões têm uma referência temporal, localização e cliente associados à venda do produto.

A tabela que se segue apresenta o modo como as tabelas operacionais ORDER e FACTTABLE estão relacionadas no repositório de dados:

ORDER	**FACTTABLE**
ORDERID	ORDERID
DATE	TIMEID
CUSTOMERID	BRANCHID
CUSTOMERID	MARKETID
QUANTITY	QUANTITYSOLD
SUBTOTAL/QUANTITY	SELLINGPRICE
SUBTOTAL	TOTALPRICE
UPC	UPC
OURCOST (tabela PRICE)	COSTTOBUY

As tabelas de dimensões deverão ter uma chave principal para um campo individual. Esta chave é frequentemente uma coluna de identidade, composta por um número incrementável gerado automaticamente. Esta chave principal foi criada porque era necessário mapear elementos de dimensões com linhas na tabela FACTTABLE. Os outros campos na tabela de dimensões contêm descrições completas de dados relevantes para o iCola. Por exemplo, a dimensão PRODUCT é baseada na tabela PRODUCT da base de dados do ICOLA. Esta tabela contém campos com o nome do produto, a marca, o tamanho, o tipo de pacote, etc. Estes campos no repositório de dados não contêm códigos com ligação a outras tabelas diferentes de FACTTABLE.

A principal diferença entre a tabela de factos e as tabelas de dimensões está relacionada com o formato:

- De um modo geral, a tabela de factos contém muitos registos, mas não contém muitas colunas. Isto acontece porque os dados existentes na tabela de factos são dados relacionados com medidas ou chaves principais das tabelas de dimensões. Por conseguinte, o formato desta tabela é longo e estreito.
- De um modo geral, as tabelas de dimensões não contêm muitos registos, mas têm muitas colunas para apresentar os detalhes das dimensões, uma vez que são descrições completas das dimensões. Por conseguinte, o formato desta tabela é curto e largo.

Esta hierarquia de dimensões permite executar funções de “pesquisa detalhada” nos dados. Seguem-se exemplos de consultas que podem ser efectuadas:

- Uma consulta para calcular somas por marca.
- Uma consulta para calcular a quantidade de produtos comprados por cada mercado.
- Cálculo da soma dos produtos individuais de um mês específico.

O repositório de dados ICOLADM contém uma tabela de factos e quatro tabelas de dimensões.

- FACTTABLE - Esta é a tabela de factos que contém informações sobre o mercado, produto, cliente e tempo, juntamente com os dados COGS e de vendas. Esta tabela é preenchida principalmente a partir da tabela ICOLA.ORDER.
- PRODUCT - Esta é a tabela de dimensão que contém informações sobre os produtos. Todas as informações desta tabela são retiradas de ICOLA.PRODUCT.
- MARKET - Esta é a tabela de dimensão que contém informações sobre os mercados. Esta tabela é preenchida com base na tabela ICOLA.CUSTOMERBRANCH da base de dados operacional.
- CUSTOMER - Esta é a tabela de dimensão que contém informações sobre o cliente. Esta tabela é preenchida com base nas tabelas ICOLA.CUSTOMERBRANCH e ICOLA.STORE da base de dados operacional.
- TIME - Esta é a tabela de dimensão que contém a cronologia. Esta tabela é preenchida por um programa em Java(TM) simples que introduz as datas equivalentes a um ano.

Existem também várias outras tabelas de ajuda no ICOLADM:

- REGION - Esta tabela contém todos os 50 estados, juntamente com as regiões e países correspondentes. (Os países foram incluídos no caso de se verificar a expansão do iCola para mercados internacionais e ser necessário adicionar estados de diferentes países). Esta tabela é utilizada pelo processo ETL para preencher a tabela MARKET.
- ETLPROCESS - Esta tabela é utilizada para indicar o último início e conclusão do processo ETL. O processo ETL será explicado numa secção posterior.

Instalação da aplicação

Repositório de dados

O repositório de dados, ICOLADM pode ser criado executando um conjunto de instruções de SQL. Consulte o Apêndice A para obter detalhes sobre as instruções de SQL.

Capítulo 3. Processo ETL

Para mover os dados operacionais para o repositório de dados, os administradores utilizam um processo para extrair, transformar e carregar (ETL - Extract, Transform, and Load). Este processo ETL é normalmente executado num intervalo agendado, quando existe pouca actividade do cliente (o cenário do iCola executa o processo ETL diariamente à meia-noite). O processo ETL é vital para manter o repositório de dados preenchido com os dados mais recentes da base de dados operacional. Um processo ETL pode ser relativamente simples ou extremamente complexo, dependendo da quantidade de dados que é necessário mover e da quantidade de alterações que é necessário efectuar nos dados.

A seguir são apresentados os passos do processo ETL em detalhe:

1. Extrair - Os dados relevantes têm de ser seleccionados a partir dos dados operacionais e devem corresponder ao intervalo de tempo agendado para o processo ETL. Se o processo ETL for executado diariamente, a aplicação deverá seleccionar os dados das últimas 24 horas. Ao escolher os dados relevantes poupa tempo e trabalho futuros.

   Do mesmo modo, o repositório de dados poderá requerer mais dados do que os fornecidos pelos dados operacionais. Segue-se um exemplo no processo ETL do iCola em que são necessários dados adicionais:

   - O repositório de dados do iCola mantém um registo das informações regionais relativamente às encomendas do cliente. Se um cliente da Rochester, MN encomendar um produto, o iCola necessita de saber que o cliente está no Midwest. O repositório de dados contém uma tabela denominada REGIONS de todos os estados dos E.U.A. juntamente com as respectivas regiões. Durante o processo ETL, estes dados regionais são extraídos juntamente com os dados relativos à cidade e ao estado e armazenados no repositório de dados.
2. Transformar - Normalmente os dados seleccionados a partir dos dados operacionais não estão no mesmo formato que os dados que necessitam de ser introduzidos no repositório de dados. O processo ETL deverá processar quaisquer transformações necessárias.

   - Segue-se um exemplo do modo como o processo ETL do iCola transforma os dados:

     Na base de dados operacional, os dados de cada encomenda são armazenados no formato de "data" SQL. No entanto, no repositório de dados, a data de cada encomenda deverá estar no formato AAAAMMDD. Para transformar a data dos dados operacionais para a data do repositório de dados, é utilizada esta fórmula de SQL:

     ((YEAR(ICOLA.ORDER.DATE)*10000) + (MONTH(ICOLA.ORDER.DATE)*100) + DAY(ICOLA.ORDER.DATE))

   Muitas vezes, os dados operacionais têm de ser "depurados" examinando os campos que possam ter várias respostas válidas com referência à mesma entidade. Por exemplo, um cliente poderá introduzir a cidade "Mt. Horeb", WI e outro cliente introduzir a cidade "Mount Horeb", WI. O processo ETL deverá reconhecer que estas cidades são as mesmas. Neste cenário simples, a depuração dos dados não é efectuada devido à natureza dos dados introduzidos que se encontram num ambiente controlado. Contudo, a depuração dos dados deverá ser considerada quando avaliar o modo como os dados serão transformados.
3. Carregar - Depois da transformação dos dados, estes deverão ser carregados no repositório de dados. Este processo pode ser efectuado inserindo instruções de SQL.

Há produtos que podem ser utilizados pelos administradores para ajudar na implementação do processo ETL. Um destes produtos é o IBM(R) DB2(R) Information Integrator. Uma vez que o iCola é um cenário simples e requer a execução de um processo ETL no iSeries(TM), foi gerado um ETL processado adequado. De futuro, se a empresa expandir, o iCola poderá considerar a utilização do IBM DB2 Information Integrator para o processo ETL.

Nas secções que se seguem, será apresentado em detalhe o processo ETL relativo ao armazenamento do iCola.

Descrição geral da aplicação

A figura 5 ilustra os vários componentes do processo ETL do iCola. O processo ETL é apenas parte do processo agendado do iCola. Além disso, o processo de preenchimento do cubo é também agendado e é descrito na secção de preenchimento do cubo.

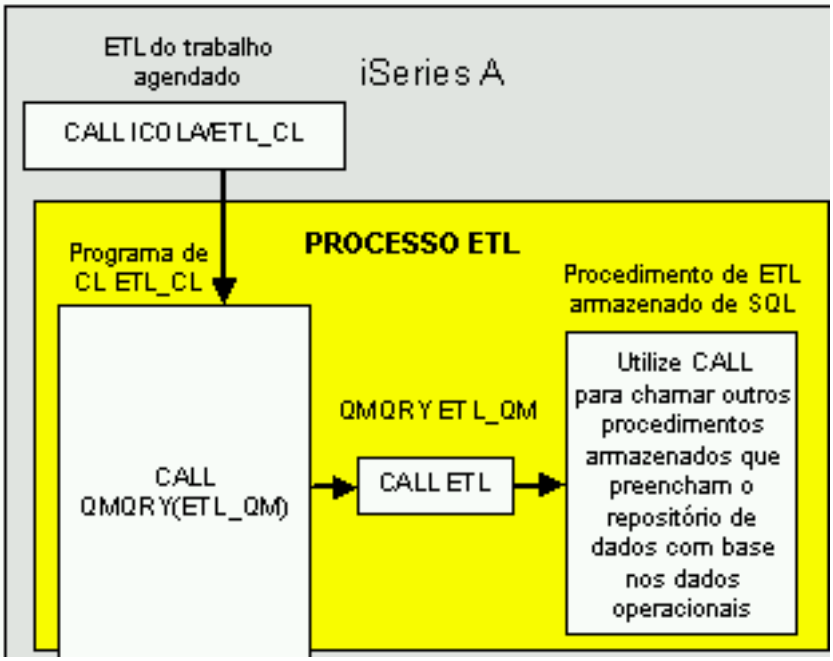

Figura 5: processo ETL do iCola (Repare que esta figura é uma parte da figura 2 na secção ″Descrição geral″)

ETL do trabalho agendado: utilizando o menu Work with Job Schedule Entries (Trabalhar com entradas do calendário de trabalhos) (WRKJOBSCDE), foi criado um trabalho agendado para activar o início automático do processo ETL, todos os dias à meia noite. A aplicação que é chamada pelo trabalho agendado é o programa ETL_CL de ETL.

Programa ETL_CL de linguagem de comando (CL): o ETL_CL tem duas funções. A parte de ETL deste programa chama a consulta do gestor de consultas ETL_QM. Após a conclusão da chamada de consulta do gestor de consultas, o ETL_CL envia também um comando para o sistema que contém o cubo (conhecido como iSeries(TM) B) para iniciar o preenchimento do cubo. O preenchimento do cubo é descrito na secção ″Cubo″.

Consulta do gestor de consultas ETL_QM: o ETL_QM chama o procedimento de SQL armazenado de ETL. O iCola utiliza uma consulta do Query Manager (Gestor de consultas) (QM) para chamar o procedimento de SQL armazenado porque os procedimentos de SQL armazenados não podem ser chamados directamente a partir de um programa da linha de comandos (CL). No entanto, os procedimentos de SQL armazenados podem ser chamados a partir de uma consulta de QM e esta pode ser chamada a partir de um programa de CL.

Procedimento de ETL armazenado de SQL: o procedimento de ETL armazenado chama outros procedimentos de SQL armazenados, que são descritos em detalhe nos pontos de concepção da aplicação:

- STARTETL

- CUSTOMER
- MARKET
- PRODUCT
- FACTTABLE
- ENDETL

Estes procedimentos de SQL armazenados consultam os dados operacionais e inserem dados no repositório de dados.

Pontos de concepção da aplicação

- Os procedimentos de SQL armazenados são a parte vital do processo ETL. A lista que se segue descreve a função de cada procedimento de SQL armazenado:
  - ETL: o principal procedimento armazenado que chama os outros procedimentos armazenados.
  - STARTETL: insere a hora de início do processo ETL na coluna START da tabela ETLPROCESS. Esta tabela é utilizada para guardar registos.
  - ENDETL: insere a hora de conclusão do processo ETL na coluna FINISH da tabela ETLPROCESS. Esta tabela é utilizada para guardar registos.
  - CUSTOMER: insere informações das tabelas operacionais CUSTOMER e CUSTOMERBRANCH na tabela CUSTOMER do repositório de dados.
  - MARKET: insere informações regionais das tabelas operacionais CUSTOMER e CUSTOMERBRANCH na tabela MARKET do repositório de dados.
  - PRODUCT: insere informações da tabela operacional PRODUCT na tabela PRODUCT do repositório de dados.
  - FACTTABLE: insere informações da tabela operacional ORDER na tabela FACTTABLE do repositório de dados.

  Se comparar os procedimentos armazenados mencionados acima com as tabelas que constituem o repositório de dados, verificará que falta uma tabela - a tabela TIME. Uma vez que o tempo é constante e previsível, é mais fácil preencher a tabela TIME com várias entradas de uma só vez, utilizando um programa em Java (TM) simples, do que criar um procedimento armazenado. A dificuldade ao criar um procedimento armazenado para preencher a tabela TIME é na gestão de situações em que não é possível executar o procedimento TIME no tempo programado. Esta situação pode acontecer, por exemplo, se estiver a ser efectuada uma gravação do sistema na altura de execução do processo ETL agendado.

Instalação da aplicação

O processo ETL para o cenário do iCola foi configurado executando estes passos:

1. Criar os procedimentos de SQL armazenados
2. Criar a consulta de QM
3. Criar o programa de CL
4. Criar o trabalho agendado
5. Certificar-se de que o perfil que está a executar o processo ETL tem as autoridades adequadas

Criar os procedimentos de SQL armazenados

O código utilizado para criar cada um dos procedimentos de SQL armazenados pode ser localizado no Apêndice B. Nos procedimentos de SQL, repare que os procedimentos FACTTABLE, CUSTOMER, MARKET e PRODUCT utilizam a palavra-chave "EXCEPT" no procedimento. A palavra-chave EXCEPT é nova na DB2 Universal Database(TM) for iSeries(TM) V5R3. Para ilustrar como funciona esta palavra-chave, consulte a figura 6:

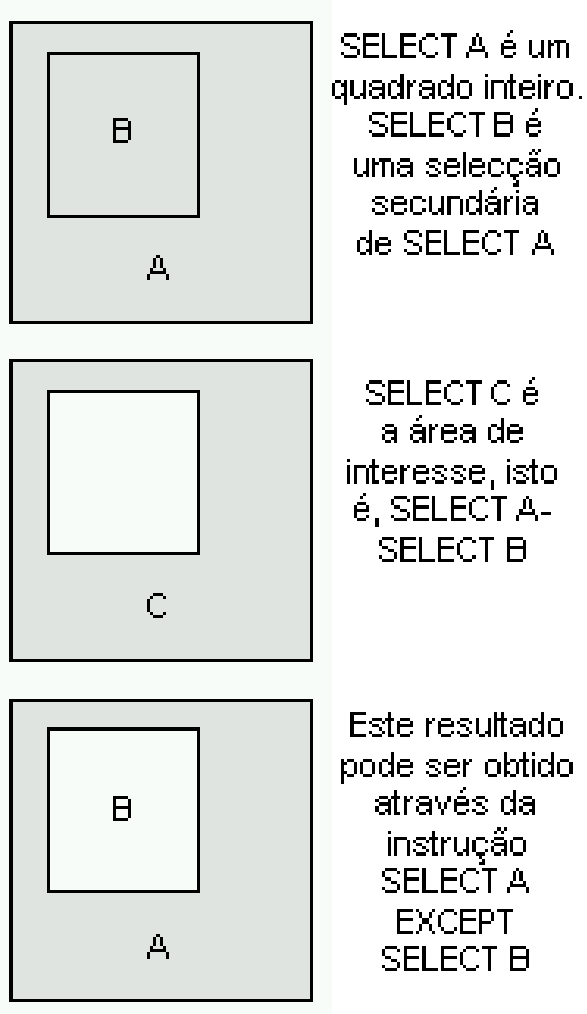

Figura 6: ilustração da palavra-chave EXCEPT

Se os resultados produzidos da execução de *SELECT C* forem os mesmos da execução de *SELECT A - SELECT B*, para localizar os resultados de *SELECT C*, deverá executar simplesmente: (*SELECT A*) EXCEPT (*SELECT B*). Do mesmo modo, para localizar os resultados de *SELECT B*, execute (*SELECT A*) EXCEPT (*SELECT C*).

A palavra-chave EXCEPT facilita a criação destes procedimentos de SQL armazenados. Uma vez que os dados operacionais são extensos, só os dados actualizados recentemente necessitam de ser introduzidos no repositório de dados, (ou seja, dados que ainda não existam). A palavra-chave EXCEPT pode ser utilizada para criar estes procedimentos.

As instruções utilizadas na criação de procedimentos de SQL estão localizadas no Apêndice B. O Apêndice B contém também instruções que podem ser utilizadas para criar os procedimentos de SQL no OS/400(R) V5R2 (uma vez que o V5R2 não suporta a palavra-chave EXCEPT).

Criar a consulta de QM

Para criar a consulta de QM de modo a chamar os procedimentos armazenados, siga estes passos:

1. Antes de criar a consulta, verifique se o perfil que está a criar e a chamar a consulta de QM tem autoridade para utilizar as palavras-chave de SQL adequadas. Como valor assumido, os perfis só

podem executar a instrução SELECT no QM. No cenário do iCola, é necessário executar a instrução CALL. Para conceder autoridade para utilizar outras palavras-chave de SQL, inicie sessão utilizando um perfil com autoridade *SECADM.

2. Inicie a interface de QM executando STRQM.
3. Para **Work with Query Manager Profiles** (Trabalhar com perfis do gestor de consultas), seleccione a opção 10.
4. Na lista apresentada seleccione o perfil adequado e escolha a opção 2 para alterar os atributos do perfil.
5. Avance para a parte inferior dos parâmetros e para a opção **Select allowed SQL statements** (Seleccionar instruções de SQL permitidas), escolha Y (S).
6. O ecrã seguinte apresentará várias palavras-chave de instruções de SQL. Para conceder autoridade para utilização de uma palavra-chave específica, escreva "1" a seguir a essa palavra-chave.
7. Depois de verificar se o perfil de utilizador tem autoridade para executar a instrução CALL, regresse ao ecrã principal Start QM (Iniciar o QM) e seleccione a opção 1.
8. Seleccione a biblioteca adequada onde pretende criar a consulta no parâmetro de biblioteca existente na parte superior do ecrã e prima Enter. Se existirem quaisquer consultas de QM nessa biblioteca, as mesmas serão apresentadas.
9. O parâmetro **Query Creation Mode** existe na biblioteca. Para criar a consulta que executa uma instrução CALL, o **Query Creation Mode** deverá ser SQL. Se **Query Creation Mode** estiver definido como PROMPT, mude para SQL premindo F19 (shift + F7).
10. Depois de verificar se **Query Creation Mode** é SQL, seleccione a opção 1 no campo de opções existente na parte superior para criar uma nova consulta de QM.
11. Um ecrã de criação de consulta deverá aparecer praticamente em branco. É aqui que é especificada a instrução CALL. Para executar a instrução CALL, escreva `CALL ICOLA/ETL`.
12. Quando terminar a criação da consulta de QM, prima F3 para guardar as alterações e atribua um nome à consulta.

Criar o programa de CL

Para criar o programa de CL utilizado para chamar a consulta de QM, siga estes passos:

1. Primeiro, crie o ficheiro fonte para armazenar o programa de CL executando o seguinte comando:
   ```
   >CRTSRCPF FILE(ICOLA/QCLSRC) RCDLEN(92) TEXT('SOURCE FILE FOR CL')
   ```
2. Visualize o Program Development Manager escrevendo STRPDM. Seleccione a opção 3 para trabalhar com membros. No ecrã **Specify Members** (Especificar membros) introduza o ficheiro QCLSRC na biblioteca ICOLA.
3. Quando estiver no ecrã **Work with Members** (Trabalhar com membros), prima F6 para criar um novo membro. Atribua um título ao membro (ETL_CL) e especifique CLLE como o tipo de origem.
4. Será apresentado um ecrã do utilitário de entrada de fontes (SEU - Source Entry Utility) no qual deve introduzir o código do programa de CL. No cenário do iCola, foi introduzido o seguinte código:
   ```
   PGM
   /* Chamar a consulta de QM */
   STRQMQRY QMQRY(ICOLA/ETL_QM)
   /* Preencher o cubo no iSeries B */
   SBMRMTCMD CMD('SBMJOB CMD(CALL PGM(ICOLA/BLD_CUBE))') DDMFILE(ICOLA/TO_SD)
   /* Imprimir para o registo de trabalhos *:
   DSPJOBLOG OUTPUT(*PRINT)
   ENDPGM
   ```
5. Quando terminar a introdução do código do programa de CL, prima F3 para guardar as alterações.
6. Para compilar o programa de CL a partir do ecrã **Work with Members using PDM** (Trabalhar com membros utilizando PDM), seleccione a opção 14 junto do membro. Para determinar se o programa foi compilado com êxito, visualize os ficheiros em spool gerados, utilizando o comando WRKSPLF.

Criar o trabalho agendado

Para criar o trabalho agendado para executar o programa de CL, siga estes passos:

1. A partir da linha de comandos, trabalhe com entradas de trabalhos agendados executando WRKJOBSCDE.
2. Prima F6 para criar um novo trabalho agendado. Quando o trabalho agendado foi criado para o iCola, os seguintes parâmetros foram listados para criar um trabalho agendado para execução diária à meia-noite:
   - Nome do trabalho: ETL
   - Comando a executar: CALL PGM(ICOLA/ETL_CL)
   - Frequência: *WEEKLY
   - Data programada: *NONE
   - Dia programado: *MON, *TUE, *WED, *THU, *FRI, *SAT, and *SUN
   - Tempo programado: 00:00:00
   - Utilizador: DB2OLAP

Autoridade

Quando criar os programas e procedimentos que constituem o processo ETL, é importante verificar se o perfil que está a executar o processo ETL tem a autoridade adequada para todos os novos objectos. No cenário do iCola, o perfil que está a executar o trabalho agendado é DB2OLAP. Uma vez que todos os objectos criados estão na mesma biblioteca (ICOLA), são executados os seguintes comandos:

- Este comando concede propriedade DB2OLAP e toda a autoridade à biblioteca do iCola:
  CHGOWN OBJ('/QSYS.LIB/ICOLA.LIB') NEWOWN(DB2OLAP)
- Este comando concede propriedade DB2OLAP e toda a autoridade a todos os objectos existentes na biblioteca do iCola:
  CHGOWN OBJ('/QSYS.LIB/ICOLA.LIB/*') NEWOWN(DB2OLAP)

Observações chave

A palavra-chave de SQL TRUNC

Ao criar consultas de SQL para o processo ETL, foram necessárias informações sobre o preço da encomenda total, bem como o preço por unidade. Na primeira tentativa para conseguir estes dados foi utilizada esta fórmula:

```
(ICOLA.ORDER.SUBTOTAL / ICOLA.ORDER.QUANTITY)
```

A divisão deixa vários dígitos à direita do decimal. Contudo, o repositório de dados do iCola apenas aceita dois dígitos à direita do decimal. Para garantir que depois do cálculo apenas dois dígitos eram deixados à direita do decimal, foi agora utilizada esta fórmula:

```
TRUNC (ICOLA.ORDER.SUBTOTAL / ICOLA.ORDER.QUANTITY , 2 )
```

Ao especificar o número de casas à direita do decimal no segundo parâmetro da palavra-chave TRUNC (neste caso 2), a fórmula deixa o número especificado de casas decimais e todos os outros números serão truncados.

Dados seleccionados pelo procedimento FACTTABLE

Outra observação chave envolve o procedimento FACTTABLE armazenado. Segue-se o SQL utilizado para criar o procedimento armazenado:

```
CREATE PROCEDURE ICOLA.FACTTABLE ( )
LANGUAGE SQL
```

```
SPECIFIC ICOLA.FACTTABLE
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN INSERT INTO ICOLADM . FACTTABLE ( ORDERID , UPC , MARKETID , TIMEID,
QUANTITYSOLD , SELLINGPRICE , TOTALPRICE , COSTTOBUY , BRANCHID )
( ( SELECT ICOLA . ORDER . ORDERID , ICOLA . ORDER . UPC , ICOLADM . MARKET . MARKETID
, ( ( YEAR ( ICOLA . ORDER . DATE ) * 10000 ) + ( MONTH ( ICOLA . ORDER . DATE ) * 100 ) + DAY (
ICOLA . ORDER . DATE ) ) , ICOLA . ORDER . QUANTITY , TRUNC ( ICOLA . ORDER . SUBTOTAL /
ICOLA . ORDER . QUANTITY, 2 ) , ICOLA . ORDER . SUBTOTAL , ICOLA . PRICE . OURCOST , ICOLA
. CUSTOMERCONTACT . BRANCHID
FROM ICOLA . ORDER , ICOLA . PRODUCT , ICOLA . PRICE , ICOLA . CUSTOMERCONTACT,
ICOLA . CUSTOMERBRANCH , ICOLADM . MARKET
WHERE ICOLA . ORDER . CUSTOMERID = ICOLA . CUSTOMERCONTACT . CUSTOMERID
AND ICOLA . CUSTOMERCONTACT . BRANCHID = ICOLA . CUSTOMERBRANCH . BRANCHID
AND ICOLA . CUSTOMERBRANCH . CITY = ICOLADM . MARKET . CITY
AND ICOLA . CUSTOMERBRANCH . STATE = ICOLADM . MARKET . STATE
AND ICOLA . ORDER . UPC = ICOLA . PRODUCT . UPC AND ICOLA . PRODUCT . PRICECODE =
ICOLA . PRICE . PRICECODE
AND ICOLA.ORDER.DATE < CURRENT DATE)
EXCEPT
( SELECT ORDERID , UPC , MARKETID , TIMEID , QUANTITYSOLD , SELLINGPRICE, TOTALPRICE ,
COSTTOBUY , BRANCHID FROM ICOLADM . FACTTABLE ) ) ;
END ;
```

Originalmente, o procedimento não incluía a condição AND ICOLA.ORDER.DATE < CURRENT DATE na cláusula where, mas esta foi adicionada pelas seguintes circunstâncias:

- Durante o processo ETL que é executado à meia-noite, e após a execução do procedimento CUSTOMER, é registado um novo cliente através do sítio da Web e efectuada uma encomenda antes da execução do procedimento FACTTABLE.

Quando isto acontece, as informações do cliente relativas a essa encomenda não serão registadas no repositório de dados. Por conseguinte, ocorrerão erros durante a criação do cubo de dados, uma vez que são necessários dados dimensionais completos. A solução para este problema consiste em adicionar a cláusula where AND ICOLA.ORDER.DATE < CURRENT DATE à instrução de SQL. Uma vez que o processo ETL está agendado para executar à meia-noite, só não serão incluídas em FACTTABLE, as encomendas efectuadas a partir da meia-noite e enquanto o procedimento FACTTABLE estiver em execução. O tempo decorrido entre a meia-noite e a conclusão da execução do procedimento FACTTABLE é de apenas alguns minutos, assim só algumas encomendas não serão inseridas em FACTTABLE. Essas encomendas serão inseridas no dia seguinte.

Capítulo 4. Cubo multidimensional

No centro do cenário do iCola encontram-se os cubos multidimensionais utilizados para conterem os dados históricos de resumo. Estes cubos podem ser criados com o IBM(R) DB2 OLAP Server. O IBM DB2 OLAP Server(TM) disponibiliza uma análise multidimensional rápida e intuitiva.

Os cubos de base de dados são constituídos por várias dimensões pelas quais os analistas de vendas se interessam, tais como produtos, tempo e contas. A figura 7 ilustra um exemplo de um cubo com estas três dimensões. As dimensões utilizadas para criar um cubo deverão basear-se nos dados que serão extraídos do repositório de dados.

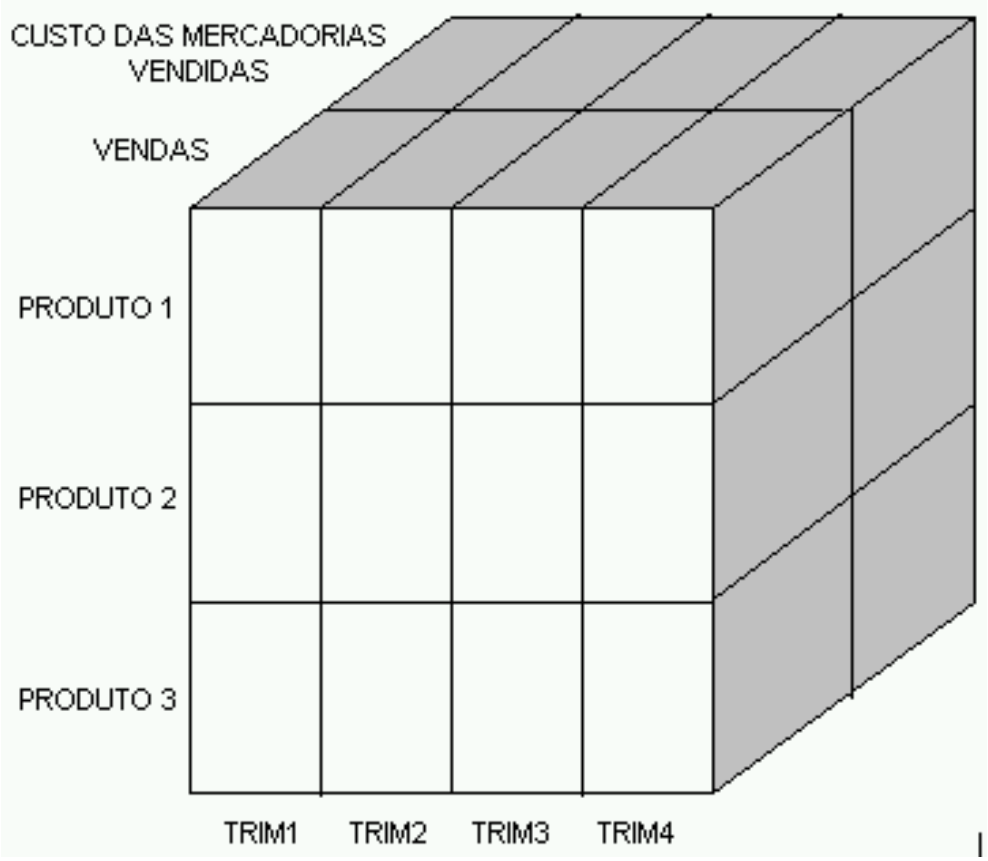

Figura 7: Cubo multidimensional.

Por exemplo, se construir um cubo com as dimensões produtos, tempo e contas, seguem-se alguns exemplos de consultas que podem ser extraídas a partir do cubo:

- Quais os produtos que geraram uma maior margem de lucro (vendas - despesas) durante o primeiro trimestre?
- Em que trimestre o Produto 1 teve uma menor margem de lucro?
- Qual o produto que no decorrer do ano teve maiores despesas iniciais?

Atente no primeiro exemplo e determine quais os produtos que geraram uma maior margem de lucro durante o primeiro trimestre. Para visualizar estes dados, apenas é necessário examinar um sector do cubo, o sector correspondente a QTR1. Do mesmo modo, para visualizar os números das vendas

correspondentes ao Produto 1 no decorrer do ano, deverá ser examinado o sector Produto 1 do cubo. Para visualizar o custo das mercadorias vendidas relativamente a todos os produtos no decorrer do ano, deverá ser examinado o sector de mercadorias vendidas. Felizmente, não é necessário que um utilizador saiba qual o sector ou secção do cubo no qual estão armazenados os dados para gerar resultados para estas consultas. Ao utilizar o IBM DB2 OLAP Server Analyzer 8.1, o utilizador indica quais os dados em que está interessado e os resultados são gerados automaticamente.

As consultas listadas previamente também podem ser geradas utilizando a base de dados operacional (uma base de dados relacional) do iCola, uma vez que contém os mesmos dados. No entanto, a complexidade verificada ao gerar estas e outras consultas diminui acentuadamente ao utilizar um cubo multidimensional. As bases de dados multidimensionais têm a capacidade de examinar dados rapidamente a partir de variados pontos de vista. Em muitas situações, os dados operacionais encontram-se em localizações diferentes, pelo que é impossível reunir informações efectuando consultas aos dados operacionais.

Para recolher as informações correspondentes ao lucros do primeiro trimestre a partir da base de dados operacional, seria necessário efectuar os seguintes cálculos na consulta:

1. Determine quais as encomendas efectuadas no primeiro trimestre.
2. Relativamente a cada encomenda efectuada no primeiro trimestre, determine os dados das vendas multiplicando o custo das vendas do item pela quantidade vendida.
3. Relativamente a cada encomenda efectuada no primeiro trimestre, determine o custo dos dados das mercadorias vendidas multiplicando o custo do produto pela quantidade.
4. Relativamente a cada encomenda efectuada no primeiro trimestre, utilize o resultado do passo 2 e subtraia o resultado do passo 3 para determinar a margem de lucro.
5. Adicione os resultados do passo 4 com base no produto vendido de acordo com a encomenda para determinar a margem de lucro de cada produto.

Ao utilizar o IBM DB2 OLAP Server, é possível aplicar o cálculo a este cenário e muitas outras consultas tornam-se simples para o utilizador final.

Descrição geral da aplicação

Escolher as dimensões do cubo

Os cubos para o cenário do iCola são baseados nestas cinco dimensões:

- TIME
- MEASURES (ACCOUNTS)
- PRODUCT
- MARKET
- STORE

Nota: TIME e MEASURES são dimensões padrão do cubo. Uma vez que os cubos de dados mantêm o registo de dados de resumo do histórico, está sempre presente uma dimensão TIME. Do mesmo modo, os cubos de dados normalmente mantêm o registo de pelo menos uma quantidade numérica (como volume de vendas ou quantidade vendida). Estas informações são registadas na dimensão MEASURES.

O cubo do iCola contém as dimensões PRODUCT, MARKET e STORE uma vez que a empresa pretendia encontrar respostas para questões como:

- De que modo a época do ano e o tipo de embalagem influenciam as vendas de determinadas marcas?
- Quais os estabelecimentos que vendem maior e menor quantidade de uma determinada marca?
- Será que determinados produtos vendem melhor em estabelecimentos mais pequenos vs uma cadeia de estabelecimentos de maiores dimensões?

- De que modo se pode comparar a competitividade nas vendas de vários produtos, por região e por estabelecimento?

Expandir as dimensões no esquema do cubo

Depois de escolher as dimensões do cubo, estas deverão ser expandidas no esquema do cubo com base nos dados utilizados para preencher cada dimensão. Normalmente, cada dimensão contém várias características. Por exemplo, a dimensão MARKET. Cada localização de mercado tem uma região, estado e cidade. Uma vez que as regiões contêm estados e os estados contêm cidades, a dimensão MARKET tem a seguinte hierarquia.

- Mercado (Geração 1/Nível 3)
  - Região (Geração 2/Nível 2)
    - Estado (Geração 3/Nível 1)
      - Cidade (Geração 4/Nível 0)

Estes níveis são também conhecidos como gerações. Os números de geração referem-se a níveis de consolidação numa dimensão e são incrementáveis, começando pela raiz da dimensão até ao último membro da estrutura. Os números de níveis também fazem referência a um ramo numa dimensão; no entanto, os níveis invertem a ordem numérica utilizada para as gerações, contando a partir do último membro em direcção à raiz da estrutura.

Ao definir estas gerações/níveis na dimensão MARKET, poderá extrair os dados de vários mercados. Por exemplo, para determinar o sucesso de um produto durante o ano com base na localização, o cubo consegue calcular por cidade, estado e região.

Carregar dados no cubo

O cubo multidimensional contém duas categorias de dados que têm de ser actualizadas regularmente. Estas categorias são conhecidas normalmente como a criação da dimensão e o carregamento dos dados:

- Criação da dimensão: esta categoria preenche os dados do esquema relativos às dimensões PRODUCT, MARKET e STORE. Estes dados são necessários porque poderão ser adicionados novos produtos, mercados e estabelecimentos ao cenário do iCola. Por exemplo, com a dimensão MARKET, poderão ser adicionadas novas cidades à medida que o iCola expande a respectiva base de mercado.
- Carregamento dos dados: dados de vendas retirados de encomendas individuais, como:
  - Que produto foi comprado?
  - Que quantidade desse produto foi comprada?
  - Quando foi comprado o produto?
  - Quem comprou o produto?

  Estes dados são necessários para calcular o cubo de modo a que as questões relacionadas com as vendas, listadas anteriormente, possam ser respondidas.

Para reunir os valores destas categorias, são criadas regras de carregamento com o gestor de aplicações do DB2 OLAP Server(TM). As regras de carregamento são as que o IBM(R) DB2(R) OLAP Server utiliza para preencher o cubo. Ao preencher o cubo, as regras de carregamento para criar dimensões deverão ser sempre executadas antes das regras de carregamento de dados. Isto porque o cubo tem de ter conhecimento dos detalhes de dimensões de todos os dados possíveis que podem ser introduzidos durante o carregamento de dados. Se o cubo não tiver conhecimento de todos os dados possíveis (por exemplo, são introduzidos novos dados sobre um estabelecimento com uma regra de carregamento de dados antes da adição do estabelecimento através de uma regra de criação de dimensão), o cubo não será correctamente criado.

Ambas as categorias de dados são extraídas do repositório de dados. Para extrair os dados, foram criadas as seguintes regras de carregamento:

- Regras de criação de dimensões:
  - MRKT-RUL: extrai informações geográficas da tabela MARKET para preencher o esquema MARKET
  - PROD-RUL: extrai informações sobre o produto da tabela PRODUCT para preencher o esquema PRODUCT
  - STOR-RUL: extrai informações de arquivo da tabela STORE para preencher o esquema STORE
- Regras do carregamento de dados:
  - EXP-2004: extrai informações sobre despesas da tabela FACTTABLE para o carregamento de dados
  - SAL-2004: extrai informações sobre vendas da tabela FACTTABLE para o carregamento de dados

Poderá obter mais informações sobre estas regras de carregamento na secção de instalação da aplicação.

O processo de carregamento de dados é automatizado utilizando o mesmo programa de CL que foi usado para executar o processo ETL. Depois de concluído o processo ETL no iSeries(TM) A, é chamado um programa no iSeries B para preencher o cubo, utilizando as regras de carregamento listadas anteriormente. (Consulte a figura 2 para obter referências sobre o iSeries A e o iSeries B.)

Pontos de concepção da aplicação

Considerações sobre a dimensão

- O IBM(R) DB2 OLAP Server(TM) maximiza o desempenho e minimiza o armazenamento dividindo as dimensões padrão de um cubo em dois tipos: dimensões densas e dimensões dispersas. Uma dimensão dispersa é uma dimensão que provavelmente tem uma baixa percentagem de posições de dados disponíveis preenchidas. Do mesmo modo, uma dimensão densa é uma dimensão que provavelmente tem uma elevada percentagem de posições de dados disponíveis preenchidas. Esta divisão permite ao IBM DB2(R) OLAP Server examinar os dados que não são distribuídos de modo uniforme, sem perder as vantagens do acesso de estilo matriz aos dados.

  A dimensão TIME é um bom exemplo de uma dimensão densa, uma vez que provavelmente as vendas ocorrerão todos os dias. A dimensão MARKET é um exemplo de uma dimensão dispersa, uma vez que nem todos os produtos são vendidos em todos os mercados.

  As dimensões densas do iCola são:
  - TIME
  - MEASURES (ACCOUNTS)

  As dimensões dispersas do iCola são:
  - PRODUCT
  - MARKET
  - STORE

Considerações sobre o esquema

- Ao criar o cubo, é importante ter em conta a quantidade de detalhes que o cubo deverá conter por forma a responder às questões relativas às vendas da empresa. No que respeita a dimensão TIME, é necessário determinar o tempo mínimo necessário para calcular os dados. No cenário do iCola, um mês é o tempo mínimo estabelecido para satisfazer as necessidades de análise da empresa. Consequentemente, o nível mais baixo da dimensão TIME é um mês.
- Outra decisão tomada diz respeito ao tipo de dados numéricos que existirão no cubo. Ou seja, dados como o volume de vendas, que normalmente são apresentados na dimensão ACCOUNTS. Com base nas consultas que tinham de ser executadas, a empresa decidiu utilizar os dados das despesas e das vendas.
- A figura 8 ilustra como as dimensões PRODUCT, MARKET e STORE são expandidas no esquema do cubo:

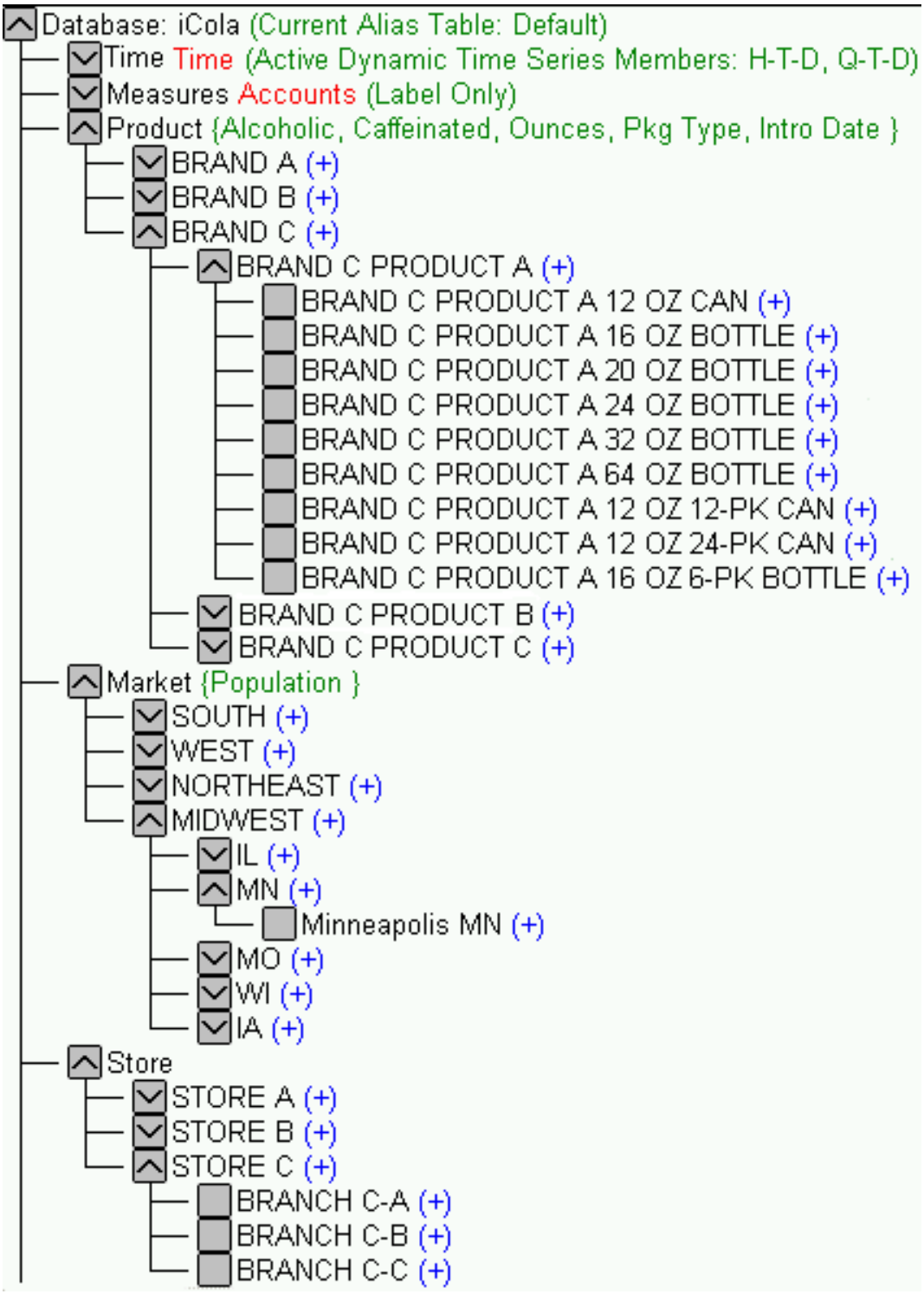

Figura 8: dimensões PRODUCT, MARKET e STORE do iCola.

- Cada produto contém determinadas funcionalidades como tamanho, tipo de pacote e uma data de introdução. O modo mais eficaz de ilustrar isto no esquema do cubo é conferir estes atributos à dimensão PRODUCT. Poderá obter mais informações sobre a criação de atributos no *IBM DB2 OLAP Server 8.1 Database* Administrator's Guide.

Considerações sobre o desempenho

- Um ponto de desempenho mencionado pelo *IBM DB2 OLAP Server 8.1 Database Administrator's Guide* refere-se à optimização do desempenho de **consulta** colocando as dimensões densas antes das dimensões dispersas. Do mesmo modo, refere para colocar as dimensões dispersas consultadas mais frequentemente antes das dimensões dispersas consultadas com menos frequência. Seguindo estas directrizes do cenário do iCola, as dimensões seriam ordenadas da seguinte forma:
  1. TIME (Densa)
  2. MEASURES (Densa)
  3. PRODUCT (Dispersa)
  4. MARKET (Dispersa)
  5. STORE (Dispersa)

  Contudo, outro ponto de desempenho mencionado refere que para optimizar o desempenho de **cálculo**, deverá colocar as dimensões densas antes das dimensões dispersas, mas ordenar as dimensões dispersas começando pelas dimensões com quantidade inferior de membros até às que têm a maior quantidade de membros. Seguindo estas directrizes do cenário do iCola, as dimensões seriam ordenadas da seguinte forma:
  1. TIME (Densa)
  2. MEASURES (Densa)
  3. MARKET (Dispersa)
  4. STORE (Dispersa)
  5. PRODUCT (Dispersa)

  Estas duas teorias entram parcialmente em conflito entre si. O *IBM DB2 OLAP Server 8.1 Database Administrator's Guide* explica que a melhor sequência de ordenação do esquema consiste em dar prioridade de acordo com o que é mais importante: consultas mais rápidas ou cálculos mais rápidos. No caso do iCola, foi decidido que o desempenho das consultas era mais importante do que o desempenho dos cálculos, principalmente porque os cálculos eram efectuados uma vez por dia e as consultas eram efectuadas várias vezes por dia.
- Outra consideração investigada em relação ao desempenho dizia respeito à utilização de vários cubos. No cenário do iCola, um cubo poderia ser utilizado para armazenar dados equivalentes a vários anos. No entanto, uma das preocupações do iCola era o tempo necessário para calcular o cubo e gerar consultas assim que aumentasse o tamanho do cubo. Uma solução seria a utilização de um cubo com partições, o qual futuramente poderá ser implementado no cenário do iCola. A solução actual consiste na utilização de um cubo diferente para cada ano. Uma vez que as informações são armazenadas no cubo mensalmente, a empresa decidiu que ter um cubo para cada ano seria uma soluç°ão prática para este problema. Actualmente, o cenário do iCola tem dois cubos, um com dados de 2003 e outro com dados de 2004.

Instalação da aplicação

Instalar o IBM(R) DB2 OLAP Server(TM)

Para os utilizadores familiarizados com o iSeries(TM) e com as operações do iSeries, a instalação do IBM DB2 OLAP Server no iSeries deverá ser uma tarefa simples. São fornecidas informações adequadas no manual de instalação que orientam o utilizador através de todo o processo de instalação. Este manual foi utilizado exaustivamente na criação da instância-objecto do DB2 OLAP Server necessária para alojar o cubo OLAP no iSeries utilizado no cenário do iCola. O manual de instalação do DB2(R) OLAP Server para o iSeries pode ser encontrado no sítio da Web: DB2 OLAP Server Installation Guide for iSeries. O capítulo 2 ("Before you install") e o capítulo 4 ("Installing Essbase/400 OLAP server components") são os capítulos chave deste documento necessários para instalar o DB2 OLAP Server.

Instalar o Cliente do IBM DB2 OLAP

O manual de instalação do DB2 OLAP Server fornece instruções sobre como instalar a aplicação do Cliente DB2 OLAP no Windows(R). O capítulo 6 ("Installing clients on Windows") permite ajudar o

utilizador neste processo e o capítulo 7 (“Connecting to Essbase/400 OLAP server”) auxilia o utilizador tornando o cliente útil estabelecendo a ligação entre as aplicações de cliente e de servidor. O manual de instalação pode ser encontrado no sítio da Web: DB2 OLAP Server Installation Guide for iSeries.

Criar o cubo

Para criar um cubo multidimensional, verifique, em primeiro lugar, se o DB2 OLAP Server foi iniciado. Para atribuir a um perfil autoridade para executar comandos ESSBASE, introduza:

- ESSBASE/GRTESSAUT USRPRF(*perfil_utilizador*)

Em seguida, utilize esse perfil para iniciar o DB2 OLAP executando este comando:

- ESSBASE/STRESSSVR SERVER(*ALL) JOBD(OLAPBLDSVR/SCJOBD)

Após o DB2 OLAP Server ter sido iniciado, a aplicação cliente pode ser iniciada fazendo clique em Start > Programs > IBM DB2 OLAP Server 8.1 > Application Manager (Iniciar > Programas > IBM DB2 OLAP Server 8.1 > Gestor de aplicações).

Seguem-se alguns termos chave que deve conhecer ao trabalhar com o IBM DB2 OLAP Server:

- Aplicação: projecto que contém bases de dados, scripts de cálculo, regras de carregamento e relatórios.
- Base de dados: o *IBM DB2 OLAP Server 8.1 Database Administrator's Guide* refere-se a um cubo como uma base de dados.

É possível encontrar detalhes sobre como criar aplicações e bases de dados na página 151 do *IBM DB2 OLAP Server 8.1 Database Administrator's Guide*.

Para adicionar dimensões ao esquema, consulte a página 168 deste manual. A forma mais simples de criar um esquema consiste em copiar uma das bases de dados de exemplo fornecidas com o IBM DB2 OLAP Server como, por exemplo, a base de dados básica da aplicação de exemplo. O iCola copiou o esquema Sample > Basic (Exemplo > Básico) e efectuou estas alterações:

- A dimensão de cenário foi removida, uma vez que estas informações não são obrigatórias.
- A dimensão de armazenamento foi adicionada, uma vez que a empresa pretendia manter o registo das informações de armazenamento.
- Uma vez que as regiões de mercado do iCola eram diferentes das regiões de mercado do exemplo básico, as regiões foram alteradas de East, Central, South e West para NorthEast, Midwest, South e West.
- Os produtos enquadrados na dimensão do produto foram eliminados, uma vez que os produtos do iCola eram diferentes dos produtos existentes na aplicação de exemplo.
- O atributo alcoólico foi adicionado à lista de atributos existentes para indicar se uma bebida é alcoólica. Para adicionar atributos a um esquema, consulte a página 249 do manual.

Não é necessário que as dimensões sejam expandidas com grande detalhe no esquema. O esquema será preenchido quando as regras de carregamento de criação de dimensões estiverem criadas e em execução.

Criar regras de carregamento: criação dinâmica de dimensões

Conforme previamente mencionado, foram criadas três regras de carregamento de criação de dimensões para assegurar que o esquema permanecia actualizado quando eram adicionados novos mercados, produtos e estabelecimentos. Para criar regras de carregamento com a finalidade de criar dimensões de forma dinâmica, siga os próximos passos: (para obter ajuda adicional, consulte o capítulo 19 do *IBM DB2 OLAP Server TM 8.1 Database Administrator's Guide*).

1. Crie um novo ficheiro de regras seleccionando o botão de regras de carregamento de dados no ecrã do cliente do DB2 OLAP Server

e fazendo clique em **New** (Novo).

2. Para definir a instrução de SQL utilizada para criar o ficheiro de regras, seleccione **File > Open SQL** (Ficheiro > Abrir SQL).
3. Seleccione o servidor, a aplicação e a base de dados apropriados e, em seguida, faça clique em OK.
4. Escolha a origem de dados apropriada e introduza a instrução de SQL para extrair as informações sobre a criação de dimensões do repositório de dados no espaço fornecido. Faça clique em **OK/Retrieve** (OK/Obter) quando tiver concluído. Se as informações de nome de utilizador e palavra-passe forem obrigatórias, escreva o nome de utilizador e a palavra-passe de administrador do DB2 OLAP correspondentes ao DB2 OLAP Server.
5. Em seguida, é apresentado o **Data Prep Editor** (Editor de preparação de dados). Se a instrução de SQL tiver tido êxito na obtenção de dados, estes são apresentados no editor.
6. É necessário associar a regra a um esquema. Para associar a regra a um esquema, seleccione **Options > Associate Outline** (Opções > Associar esquema) no menu principal. Seleccione o servidor, a aplicação e a base de dados apropriados e, em seguida, faça clique em **OK**.
7. Preencha as propriedades do campo do esquema fazendo clique no botão **Define Properties** (Definir propriedades)

no ecrã **Data Prep Editor** (Editor de preparação de dados).

a. Ao criar uma regra de carregamento para criar uma dimensão de forma dinâmica, verifique estas propriedades em todas as colunas no separador **Global Properties** (Propriedades globais):
   - A caixa **Data Field** (Campo de dados) deverá estar desmarcada. O facto de estar desmarcada justifica-se uma vez que a regra não se aplica ao carregamento de dados.
   - A caixa **Ignore field during dataload** (Ignorar campo durante o carregamento de dados) deverá estar marcada. O facto de estar marcada justifica-se uma vez que esta regra é utilizada para modificar uma dimensão e não para carregar dados.
   - A caixa **Ignore field during dimension build** (Ignorar campo durante a criação de dimensões) deverá estar desmarcada. O facto de estar desmarcada justifica-se uma vez que se trata de uma regra para modificar a dimensão e não deve ser ignorada.
   - A caixa **Drop leading/trailing whitespace** (Deixar espaço em branco inicial/final) deverá estar marcada. Este procedimento permite assegurar que, se um campo tiver um espaço em branco adicional, a criação da dimensão não será afectada.

b. Uma vez que a caixa **Ignore field during dataload** (Ignorar campo durante o carregamento de dados) está desmarcada, o utilizador não poderá seleccionar o separador **Data Load Properties** (Propriedades de carregamento de dados). No entanto, o utilizador pode e deve seleccionar o separador **Dimension Build Properties** (Propriedades de criação da dimensão) para mapear todos os campos para o esquema da dimensão.
   - Em **Field Type** (Tipo de campo), seleccione Generation (Geração) e introduza o número de geração apropriado na caixa **Number** (Número).
   - Em **Dimension** (Dimensão), escolha a dimensão apropriada a partir da lista pendente **Dimension** (Dimensão).

   Se estiver a mapear um atributo para um campo:
   - Em **Field Type** (Tipo de campo) seleccione o nome do atributo, listado em Attribute Dimensions (Dimensões do atributo) na lista pendente **Field Type** (Tipo de campo).
   - Introduza o número de geração de nível mais baixo correspondente à dimensão na caixa **Number** (Número).

- Em **Dimension** (Dimensão), escolha a dimensão definida pelo atributo.

8. Verifique **Dimension Build Settings** (Definições da criação de dimensões) fazendo clique no botão

. No separador **Dimension Build Settings** (Definições da criação de dimensões), certifique-se de que **Build Method** (Método de criação) corresponde a: **Use Generation References** (Utilizar referências de geração).

9. Guarde o ficheiro de regras.

Para ajudar a familiarizar-se melhor com o modo como as regras de criação de dimensões são criadas, são apresentados, em seguida, os passos para criar uma das regras de criação de dimensões do iCola, MRKT-RUL:

1. Foi criado um novo ficheiro de regras seleccionando o botão de regras de carregamento de dados no ecrã do DB2 OLAP Server

e fazendo clique em New (Novo).

2. Em seguida, a instrução de SQL utilizada para criar o ficheiro de regras é definida seleccionando **File > Open SQL** (Ficheiro > Abrir SQL).
3. Foram seleccionados o servidor iSeries B (consulte a figura 2 para obter referências ao iSeries A e ao iSeries B), a aplicação iCola e a base de dados 2004.
4. Para a origem de dados, foi escolhido o iSeries A e foi introduzida a seguinte instrução de SQL:
   SELECT ICOLADM.MARKET.REGION, ICOLADM.MARKET.STATE,
   CONCAT(RTRIM(ICOLADM.MARKET.CITY), CONCAT(' ', ICOLADM.MARKET.STATE))
   FROM ICOLADM.MARKET

   Esta instrução de SQL é utilizada, uma vez que extrai informações para cada geração/nível na dimensão MARKET (MERCADO).
5. Após premir **OK/Retrieve** (OK/Obter), é apresentado um pedido de informação correspondente ao nome de utilizador e palavra-passe.
6. O **Data Prep Editor** (Editor de preparação de dados) é apresentado, conforme ilustrado na figura 9. Repare que, em vez de ser extraída apenas a cidade, esta foi concatenada com o estado. Esta situação ocorre, uma vez que é necessário que cada elemento de dados da dimensão seja único durante o carregamento de dados. Por exemplo, ao extrair informações relativas a Rochester, MN, se for retirado apenas o nome da cidade Rochester, o DB2 OLAP Server não detectaria a diferença entre Rochester, MN, e Rochester, NY, ao executar o carregamento de dados. A concatenação dos campos de cidade e estado permite assegurar que o campo de cidade é único.

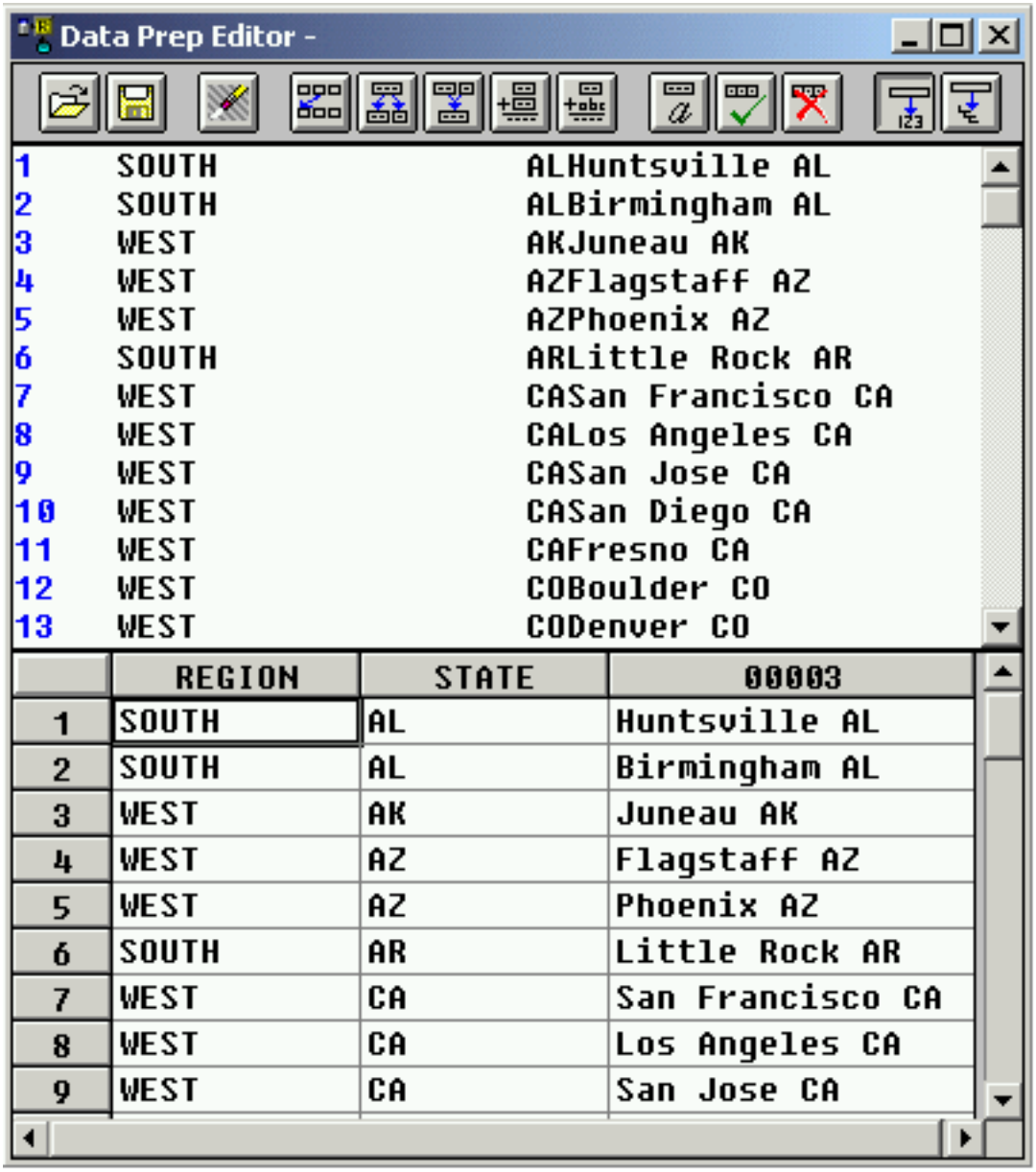

Figura 9: Campos de dimensão correspondentes à dimensão de mercado

7. Em seguida, a regra foi associada ao esquema de 2004.
8. Propriedades dos campos:
   - Nas propriedades dos campos, foram marcados os campos apropriados no separador **Global Properties** (Prorpiedades globais).
   - As figuras de 10 a 12 fornecem os valores introduzidos para cada geração no separador **Dimension Build Properties** (Propriedades de criação de dimensões). Repare que estas informações correspondem às informações de geração da secção ″Descrição geral da aplicação″.

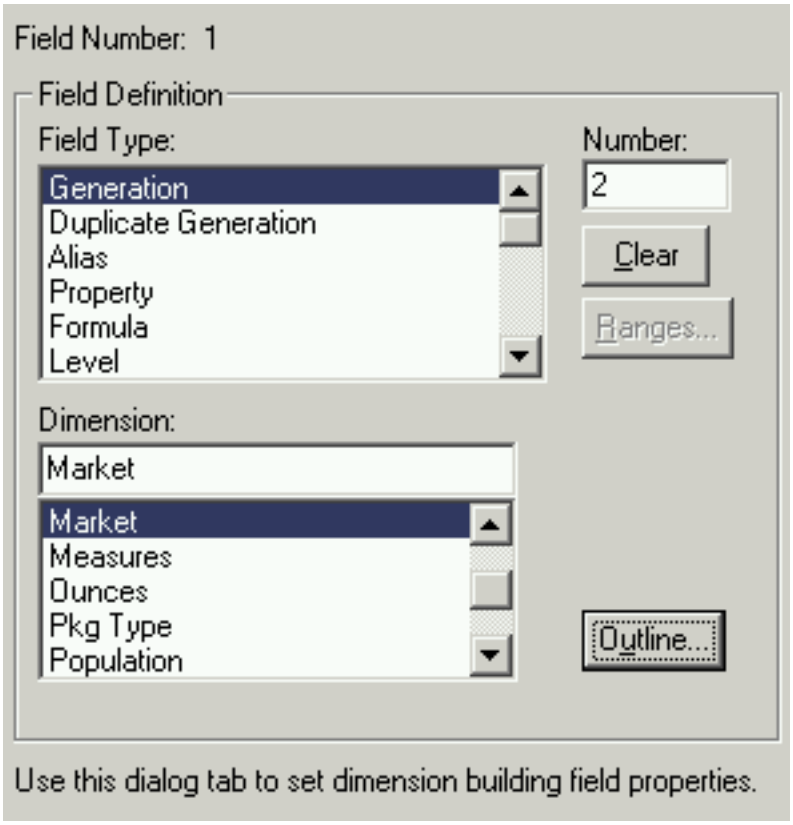

Figura 10: Definição do campo de região

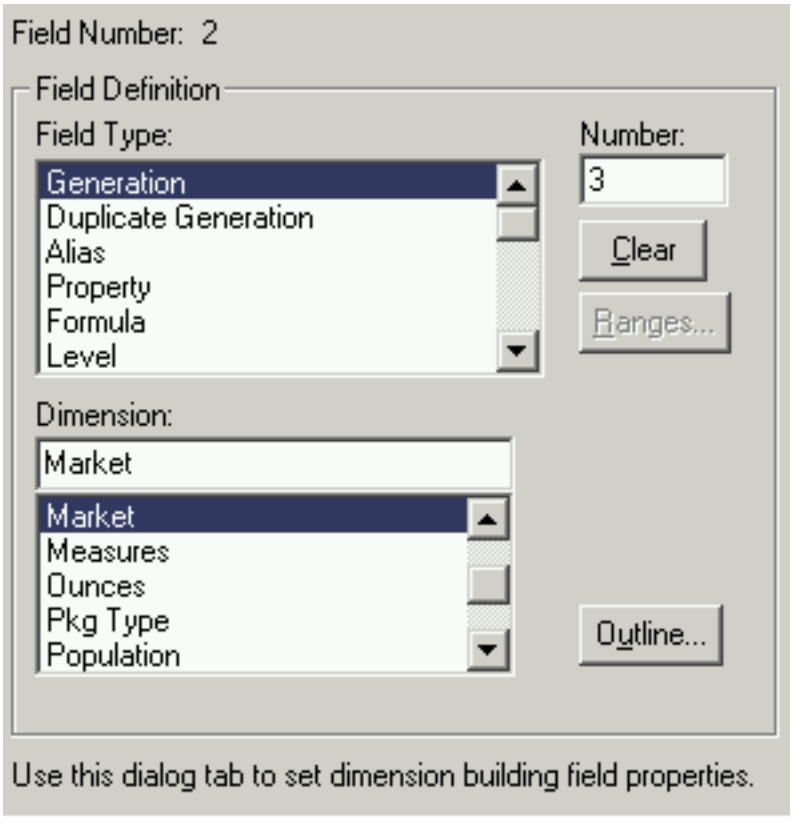

Figura 11: Definição do campo de estado

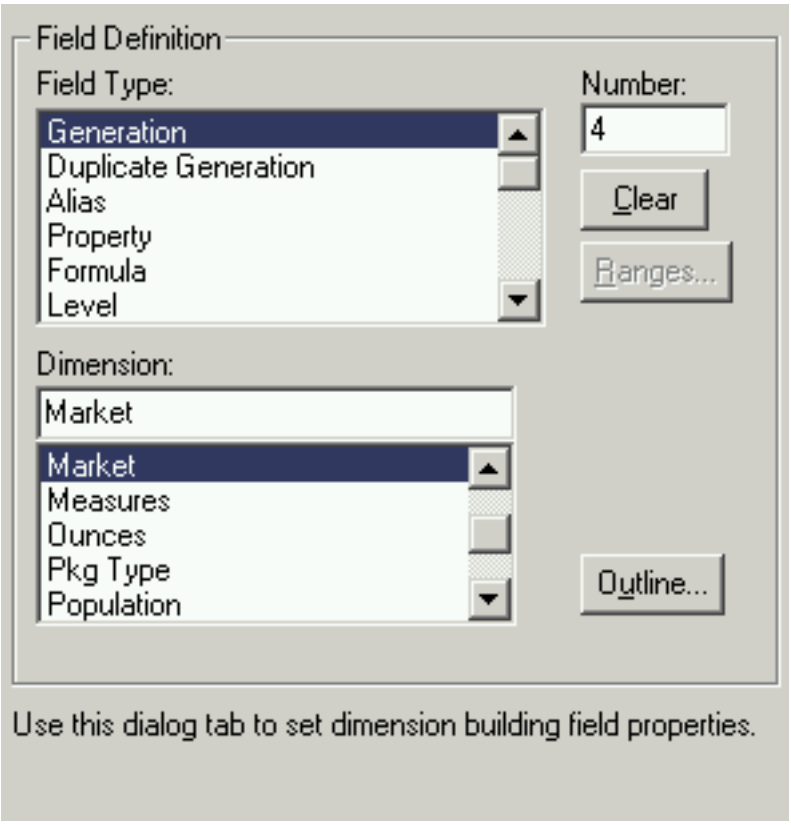

Figura 12: Definição do campo de cidade

9. A definição **Use Generation References** (Utilizar referências de geração) estava marcada no separador **Dimension Build Settings** (Definições de criação de dimensões).
10. O ficheiro de regras foi guardado.

As instruções de SQL utilizadas para criar regras de criação de dimensões do iCola são fornecidas no Apêndice C.

Criar regras de carregamento - carregar dados

Além de garantir que as dimensões são actuais, é necessário que os dados do cubo sejam actuais. Foram criadas regras de carregamento para carregar os dados actuais. Para criar regras de carregamento de dados, siga os próximos passos: (para obter informações adicionais, consulte também o capítulo 20 do manual *IBM DB2 OLAP Server 8.1 Database Administrator's Guide*)

1. Execute os passos de 1 a 6 da secção de criação dinâmica de dimensões para criar a regra de carregamento.

   A instrução de SQL utilizada para carregar os dados actuais contém o número de geração mais elevado (também conhecido como número de nível mais baixo) para cada dimensão. Por exemplo, nas regras de carregamento de dados do iCola, a instrução de SQL contém:

   - o mês em que foi efectuada a venda
   - se o número se refere às despesas ou às vendas
   - o produto vendido
   - a cidade na qual foi efectuada a venda
   - o estabelecimento no qual foi vendido o produto

   A instrução de SQL também contém a quantidade total da venda. Ou seja, o volume de dados quantitativos que será utilizado nos próximos passos.

Consulte o Apêndice C para obter as instruções de SQL utilizadas pelo iCola para criar as regras de carregamento de dados.

2. Preencha as propriedades de campo do esquema. No ecrã **Data Prep Editor** (Editor de preparação de dados), faça clique no botão **Define Properties** (Definir propriedades)

.

   a. Ao criar uma regra de carregamento de dados, verifique estas propriedades para todas as colunas **excepto** para o campo que contém os dados quantitativos que pretende carregar (geralmente um valor numérico) no separador **Global Properties** (Propriedades globais):
      - A caixa **Data Field** (Campo de dados) deverá estar desmarcada. O facto de estar desmarcada justifica-se uma vez que o campo específico não é o campo que contém os dados.
      - A caixa **Ignore field during dataload** (Ignorar campo durante o carregamento de dados) deverá estar desmarcada. O facto de estar desmarcada justifica-se uma vez que esta regra é utilizada para carregar dados e não deverá ser ignorada.
      - A caixa **Ignore field during dimension build** (Ignorar campo durante a criação de dimensões) deverá estar marcada. O facto de estar marcada justifica-se uma vez que se trata de uma regra para carregamento de dados e não para criar uma dimensão.
      - A caixa **Drop leading/trailing whitespace** (Deixar espaço em branco inicial/final) deverá estar marcada. Este procedimento permite assegurar que, se um campo tiver um espaço em branco adicional, não afectará a criação da dimensão.

      No campo que contém os dados quantitativos, a caixa **Data Field** (Campo de dados) deverá estar marcada. Todos os restantes campos permanecem idênticos como na lista anterior.

   b. Uma vez que a caixa Ignore field during dimension build (Ignorar campo durante a criação de dimensões) está desmarcada, o utilizador não poderá seleccionar o separador **Dimension Build Properties** (Propriedades de criação de dimensões). No entanto, o utilizador pode seleccionar o separador **Data Load Properties** (Propriedades de carregamento de dados) e atribuir nomes aos campos de carregamento de dados. Contudo, este passo não é obrigatório.

3. Marque **Dimension Build Settings** (Definições da criação de dimensões) fazendo clique no botão

. No separador **Dimension Build Settings** (Definições da criação de dimensões), verifique se **Build Method** (Método de criação) corresponde a: **Use Generation References** (Utilizar referências de geração).

4. Guarde o ficheiro de regras.

Para visualizar as instruções de SQL utilizadas para criar regras de criação de dimensões do iCola, consulte o Apêndice C.

Limpar dados

Antes de carregar dados para o cubo, deverá limpar o cubo para assegurar um cálculo correcto. Para limpar um cubo, a partir do menu principal seleccione Database > Clear Data > All (Base de dados > Limpar dados > Tudo).

Executar regras de carregamento de dados e criação de dimensões manualmente

Após criar as regras de carregamento de dados e de criação de dimensões, é necessário executar as regras para modificar a base de dados. Para executar as regras manualmente, siga estes passos:

1. A partir do menu principal, seleccione **Database** > **Load Data** (**Base de dados** > **Carregar dados**).
2. Seleccione o servidor, a aplicação e a base de dados para criar o esquema.

3. Seleccione Type (Tipo) como SQL.
4. Nas opções, verifique se **Modify Outline** (Modificar esquema) está seleccionado e se **Load Data** (Carregar dados) está desmarcado, se estiver a modificar o esquema. Se estiver a carregar dados, verifique se **Modify Outline** (Modificar esquema) está desmarcado e se **Load Data** (Carregar dados) está marcado.
5. Relativamente ao utilizador e palavra-passe de SQL, escreva o nome de utilizador e a palavra-passe de administrador de DB2 OLAP.
6. Marque a caixa **Use Rules File** (Utilizar ficheiro de regras) e utilize o botão de pesquisa para seleccionar a regra apropriada.
7. Faça clique em OK. É possível que seja apresentado um pedido de informação solicitando se deverá ser sobreposto ou anexado um novo ficheiro de erros: seleccione sim ou não conforme apropriado.

Calcular dados

É necessário que os dados sejam calculados após serem carregados para o cubo. Para calcular os dados, seleccione Database > Calculate (Base de dados > Calcular). Verifique se estão seleccionados o servidor, a aplicação, a base de dados e o script de cálculo correctos (Um script de calculadora fornece informações ao cubo sobre como calcular dados. O script contém comandos, equações e fórmulas de cálculo. Se não tiverem sido criados quaisquer scripts de cálculo, pode ser utilizado o script assumido). Faça clique em OK para calcular os dados.

Automatizar o processo de carregamento de dados e criação de dimensões

Uma vez que é necessário que os processos de limpeza de dados, modificação do esquema, carregamento de dados e cálculo de dados sejam executados todos os dias, foi criado um programa de CL denominado BLD_CUBE para automatizar estes processos. Os comandos do DB2 OLAP Server fornecidos para o iSeries foram utilizados para criar BLD_CUBE. O programa BLD_CUBE é chamado após o processo ETL ser concluído.

Ainda que a medida mais pequena na dimensão TIME (TEMPO) seja de um mês, o cenário do iCola executa o programa BLD_CUBE diariamente. É executado diariamente para permitir que a empresa execute análises em meados do mês às vendas.

O programa BLD_CUBE contém o código que segue.

```
PGM
/* Iniciar sessão no DB2 OLAP com a palavra-passe e ID de administrador do DB2 OLAP */
ESSBASE/LOGINESS SVRUSER(DB2OLAP) SVRPW(PASSW0RD)
/* Escolher a base de dados 2004 na aplicação do ICOLA*/
ESSBASE/RUNESSCMDC COMMAND('SELECT "ICOLA" "2004"')
/* Limpar a base de dados */
ESSBASE/CLRESSDB APPNAME(ICOLA) DBNAME(2004)
/* Criar as dimensões utilizando os ficheiros de regras */
ESSBASE/BLDESSDIM APPNAME(ICOLA) DBNAME(2004) RULEFILE('MRKT-ALL') ERRFILE(ERROR)
SQLUSER(DB2OLAP) +
SQLPW(PASSW0RD)
ESSBASE/BLDESSDIM APPNAME(ICOLA) DBNAME(2004) RULEFILE('PROD-ALL') ERRFILE(ERROR)
SQLUSER(DB2OLAP) +
SQLPW(PASSW0RD)
ESSBASE/BLDESSDIM APPNAME(ICOLA) DBNAME(2004) RULEFILE('STOR-ALL') ERRFILE(ERROR)
SQLUSER(DB2OLAP) +
SQLPW(PASSW0RD)
/* Carregar os dados utilizando os ficheiros de regras */
ESSBASE/IMPESSSQL APPNAME(ICOLA) DBNAME(2004) RULEFILE('SAL-2004') +
ERRFILE('/ESSBASE/APP/ICOLA/2004/ERROR.TXT') SQLUSER(DB2OLAP) SQLPW(PASSW0RD)
```

```
ESSBASE/IMPESSSQL APPNAME(ICOLA) DBNAME(2004) RULEFILE('EXP-2004') +
ERRFILE('/ESSBASE/APP/ICOLA/2004/ERROR.TXT') SQLUSER(DB2OLAP) SQLPW(PASSW0RD)
/* Calcular os dados */
ESSBASE/CLCESSDFT APPNAME(ICOLA) DBNAME(2004)


ENDPGM
```

Observações chave

Chamar o processo de preenchimento do cubo remotamente

O programa que carrega o cubo faz parte do processo ETL agendado. Lembre-se que os dados operacionais e o repositório de dados estão armazenados no iSeries(TM) A e o cubo está armazenado no iSeries B. Depois de concluído o processo ETL no iSeries A, é chamado um programa no iSeries B para preencher o cubo. Originalmente, havia um programa agendado no iSeries B para criar o cubo. No entanto, foi difícil determinar a conclusão do processo ETL, e uma vez que o repositório de dados aumentava continuamente, o tempo necessário para a conclusão do processo ETL variava todas as noites. Para resolver o problema relacionado com a determinação da conclusão do processo ETL no iSeries A, o preenchimento do cubo foi chamado remotamente a partir do iSeries A e, deste modo, o preenchimento do cubo teria início assim que o processo ETL terminasse. Assim, se ocorresse alguma falha durante o processo ETL, o cubo não voltaria a ser preenchido.

Para chamar o processo de preenchimento do cubo remotamente a partir do iSeries A, foram executados os seguintes passos:

1. Era necessário que o iSeries A e o iSeries B comunicassem entre si. No cenário do iCola, foi adicionada uma entrada do sistema central à tabela do sistema central do iSeries A para apontar para o iSeries B.
2. Foi criado um ficheiro da gestão de dados distribuídos (DDM - Distributed Data Management) no iSeries A com referência ao iSeries B. A DDM controla o processamento de ficheiros remotos e permite que aplicações em execução num servidor iSeries acedam aos ficheiros de dados armazenados num outro servidor que suporte DDM. Do mesmo modo, outros sistemas com DDM conseguem aceder a ficheiros na base de dados do servidor iSeries local. A DDM facilita a distribuição do processamento de ficheiros entre dois ou mais servidores.

   Este ficheiro de DDM foi criado através da execução do seguinte comando no iSeries A:

   - CRTDDMF FILE(ICOLA/TO_SYSTM_B) RMTFILE(ICOLA/QCLSRC) RMTLOCNAME(*iSeries_B_Host_Name* *IP)

Após a criação do ficheiro de DDM, poderá ser chamado o programa de preenchimento do cubo no iSeries B através do comando de envio remoto (SBMRMTCMD).

Chamar o processo de preenchimento do cubo como um trabalho não interactivo

Originalmente, após a conclusão do processo ETL no iSeries A, o programa para criar o cubo no iSeries B era chamado directamente:

- SBMRMTCMD CMD('CALL PGM(ICOLA/BLD_CUBE)') DDMFILE(ICOLA/TO_SYSTM_B)

No entanto, chamar o programa directamente significava que o iSeries A teria de aguardar pela conclusão do preenchimento do cubo do iSeries B para conseguir concluir o respectivo programa. Esta não parecia uma solução ideal, assim a chamada do programa passou a ser submetida como um trabalho não interactivo:

- SBMRMTCMD CMD('SMBJOB CMD(CALL PGM(ICOLA/BLD_CUBE))') DDMFILE(ICOLA/TO_SYSTM_B)

Ao executar o programa de preenchimento do cubo submetendo-o como um trabalho não interactivo permite que o iSeries A conclua o respectivo programa enquanto o iSeries B preenche o cubo.

Permitir a ligação do iSeries B ao repositório de dados do iSeries A

Como mencionado anteriormente, os dados do repositório de dados no iSeries A são utilizados para preencher o cubo no iSeries B. Para que o iSeries B consiga extrair dados do iSeries A, foi adicionada uma entrada da base de dados relacional (RDB - Relational Database) no iSeries B para apontar para o iSeries A. Para adicionar uma entrada da base de dados relacional, utilize o comando Work with Relational Database Directory Entries (WRKRDBDIRE).

Do mesmo modo, para que o iSeries B acedesse ao repositório de dados do iSeries A, era necessário que existisse um utilizador no iSeries A com o mesmo nome de perfil e palavra-passe do utilizador no iSeries B que estava a pedir os dados.(No cenário do iCola, este utilizador era o administrador do OLAP DB2(R).) O perfil correspondente no iSeries A tem também a autoridade para as tabelas de base de dados que estão a ser acedidas. Se não existir um perfil no iSeries A que corresponda ao perfil do iSeries que está a pedir os dados (iSeries B), poderão ocorrer erros de autoridade.

Utilizar palavras-chave de SQL como nomes de tabelas

As regras de carregamento de dados foram criadas inicialmente utilizando estas instruções de SQL:

- SELECT ..., **TIME**.MONTH, ...)
  FROM ICOLADM.FACTTABLE, ICOLADM.CUSTOMER, ICOLADM.**TIME**, ICOLADM.PRODUCT, ICOLADM.MARKET
  WHERE... AND ICOLADM.FACTTABLE.TIMEID = **TIME**.TIMEID AND ...

Ao utilizar estas instruções, foram encontrados erros referentes à tabela TIME. No V5R3, TIME é uma palavra reservada em SQL, com significado especial. Para executar a consulta sem ter de mudar o nome da tabela, foi atribuído um novo alias à tabela, como demonstrado a seguir:

- SELECT ..., **MYTIME**.MONTH, ...)
  FROM ICOLADM.FACTTABLE, ICOLADM.CUSTOMER, ICOLADM.**TIME AS MYTIME**, ICOLADM.PRODUCT, ICOLADM.MARKET
  WHERE... AND ICOLADM.FACTTABLE.TIMEID = **MYTIME**.TIMEID AND ...

Utilizar palavras-chave de SQL menos conhecidas

Algumas palavras-chave de SQL menos conhecidas revelaram ser muito úteis durante a criação de regras de carregamento:

- RTRIM: esta palavra-chave corta os espaços em branco do lado direito do campo de entrada. Por exemplo, a cadeia RTRIM('Rochester ') = 'Rochester'. A palavra-chave RTRIM foi útil para assegurar a remoção de espaços em branco excedentes ao concatenar cadeias.

  (Existe uma função LTRIM correspondente para cortar espaços em branco do lado esquerdo)
- CONCAT: a palavra-chave CONCAT associa duas cadeias diferentes numa única cadeia.
  - Por exemplo, CONCAT('String 1', 'String 2') = 'String 1String 2'.

  Para adicionar um espaço entre as duas cadeias, utilize CONCAT('String 1, CONCAT(' ', 'String 2')). Este procedimento foi útil na criação de campos de geração únicos através da associação de dois elementos.

Capítulo 5. Analyzer

O IBM(R) DB2 OLAP Server(TM) Analyzer 8.1 é um produto que foi concebido para efectuar a integração e optimizar toda a capacidade da aplicação do DB2 OLAP Server. Fornece uma interface intuitiva baseada na Web para visualização dos dados existentes nos cubos OLAP gerados pelo DB2 OLAP Server, além disso, e mais importante, permite aos utilizadores criar relatórios analíticos com base nos dados incluídos nos cubos OLAP subjacentes. Em seguida, estes relatórios podem ser utilizados para tomar decisões comerciais importantes no sentido de melhorar as práticas e tendências comerciais actuais ou passadas.

O Analyzer permite aos utilizadores transformar dados multidimensionais e relacionais complexos em relatórios completos que conseguem reflectir as tendências na venda de produtos, os lucros do negócio e outras métricas de desempenho comercial importantes. Em seguida, as análises podem ser partilhadas e geridas pelos departamentos de vendas, marketing, entre outros departamentos da área numa empresa.

É possível criar vários tipos de relatórios utilizando o Analyzer: folhas de cálculo, gráficos de linhas, gráficos circulares, gráficos de barras, cronologias, gráficos de espaço, etc. A criação de um relatório é uma tarefa tão simples como a utilização dos controlos de arrastar e largar, mas numa fase inicial requer algum raciocínio e esquematização. Quando tiver criado um relatório, tem a possibilidade de efectuar a pesquisa alargada ou a pesquisa detalhada nos dados para criar um novo relatório a partir das vistas recentemente geradas. Em seguida, cada relatório pode ser guardado no servidor do Analyzer que armazena a definição dos relatórios para uma posterior utilização. Sempre que a definição do relatório for novamente acedida visualizando um cubo OLAP, os dados do relatório são actualizados com base nos dados actuais existentes no cubo. As opções de filtro também estão disponíveis para restringir de forma mais significativa as análises executadas num relatório.

O Analyzer fornece duas interfaces principais: um cliente Web baseado em Java(TM) e um cliente Web baseado em HTML. Ao utilizar o cliente Web Java, os utilizadores têm permissão para criar e guardar relatórios analíticos muito complexos através da utilização de uma interface gráfica elaborada. Esta parte da aplicação destina-se aos programadores de relatórios e utilizadores experientes que necessitam de ter capacidade para utilizar funcionalidades mais avançadas ao trabalhar com relatórios. Por outro lado, o cliente Web HTML destina-se a um público diferente: os indivíduos que utilizam os relatórios de análise comercial, mas que não os criam. Os directores comerciais e os gestores de projecto são geralmente os utilizadores deste tipo de interfaces. Os programadores também podem optar por expandir as capacidades do IBM DB2 OLAP Server Analyzer 8.1 utilizando o conjunto de ferramentas API do Analyzer, que permite criar aplicações de análise comercial personalizadas e baseadas na Web.

O DB2 OLAP Server Analyzer não se trata de uma ferramenta de exploração de dados. Em contrapartida, centra-se na criação de relatórios comerciais com base nas "consultas" ou questões colocadas pelo programador de relatórios. Geralmente, o programador de relatórios cria um relatório que cumpre os requisitos de que um gestor ou director comercial necessita. A exploração de dados pode ser realizada através da utilização de outras ferramentas; o Analyzer permite ao utilizador criar relatórios a partir de dados OLAP e determinar acções comerciais com base nas tendências visualizadas nos relatórios.

O próprio IBM DB2 OLAP Server Analyzer 8.1 baseia-se no Hyperion Analyzer 6.1 e, por valor assumido, utiliza o IBM WebSphere(R) Application Server compatível com J2EE. O IBM DB2(R) Personal Edition é utilizado como o repositório relacional assumido necessário para funcionalidades específicas do Analyzer. No que diz respeito às finalidades do cenário do iCola, não foram utilizadas as duas ferramentas de administração do Analyzer incluídas com a aplicação Analyzer. Apenas foram necessários para esta aplicação um servidor, um utilizador/grupo e uma ligação de base de dados.

Descrição geral da aplicação

No cenário do iCola, o principal mecanismo para visualizar e analisar directamente os dados contidos num cubo de dados do DB2(R) OLAP é o produto DB2 OLAP Analyzer. A aplicação de servidor Analyzer é utilizada neste cenário como principal terminal para visualizar os dados do cubo do DB2 OLAP e verifica se os dados de análise são alterados após os cubos terem sido recriados. No mundo comercial, o Analyzer é o componente chave para gerar relatórios e gráficos que orientem a estratégia comercial futura de uma empresa, utilizando o DB2 OLAP Server(TM) como a respectiva solução de Business Intelligence.

Após instalar o Analyzer Server for Windows(R) e iniciar a aplicação, o utilizador é direccionado para a página de inicialização do Analyzer (Figura 13), que contém essencialmente ligações para todas as ferramentas disponíveis fornecidas pela aplicação.

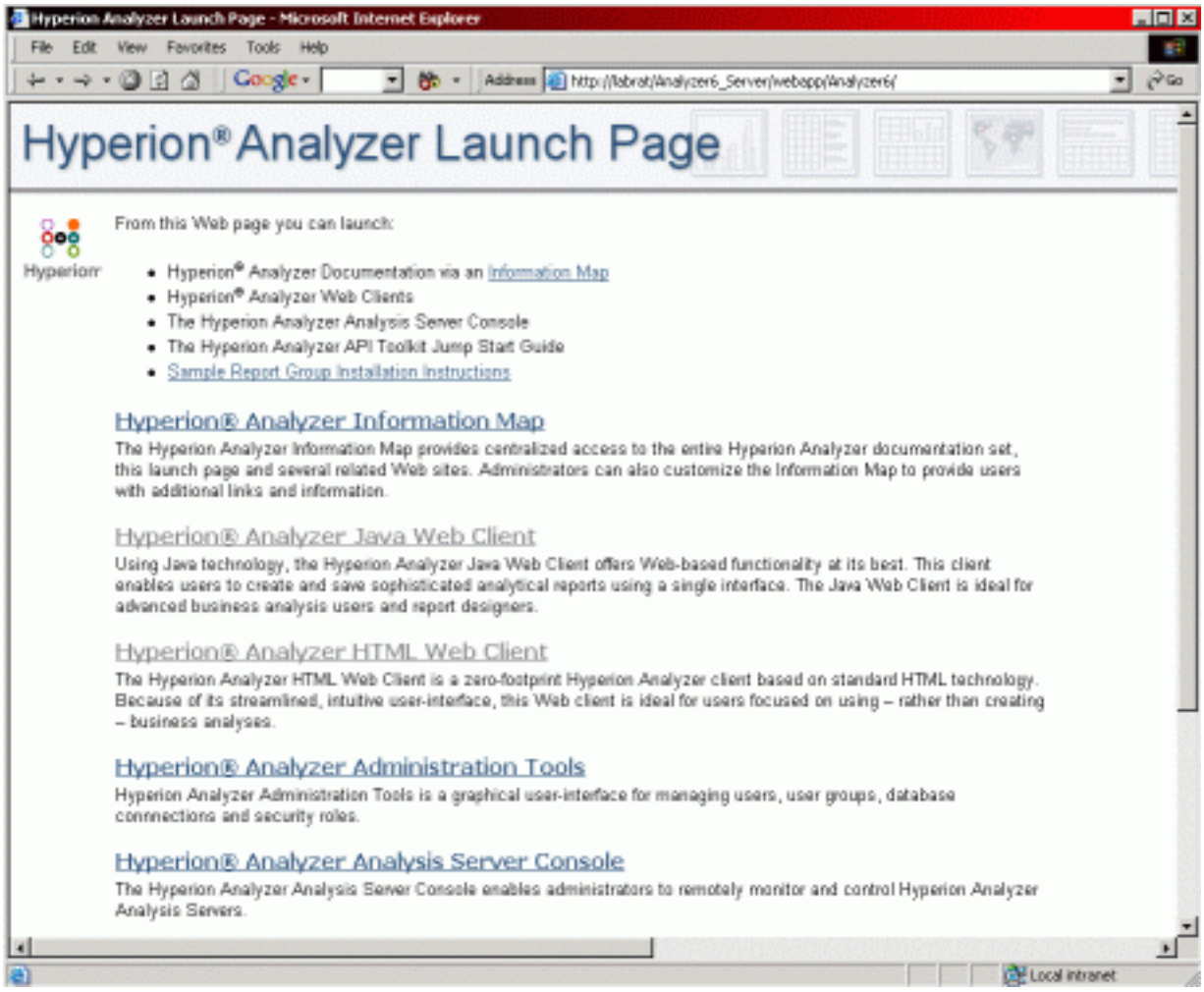

Figura 13: Página de inicialização do Analyzer

O cliente Web baseado em Java(TM) e o cliente Web baseado em HTML são utilizados no cenário do iCola, embora cada cliente tenha um objectivo próprio específico no cenário. O cliente Web baseado em Java (Figura 14) foi utilizado durante a fase de concepção para criar relatórios com base no conjunto seleccionado de questões comerciais revistas antes de serem implementadas.

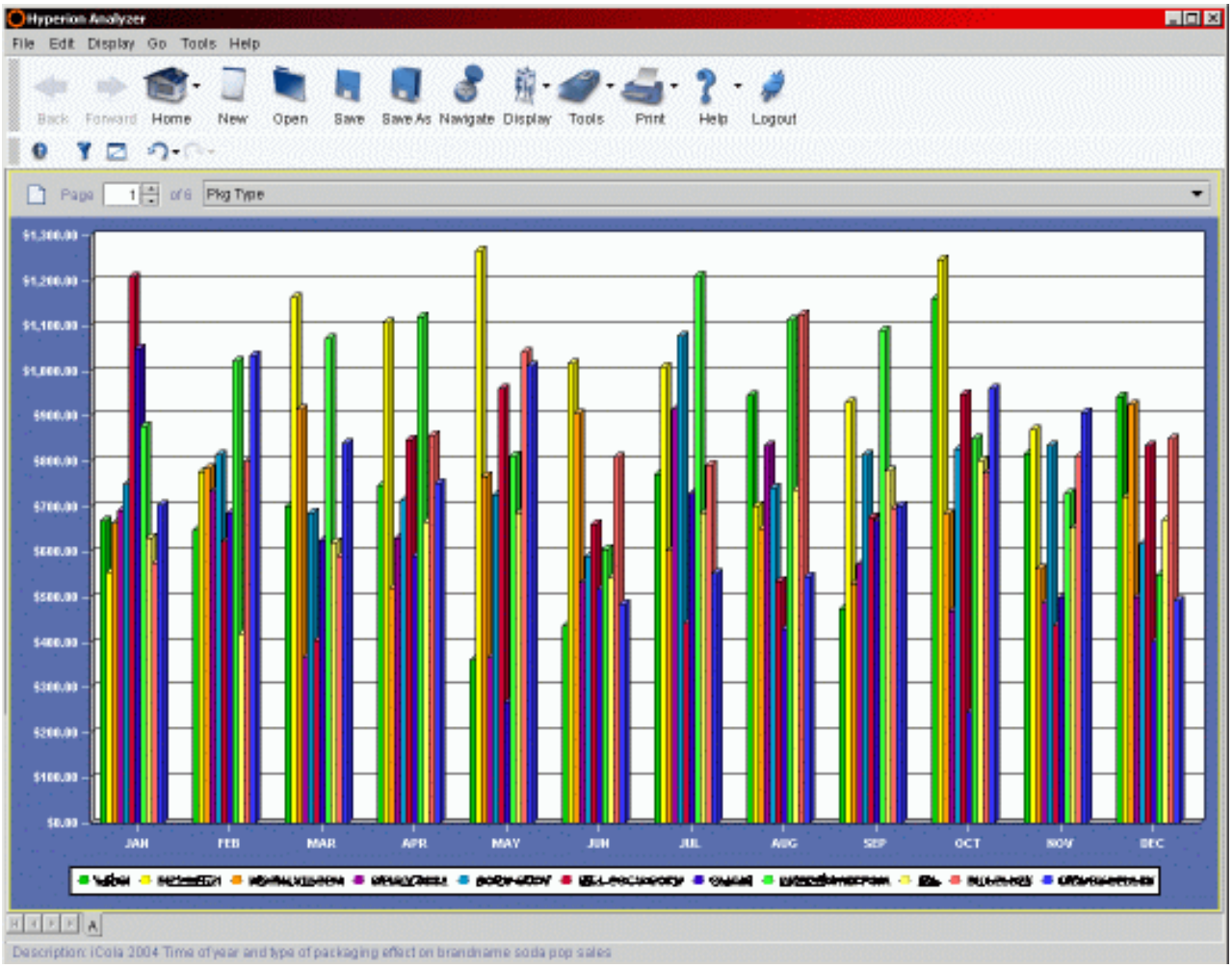

Figura 14: Cliente Web baseado em Java

Tendo em conta os objectivos do cenário do iCola, foi utilizado o formato de estilo de gráfico nos relatórios, ainda que existam diversas outras capacidades para relatórios de combinação (folhas de cálculo e gráficos) e folhas de cálculo. A finalidade da inclusão do produto Analyzer no cenário do iCola não consiste essencialmente em testar exaustivamente o próprio produto Analyzer, mas em contrapartida, consiste em utilizar os cubos OLAP e funcionar como um comprovativo conceptual sobre como o produto Analyzer pode ser utilizado.

A aplicação do Analyzer contida no cenário do iCola consiste num dos dois itens directamente utilizados por uma aplicação de volume de trabalho automatizada no ambiente de testes. O cliente Web baseado em HTML (Figura 15) fornece uma excelente interface para o volume de trabalho (o volume de trabalho consiste num script automatizado que simula a acção do utilizador numa interface Web) e com sentido lógico, uma vez que as alterações constantes à concepção e definições dos relatórios não constituem uma situação realista. Em alternativa, as mesmas vistas podem ser acedidas em várias alturas por vários comunicadores de dados, que, por sua vez, disponibilizam estes relatórios ou um resumo dos mesmos aos respectivos gestores ou directores comerciais.

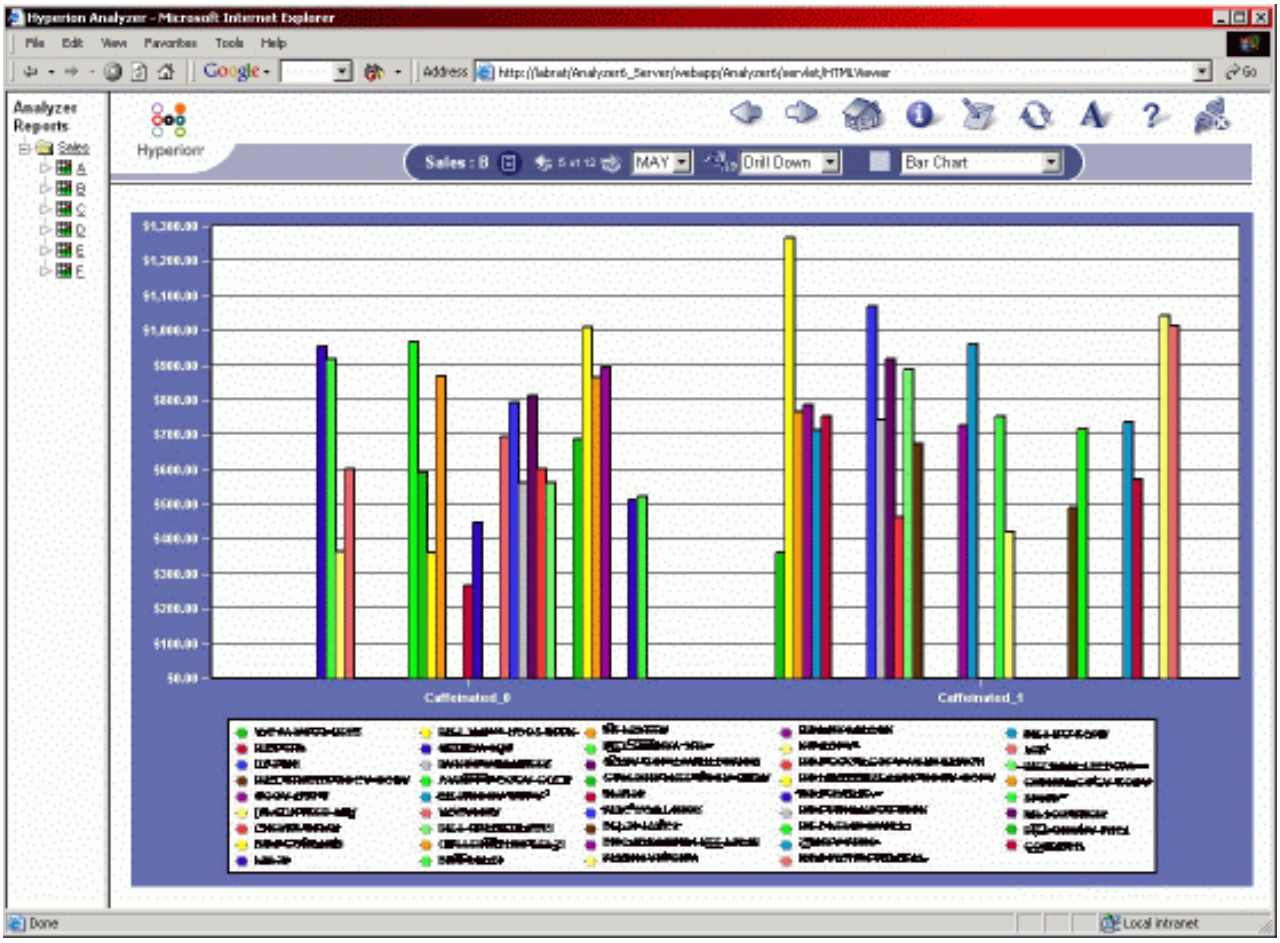

Figura 15: Cliente Web baseado em HTML

Pontos de concepção da aplicação

Não se verificou uma concepção inicial intensa relativamente a esta parte do cenário, uma vez que a utilização do produto foi essencialmente "inovadora", sem grande configuração adicional. No IBM(R) DB2 OLAP Server(TM) Analyzer, estão disponíveis várias personalizações que podem ser aplicadas utilizando a API Java(TM) fornecida, no entanto, tal não é necessário uma vez que as necessidades básicas do cenário são preenchidas utilizando o produto base existente. Após a aplicação de servidor do Analyzer ter sido instalada, foi necessário executar alguns passos de configuração de menor dimensão para que aplicação comunicasse correctamente com o OLAP Server. Estes passos são tratados posteriormente, quando a instalação da aplicação do Analyzer for debatida.

Uma vez que o próprio Analyzer não requereu a execução de um passo específico no que respeita à concepção no âmbito do cenário, o processo de concepção centra-se, em alternativa, no modo como a aplicação é utilizada como parte das operações diárias do iCola. Uma vez que o principal objectivo do Analyzer consiste em fornecer um mecanismo de comunicação de vários dados comerciais importantes, foi necessário colocar várias questões comerciais com base nos critérios disponíveis. Estas "consultas" do Analyzer foram formuladas tendo em conta as tendências comerciais anteriores e actuais, para que fossem iniciadas actividades e práticas comerciais de melhor qualidade em etapas actuais e futuras do negócio.

Por exemplo, estas consultas podem ser utilizadas para examinar em que meses a quantidade de encomendas de determinadas marcas de refrigerantes teve tendência a baixar e, deste modo, essas informações podem ser utilizadas para criar um incentivo na atribuição de preços junto dos compradores

durante esses meses. Num exemplo utilizando a partilha de dados, estes podem ser utilizados para iniciar uma campanha através de cupões durante meses de grande volume com um parceiro, como, por exemplo, um fabricante/distribuidor de batatas fritas para ajudar a impulsionar as vendas de ambas as partes. Os dados reunidos pela empresa iCola também podem ser reunidos em formato de relatório e vendidos aos próprios fabricantes reais de bebidas. Deste modo, os fabricantes podem também tomar decisões comerciais baseadas na experiência sobre os respectivos produtos.

As seguintes questões são actualmente executadas através dos dados do cubo OLAP utilizando o produto Analyzer:

- De que forma a época do ano e o tipo de embalagem afectam as vendas de refrigerantes de marcas populares
- Quais os melhores e os piores tipos de vendas de marcas de refrigerantes no ano anterior, por mês, separando os produtos com cafeína e sem cafeína
- Quais os estabelecimentos populares, por região, que venderam a maior e a menor quantidade de cerveja, por marca, no corrente ano
- De que forma se pode comparar a competitividade nas vendas entre colas no corrente ano; Marca A vs Marca B
- De que forma se pode comparar a competitividade nas vendas entre cervejas no corrente ano; Marca C vs Marca D
- Qual a previsão global de lucro para o corrente ano, trimestre e mês

Actualmente, a concepção também permite que a vista seja alterada de forma dinâmica com base num ano seleccionado para que qualquer destas consultas possa ser direccionada para quaisquer dados do cubo referentes a anos anteriores, tendo em vista uma análise de tendências mais aprofundada. Existem também outras questões importantes que podem ser utilizadas para analisar os dados da base de dados e posteriormente, à medida que o cenário se for desenvolvendo e expandindo, serão adicionadas outras consultas.

Instalação da aplicação

O IBM(R) DB2(R) OLAP Analyzer 8.1 fornece fornece documentação de instalação para o Windows(R) e para o AIX(R), no entanto, após utilizá-la, a avaliação global revelou que as informações são de difícil compreensão. Surgiram questões durante o processo de instalação, algumas das quais não tiveram resposta na documentação nem na Web.

Inicialmente, foi efectuada uma tentativa para instalar a aplicação num servidor IBM pSeries(R) com o AIX(R). No entanto, após contactar o suporte técnico, detectou-se que a utilização do DB2 OLAP Server(TM) Analyzer no AIX para aceder ao DB2 OLAP Server no iSeries(TM) V5R2 não é suportada. Deste modo, estabeleceu-se que a utilização da versão do Windows do DB2 OLAP Server Analyzer em execução num servidor IBM xSeries(R) seria uma opção mais adequada.

Segue-se um resumo de alto nível do processo de instalação necessário para instalar o IBM DB2 OLAP Server Analyzer 8.1 para a plataforma Windows XP Professional no IBM xSeries:

1. Obtenha o CD-ROM do IBM DB2 OLAP Server Analyzer 8.1 for Windows.
2. Verifique se o servidor xSeries que está a ser utilizado tem uma versão compatível do sistema operativo Windows. Verifique as recomendações do hardware para o DB2 OLAP Server Analyzer de modo a assegurar que será executado correctamente no servidor xSeries.
3. Ligue o xSeries (caso ainda não esteja ligado) e inicie sessão no Windows utilizando uma conta administrativa.
4. Abra Control Panel (Painel de controlo) do Windows e procure o Java(TM) Plugin 1.3.0_02.

5. Se o Java Plugin 1.3.0_02 não estiver instalado, instale o mesmo agora utilizando o programa de instalação que se encontra no CD-ROM do Analyzer (utilize o ficheiro .exe existente no directório Java_Plugin_1_3_0_02). Execute a instalação do suplemento Java antes de avançar para o próximo passo.
6. Verifique a instalação do Java Plugin 1.3.0_02 abrindo de novo Control Panel (Painel de controlo) do Windows.
7. Utilizando o CD-ROM do Analyzer, instale o DB2 Personal Edition versão 7.2 no sistema. Este ficheiro encontra-se no directório DB2 no CD-ROM. Durante a instalação, escolha todas as opções assumidas; excepto o preenchimento do selector de confirmação correspondente a "Admin Client" e utilize db2admin e "password" para todos os IDs de utilizador e palavras-passe. Os IDs de utilizador e palavras-passe podem ser alterados posteriormente se necessário. Execute a instalação de DB2 antes de avançar para o próximo passo.
8. Reinicialize o xSeries e inicie novamente sessão no Windows utilizando a conta administrativa previamente utilizada.
9. Pare todos os serviços do DB2 actualmente em execução. Para efectuar este procedimento, abra Control Panel (Painel de controlo) do Windows e seleccione **Administrative Tools** (Ferramentas administrativas), em seguida, seleccione **Services** (Serviços) e pare os três serviços do DB2 actualmente em execução. Deixe esta janela aberta.
10. Abra uma linha de comandos do DOS e execute o ficheiro **usejdbc2.bat** que se encontra no directório SQLLIB\java12 (geralmente C:\SQLLIB\java12). Quando a execução deste ficheiro tiver sido concluída, feche a linha de comandos do DOS.
11. Regresse à janela **Services** (Serviços) previamente aberta. Inicie todos os serviços do DB2 previamente em execução (Antes do passo 9 ser executado).
12. Crie a base de dados ANALYZ60 utilizando o assistente **Create DB** (Criar BD) na aplicação DB2 Control Center. No menu Start (Iniciar), aceda à pasta do programa IBM DB2 e seleccione **Control Center** (Centro de controlo). Na aplicação DB2 Control Center, navegue para o nome do sistema, em seguida, para **Instance** (Instância-objecto), **DB2** e **Databases** (Bases de dados). Faça clique com o botão direito do rato na pasta **Databases** (Bases de dados) e seleccione Create -> Database Using Wizard (Criar -> Base de dados utilizando o assistente). Escreva ANALYZ60 como nome alternativo e nome da base de dados e, em seguida, faça clique no botão **Next** (Seguinte). Nos restantes passos do assistente, aceite apenas as opções assumidas (Faça clique no botão **Next** [Seguinte] até ser apresentada a página Summary [Resumo] e, em seguida, faça clique em **Finish** [Terminar]). Aguarde a conclusão da criação da base de dados. Quando a base de dados estiver concluída, saia da aplicação DB2 Control Center.
13. Instale o WebSphere(R) Application Server 3.5 a partir do CD-ROM do Analyzer (localizado no CD no directório WebSphere35/NT). Aceite todas as opções assumidas, excepto no ecrã da base de dados. Ao escolher a base de dados, seleccione **InstantDB** (BD instantânea) como o tipo de base de dados a ser utilizado pelo WebSphere para que o processo de instalação funcione correctamente. Continue a selecção de valores assumidos até ser apresentado um ecrã de resumo e, em seguida, conclua o processo de instalação. Aguarde até a instalação estar concluída antes de avançar para o próximo passo.
14. Reinicialize o xSeries e inicie novamente sessão no Windows utilizando a conta administrativa previamente utilizada.
15. Instale o WebSphere Application Server 3.5 Fixpak #5. Este pacote de correcções altera o nível da aplicação do WebSphere de 3.5 para 3.5.5, necessário para que o Analyzer seja correctamente instalado. O ficheiro do pacote de correcções pode ser encontrado no directório raiz do CD-ROM do Analyzer (X:\was35_std_ptf_5.zip). Descompacte este ficheiro num directório local no PC e, em seguida, execute o ficheiro de instalação extraído. Escreva os parâmetros à medida que o pacote de correcções solicitar os mesmos. Quando lhe for solicitada a actualização do IBM HTTP Server (IHS), introduza a localização do directório do IHS no sistema local para actualizar esta aplicação.
16. Reinicialize o xSeries e inicie novamente sessão no Windows utilizando a conta administrativa previamente utilizada.

17. Instale o produto IBM DB2 OLAP Server Analyzer a partir do CD_ROM do DB2 OLAP Server Analyzer. Aceite todos os valores assumidos a medida que são apresentados (desde que correspondam aos programas previamente instalados, como, por exemplo, o WebSphere e o DB2 no xSeries). O único desvio face aos valores assumidos consiste em seleccionar o nome de utilizador RDBMS para **db2admin**, em vez do valor assumido, **Analyzer**.
18. Por fim, para começar a utilizar o IBM DB2 OLAP Server Analyzer aceda ao URL (http://sistemacentrallocal/Analyzer6_Server/webapp/Analyzer6/index.html: substitua **sistemacentrallocal** pelo nome de sistema central para utilização deste produto remotamente).

Observações chave

A observação mais importante a salientar durante todo o processo ao trabalhar com o IBM(R) DB2 OLAP Server(TM) Analyzer 8.1 consiste na dificuldade de instalação da aplicação. Esta situação deve-se essencialmente ao facto de as instruções de instalação fornecidas com a documentação do Analyzer não serem exactas. As instruções de instalação fornecidas na secção ″Instalar a aplicação″ deste capítulo foram de grande valia durante o processo de instalação. Quando o produto foi instalado, revelou-se como uma aplicação bastante útil e bem estruturada adequada às necessidades de Business Intelligence do cenário do iCola.

Os testes automatizados de volume de trabalho que simularam os acessos de utilizador à aplicação do Analyzer foram executados concorrentemente com o restante cenário do iCola para fazer uso dos cubos OLAP do programa emissor bem como para fazer uso da aplicação Web do Analyzer. Aparentemente, quando foi executado um teste automatizado de volume de trabalho de utilizador único na aplicação, os resultados foram menos favoráveis, se efectuada a comparação com o acesso manual à aplicação por um único indivíduo. No entanto, é possível que outros factores tenham afectado o desempenho durante uma execução de teste de volume de trabalho.

O Windows(R) XP Professional foi o sistema operativo instalado na caixa xSeries(R), embora os testes anteriores efectuados no Windows 2000 tenham decorrido normalmente. Outra observação em relação ao desempenho: os processos Java(TM) utilizados pela aplicação do Analyzer foram adaptados para serem executados com prioridade em tempo real; esta alteração pode ser efectuada no utilitário Task Manager (Gestor de tarefas) do Windows. Esta alteração foi efectuada para melhorar o desempenho da aplicação do Analyzer e, uma vez que se trata do único processo real necessário para ser executado no sistema, ao ser-lhe atribuída a prioridade mais alta não afecta outras aplicações no xSeries. Se o xSeries a ser utilizado for um servidor multi-funcional e não for dedicado apenas à execução do Analyzer, a alteração da prioridade de tempo de execução poderá não ser a melhor opção por motivos de desempenho.

Uma das descobertas chave resultantes da utilização do Analyzer reside no facto de os próprios dados de Business Intelligence poderem ser utilizados também como um elemento de venda lucrativo. Os dados e relatórios reunidos no seguimento da utilização do Analyzer poderiam ter uma utilização diferente e não servirem apenas como dados de Business Intelligence destinados à empresa iCola; os dados alvo específicos destinados a um fabricante de bebidas em particular poderiam ser vendidos na qualidade de produto ao fabricante de bebidas real para respectiva utilização a nível comercial. Além disso, alguns estabelecimentos poderão nem sempre manter o registo de todas as tendências comerciais nas vendas de determinados produtos na respectiva companhia. Nas empresas deste tipo, poderá ser vendido um “sector” do cubo como propriedade intelectual que a própria companhia poderá utilizar para aumentar as vendas face à concorrência em função dos resultados da sua própria análise. Mais uma vez o Analyzer desempenharia um papel muito importante nas decisões comerciais de outras empresas aplicando os dados geridos pelo DB2 OLAP Server à empresa de distribuição de bebidas do iCola.

Na generalidade, o IBM DB2 OLAP Server Analyzer 8.1 constitui um elemento valioso tendo em vista as finalidades do cenário do iCola, apesar de ter sido mais difícil de instalar do que o previsto. O produto em si colocar algumas dificuldades na respectiva aprendizagem, mas torna-se de fácil utilização uma vez correctamente entendido. Quando estiver familiarizado com o funcionamento da aplicação, é muito provável que o produto tenha um impacto positivo no modo como os dados de Business Intelligence são fornecidos às entidades com poder de decisão num cenário comercial. Para qualquer indivíduo que

utilize o DB2 OLAP Server, a interface Web do DB2 OLAP Server Analyzer é um elemento indispensável, no que respeita a uma análise com êxito dos dados de Business Intelligence reunidos utilizando o DB2(R) OLAP Server.

Capítulo 6. Ideias para o futuro

À medida que as entidades com poder de decisão, financeiro e estratégico se familiarizam com o tipo de informações obtidas de Business Intelligence os pedidos aumentarão. Fases futuras do projecto de Business Intelligence envolverão a criação de mais cubos multidimensionais para análise de dados. Além disso, existem planos para beneficiar das funcionalidades de exploração de dados incluídas no DB2 OLAP Server(TM) de modo a identificar padrões que ajudarão as entidades com poder de decisão a planear estratégias para os seus negócios.

Além das vantagens das funcionalidades avançadas do DB2 OLAP Server e do endereçamento de pedidos de dados de entidades com poder de decisão e estratégico, o cenário do iCola pretende beneficiar de novas funcionalidades da base de dados do iSeries(TM), como tabelas de consulta materializadas.

À medida que a empresa expande e as origens de dados diversificam, a necessidade de expandir o repositório de dados para um armazém de dados poderá tornar-se evidente. Se surgir a necessidade de um armazém de dados, será necessário analisar as ferramentas disponíveis que poderão auxiliar nos processos de extracção, transformação e carregamento mais complexos. Além disso, um único armazém de dados poderá preencher vários repositórios de dados orientados por assunto.

Serão também revistas ferramentas adicionais para fornecerem mais valores a partir dos dados. Serão pesquisadas mais ferramentas e técnicas avançadas para exploração de dados e tomada de decisões.

Capítulo 7. Apêndice A

Instruções de SQL utilizadas para criar o repositório de dados, ICOLADM

A seguinte instrução de SQL é utilizada para criar a biblioteca ICOLADM:

```
CREATE SCHEMA ICOLADM ;
```

As seguintes instruções de SQL são utilizadas para criar as tabelas de dimensão e a tabela de factos ICOLADM:

```
CREATE TABLE ICOLADM.CUSTOMER (
BRANCHID INTEGER NOT NULL ,
BRANCHNAME CHAR(25) CCSID 37 NOT NULL ,
STOREID INTEGER NOT NULL ,
STORENAME CHAR(25) CCSID 37 NOT NULL ,
CONSTRAINT ICOLADM.QSYS_CUSTOMER_00001 PRIMARY KEY( BRANCHID ) ) ;

CREATE TABLE ICOLADM.FACTTABLE (
ORDERID INTEGER NOT NULL ,
UPC INTEGER NOT NULL ,
MARKETID INTEGER NOT NULL ,
TIMEID INTEGER NOT NULL ,
QUANTITYSOLD FOR COLUMN QUANT00001 INTEGER NOT NULL ,
BRANCHID INTEGER NOT NULL ,
SELLINGPRICE FOR COLUMN SELLI00001 DECIMAL(9, 2) NOT NULL ,
TOTALPRICE DECIMAL(9, 2) NOT NULL ,
COSTTOBUY DECIMAL(9, 2) NOT NULL )
CONSTRAINT ICOLADM.QSYS_FACTTABLE_ORDERID_00001 PRIMARY KEY( ORDERID) ;

CREATE TABLE ICOLADM.MARKET (
MARKETID INTEGER GENERATED ALWAYS AS IDENTITY (
START WITH 1 INCREMENT BY 1
NO MINVALUE NO MAXVALUE
NO CYCLE NO ORDER
CACHE 20 ) ,
REGION CHAR(20) CCSID 37 NOT NULL ,
STATE CHAR(2) CCSID 37 NOT NULL ,
CITY CHAR(20) CCSID 37 NOT NULL ,
COUNTRY CHAR(2) CCSID 37 NOT NULL ,
POPULATION INTEGER NOT NULL ,
CONSTRAINT ICOLADM.QSYS_MARKET_00001 PRIMARY KEY( MARKETID ) ) ;

CREATE TABLE ICOLADM.PRODUCT (
UPC INTEGER NOT NULL ,
SIZE INTEGER NOT NULL ,
CAFFEINATED FOR COLUMN CAFFE00001 DECIMAL(1, 0) NOT NULL ,
ALCOHOLIC DECIMAL(1, 0) NOT NULL ,
INTRODATE DATE NOT NULL ,
DISCONTINUEDATE FOR COLUMN DISCO00001 DATE DEFAULT NULL ,
PACKAGETYPE FOR COLUMN PACKA00001 CHAR(13) CCSID 37 NOT NULL DEFAULT '' ,
NAME CHAR(80) CCSID 37 NOT NULL DEFAULT 'No default' ,
```

```
BRAND CHAR(40) CCSID 37 NOT NULL DEFAULT 'No default' ,
CONSTRAINT ICOLADM.Q_ICOLADM_PRODUCT_UPC_00002 PRIMARY KEY( UPC ) ) ;

CREATE TABLE ICOLADM.TIME (
TIMEID INTEGER NOT NULL ,
"DIA" SMALLINT NOT NULL ,
"M╩S" CHAR(3) CCSID 37 DEFAULT NULL ,
"ANO" SMALLINT NOT NULL ,
QUARTER CHAR(4) CCSID 37 DEFAULT NULL ,
SEASON CHAR(6) CCSID 37 DEFAULT NULL ,
DAYOFWEEK CHAR(9) CCSID 37 DEFAULT NULL ,
CONSTRAINT ICOLADM.QSYS_TIME_00001 PRIMARY KEY( TIMEID ) ) ;
```

As seguintes instruções de SQL são utilizadas para criar as tabelas auxiliares ICOLADM ETLPROCESS (utilizada para registo) e REGIONS (utilizada para preencher a tabela MARKET):

```
CREATE TABLE ICOLADM.ETLPROCESS (
START TIMESTAMP NOT NULL ,
FINISH TIMESTAMP DEFAULT NULL ) ;

CREATE TABLE ICOLADM.REGIONS (
STATE CHAR(2) CCSID 37 NOT NULL ,
REGION CHAR(15) CCSID 37 NOT NULL ,
COUNTRY CHAR(2) CCSID 37 NOT NULL DEFAULT 'US' ,
CONSTRAINT ICOLADM.QSYS_REGIOINS_00001 PRIMARY KEY( STATE ) ) ;
```

Capítulo 8. Apêndice B

Procedimentos de SQL armazenados para o processo ETL

É apresentado, em seguida, o código utilizado pelo iCola para criar cada um dos procedimentos de SQL armazenados.

O procedimento ETL chama todos os outros procedimentos de SQL:

```
CREATE PROCEDURE ICOLA.ETL ( )
LANGUAGE SQL
SPECIFIC ICOLA.ETL
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
CALL ICOLA . STARTETL ;
CALL ICOLA . CUSTOMER ;
CALL ICOLA . MARKET ;
CALL ICOLA . PRODUCT ;
CALL ICOLA . FACTTABLE ;
CALL ICOLA . ENDETL ;
END;
```

Os procedimentos STARTETL e ENDETL registam as informações de hora gravando o CURRENT TIMESTAMP:

```
CREATE PROCEDURE ICOLA.STARTETL ( )
LANGUAGE SQL
SPECIFIC ICOLA.STARTETL
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . ETLPROCESS ( START ) VALUES ( CURRENT TIMESTAMP ) ;
END ;

CREATE PROCEDURE ICOLA.ENDETL ( )
LANGUAGE SQL
SPECIFIC ICOLA.ENDETL
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
UPDATE ICOLADM . ETLPROCESS SET FINISH = ( CURRENT TIMESTAMP )
WHERE START = ( SELECT MAX ( START ) FROM ICOLADM . ETLPROCESS ) ;
END ;
```

Os procedimentos CUSTOMER, MARKET e PRODUCT inserem informações nas respectivas tabelas de repositório de dados. Estas versões dos procedimentos (que utilizam a palavra-chave EXCEPT) só podem ser utilizadas no OS/400 V5R3:

```
CREATE PROCEDURE ICOLA.CUSTOMER ( )
LANGUAGE SQL
SPECIFIC ICOLA.CUSTOMER
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . CUSTOMER ( BRANCHID , BRANCHNAME , STOREID , STORENAME )
( ( SELECT ICOLA . CUSTOMERBRANCH . BRANCHID , ICOLA . CUSTOMERBRANCH .
BRANCHNAME , ICOLA . CUSTOMERBRANCH . STOREID , ICOLA . STORE . BUSINESSNAME
FROM ICOLA . CUSTOMERBRANCH , ICOLA . STORE
WHERE ICOLA . CUSTOMERBRANCH . STOREID = ICOLA . STORE . STOREID )
EXCEPT
( SELECT BRANCHID , BRANCHNAME , STOREID , STORENAME FROM ICOLADM . CUSTOMER ) )
;
END ;

CREATE PROCEDURE ICOLA.MARKET ( )
LANGUAGE SQL
SPECIFIC ICOLA.MARKET
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . MARKET ( REGION , STATE , CITY , COUNTRY , POPULATION )
( ( SELECT ICOLADM . REGIONS . REGION , ICOLA . CUSTOMERBRANCH . STATE, ICOLA .
CUSTOMERBRANCH . CITY , ICOLADM . REGIONS . COUNTRY
FROM ICOLA . CUSTOMERBRANCH INNER JOIN ICOLADM . REGIONS ON ICOLA .
CUSTOMERBRANCH . STATE = ICOLADM . REGIONS . STATE )
EXCEPT
( SELECT REGION , STATE , CITY , COUNTRY , 0 FROM ICOLADM . MARKET ) ) ;
END ;

CREATE PROCEDURE ICOLA.PRODUCT ( )
LANGUAGE SQL
SPECIFIC ICOLA.PRODUCT
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . PRODUCT ( UPC , NAME , BRAND , SIZE , CAFFEINATED ,
ALCOHOLIC , PACKAGETYPE , INTRODATE )
( ( SELECT UPC , NAME , BRAND , SIZE , CAFFEINATED , ALCOHOLIC , PACKAGETYPE,
INTRODATE
FROM ICOLA . PRODUCT )
EXCEPT
( SELECT UPC , NAME , BRAND , SIZE , CAFFEINATED , ALCOHOLIC , PACKAGETYPE,
INTRODATE FROM ICOLADM . PRODUCT ) ) ;
UPDATE ICOLADM . PRODUCT SET DISCONTINUEDATE =
( SELECT DISCONTINUEDATE FROM ICOLA . PRODUCT
WHERE ICOLADM . PRODUCT . UPC = ICOLA . PRODUCT . UPC
AND ICOLA . PRODUCT . DISCONTINUEDATE IS NOT NULL ) ;
END ;
```

Estas versões dos procedimentos CUSTOMER, MARKET e PRODUCT podem ser utilizadas no OS/400 V5R2 e V5R3:

```
CREATE PROCEDURE ICOLA.CUSTOMER ( )
LANGUAGE SQL
SPECIFIC ICOLA.CUSTOMER
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . CUSTOMER (BRANCHID , BRANCHNAME , STOREID , STORENAME )
(SELECT ICOLA.CUSTOMERBRANCH.BRANCHID, ICOLA.CUSTOMERBRANCH.BRANCHNAME,
ICOLA .CUSTOMERBRANCH.STOREID, ICOLA.STORE. BUSINESSNAME
FROM ICOLA.CUSTOMERBRANCH, ICOLA.STORE
WHERE ICOLA.CUSTOMERBRANCH.STOREID = ICOLA.STORE.STOREID
AND ICOLA.CUSTOMERBRANCH.BRANCHID NOT IN
(SELECT ICOLADM.CUSTOMER.BRANCHID FROM ICOLADM.CUSTOMER)) ;
END ;

CREATE PROCEDURE ICOLA.MARKET ( )
LANGUAGE SQL
SPECIFIC ICOLA.MARKET
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . MARKET ( REGION , STATE , CITY , COUNTRY , POPULATION )
(SELECT DISTINCT ICOLADM.REGIONS.REGION, ICOLA.CUSTOMERBRANCH.STATE ,
ICOLA.CUSTOMERBRANCH.CITY, ICOLADM.REGIONS.COUNTRY, 0
FROM ICOLA.CUSTOMERBRANCH, ICOLADM.REGIONS, ICOLADM.MARKET
WHERE CONCAT(RTRIM(ICOLA.CUSTOMERBRANCH.CITY), CONCAT(' ',
ICOLA.CUSTOMERBRANCH.STATE)) NOT IN
(SELECT CONCAT(RTRIM(ICOLADM.MARKET.CITY), CONCAT(' ', ICOLADM.MARKET.STATE))
FROM ICOLADM.MARKET)
AND ICOLA.CUSTOMERBRANCH.STATE = ICOLADM.REGIONS.STATE ) ;
END ;

CREATE PROCEDURE ICOLA.PRODUCT ( )
LANGUAGE SQL
SPECIFIC ICOLA.PRODUCT
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN
INSERT INTO ICOLADM . PRODUCT ( UPC , NAME , BRAND , SIZE , CAFFEINATED ,
ALCOHOLIC , PACKAGETYPE , INTRODATE )
(SELECT UPC, NAME, BRAND, SIZE, CAFFEINATED, ALCOHOLIC, PACKAGETYPE, INTRODATE
FROM ICOLA.PRODUCT
WHERE UPC NOT IN (SELECT UPC FROM ICOLADM.PRODUCT) ) ;
UPDATE ICOLADM . PRODUCT SET DISCONTINUEDATE =
( SELECT DISCONTINUEDATE FROM ICOLA . PRODUCT
WHERE ICOLADM . PRODUCT . UPC = ICOLA . PRODUCT . UPC
AND ICOLA . PRODUCT . DISCONTINUEDATE IS NOT NULL ) ;
END ;
```

O procedimento FACTTABLE insere informações na tabela FACTTABLE do repositório de dados. O procedimento FACTTABLE retira informações de encomenda da tabela ORDER de dados operacionais, com a ajuda de outras tabelas de dados operacionais para preencher detalhes adicionais. Esta versão do procedimento FACTTABLE (que utiliza a palavra-chave EXCEPT) só pode ser utilizada no OS/400 V5R3:

```
CREATE PROCEDURE ICOLA.FACTTABLE ( )
LANGUAGE SQL
SPECIFIC ICOLA.FACTTABLE
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN INSERT INTO ICOLADM . FACTTABLE ( ORDERID , UPC , MARKETID , TIMEID,
QUANTITYSOLD , SELLINGPRICE , TOTALPRICE , COSTTOBUY , BRANCHID )
( ( SELECT ICOLA . ORDER . ORDERID , ICOLA . ORDER . UPC , ICOLADM . MARKET . MARKETID
, ( ( YEAR ( ICOLA . ORDER . DATE ) * 10000 ) + ( MONTH ( ICOLA . ORDER . DATE ) * 100 ) + DAY (
ICOLA . ORDER . DATE ) ) , ICOLA . ORDER . QUANTITY , TRUNC ( ICOLA . ORDER . SUBTOTAL /
ICOLA . ORDER . QUANTITY, 2 ) , ICOLA . ORDER . SUBTOTAL , ICOLA . PRICE . OURCOST , ICOLA
. CUSTOMERCONTACT . BRANCHID
FROM ICOLA . ORDER , ICOLA . PRODUCT , ICOLA . PRICE , ICOLA . CUSTOMERCONTACT,
ICOLA . CUSTOMERBRANCH , ICOLADM . MARKET
WHERE ICOLA . ORDER . CUSTOMERID = ICOLA . CUSTOMERCONTACT . CUSTOMERID
AND ICOLA . CUSTOMERCONTACT . BRANCHID = ICOLA . CUSTOMERBRANCH . BRANCHID
AND ICOLA . CUSTOMERBRANCH . CITY = ICOLADM . MARKET . CITY
AND ICOLA . CUSTOMERBRANCH . STATE = ICOLADM . MARKET . STATE
AND ICOLA . ORDER . UPC = ICOLA . PRODUCT . UPC AND ICOLA . PRODUCT . PRICECODE =
ICOLA . PRICE . PRICECODE
AND ICOLA.ORDER.DATE < CURRENT DATE)
EXCEPT
( SELECT ORDERID , UPC , MARKETID , TIMEID , QUANTITYSOLD , SELLINGPRICE, TOTALPRICE ,
COSTTOBUY , BRANCHID FROM ICOLADM . FACTTABLE ) ) ;
END ;
```

Esta versão do procedimento FACTTABLE pode ser utilizada no OS/400 V5R2 e V5R3:

```
CREATE PROCEDURE ICOLA.FACTTABLE ( )
LANGUAGE SQL
SPECIFIC ICOLA.FACTTABLE
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
BEGIN INSERT INTO ICOLADM . FACTTABLE ( ORDERID , UPC , MARKETID , TIMEID,
QUANTITYSOLD , SELLINGPRICE , TOTALPRICE , COSTTOBUY , BRANCHID )
(SELECT ICOLA.ORDER.ORDERID, ICOLA.ORDER.UPC , ICOLADM.MARKET.MARKETID,
((YEAR(ICOLA.ORDER.DATE) * 10000) + (MONTH(ICOLA.ORDER.DATE) * 100) +
DAY(ICOLA.ORDER.DATE)), ICOLA.ORDER.QUANTITY,
TRUNC(ICOLA.ORDER.SUBTOTAL/ICOLA.ORDER.QUANTITY, 2), ICOLA.ORDER.SUBTOTAL,
ICOLA.PRICE.OURCOST, ICOLA.CUSTOMERCONTACT.BRANCHID
FROM ICOLA.ORDER, ICOLA.PRODUCT, ICOLA.PRICE, ICOLA.CUSTOMERCONTACT,
ICOLA.CUSTOMERBRANCH, ICOLADM.MARKET
WHERE ICOLA.ORDER.CUSTOMERID = ICOLA.CUSTOMERCONTACT.CUSTOMERID
AND ICOLA.CUSTOMERCONTACT.BRANCHID = ICOLA.CUSTOMERBRANCH.BRANCHID
AND ICOLA.CUSTOMERBRANCH.CITY = ICOLADM.MARKET.CITY
AND ICOLA.CUSTOMERBRANCH.STATE = ICOLADM.MARKET.STATE
AND ICOLA.ORDER.UPC = ICOLA.PRODUCT.UPC AND ICOLA.PRODUCT.PRICECODE =
ICOLA.PRICE.PRICECODE
AND ICOLA.ORDER.DATE < CURRENT DATE
AND ICOLA.ORDER.ORDERID NOT IN (SELECT ORDERID FROM ICOLADM.FACTTABLE)
) ;
END ;
```

Capítulo 9. Apêndice C

Instruções de SQL utilizadas para criar ficheiros de regras

Preenchimento da dimensão MARKET: MRKT-RUL

```
SELECT ICOLADM.MARKET.REGION, ICOLADM.MARKET.STATE,
CONCAT(RTRIM(ICOLADM.MARKET.CITY), CONCAT(' ', ICOLADM.MARKET.STATE))
FROM ICOLADM.MARKET
```

Preenchimento da dimensão PRODUCT: PROD-RUL

```
SELECT ICOLADM.PRODUCT.BRAND, ICOLADM.PRODUCT.NAME,
CONCAT(RTRIM(ICOLADM.PRODUCT.NAME), CONCAT(' ',
CONCAT(CAST(ICOLADM.PRODUCT.SIZE AS CHAR(2)), CONCAT(' OZ ',
RTRIM(ICOLADM.PRODUCT.PACKAGETYPE)))))), ICOLADM.PRODUCT.SIZE,
ICOLADM.PRODUCT.PACKAGETYPE, ICOLADM.PRODUCT.CAFFEINATED,
ICOLADM.PRODUCT.ALCOHOLIC
FROM ICOLADM.PRODUCT
```

Preenchimento da dimensão STORE: STOR-RUL

```
SELECT ICOLADM.CUSTOMER.STORENAME,
CONCAT(RTRIM(ICOLADM.CUSTOMER.STORENAME), CONCAT(' ',
RTRIM(ICOLADM.CUSTOMER.BRANCHNAME)))
FROM ICOLADM.CUSTOMER
```

Consulta utilizada para carregar os dados de despesas: EXP-2003

```
SELECT CONCAT(RTRIM(ICOLADM.CUSTOMER.STORENAME), CONCAT(' ',
RTRIM(ICOLADM.CUSTOMER.BRANCHNAME))), MYTIME.MONTH,
CONCAT(RTRIM(ICOLADM.PRODUCT.NAME), CONCAT(' ',
CONCAT(CAST(ICOLADM.PRODUCT.SIZE AS CHAR(2)), CONCAT(' OZ ',
RTRIM(ICOLADM.PRODUCT.PACKAGETYPE)))))), CONCAT(RTRIM(ICOLADM.MARKET.CITY),
CONCAT(' ', ICOLADM.MARKET.STATE)), 'COGS', SUM(ICOLADM.FACTTABLE.COSTTOBUY *
ICOLADM.FACTTABLE.QUANTITYSOLD)
FROM ICOLADM.FACTTABLE, ICOLADM.CUSTOMER, ICOLADM.TIME AS MYTIME,
ICOLADM.PRODUCT, ICOLADM.MARKET
WHERE ICOLADM.FACTTABLE.BRANCHID = ICOLADM.CUSTOMER.BRANCHID AND
ICOLADM.FACTTABLE.TIMEID = MYTIME.TIMEID AND ICOLADM.FACTTABLE.UPC =
ICOLADM.PRODUCT.UPC AND ICOLADM.FACTTABLE.MARKETID =
ICOLADM.MARKET.MARKETID AND MYTIME.YEAR = 2003
GROUP BY CONCAT(RTRIM(ICOLADM.CUSTOMER.STORENAME), CONCAT(' ',
RTRIM(ICOLADM.CUSTOMER.BRANCHNAME))), MYTIME.MONTH,
CONCAT(RTRIM(ICOLADM.PRODUCT.NAME), CONCAT(' ',
CONCAT(CAST(ICOLADM.PRODUCT.SIZE AS CHAR(2)), CONCAT(' OZ ',
RTRIM(ICOLADM.PRODUCT.PACKAGETYPE)))))), CONCAT(RTRIM(ICOLADM.MARKET.CITY),
CONCAT(' ', ICOLADM.MARKET.STATE))
```

Consulta utilizada para carregar os dados de vendas: SAL-2003

```
SELECT CONCAT(RTRIM(ICOLADM.CUSTOMER.STORENAME), CONCAT(' ',
RTRIM(ICOLADM.CUSTOMER.BRANCHNAME))), MYTIME.MONTH,
```

```
CONCAT(RTRIM(ICOLADM.PRODUCT.NAME), CONCAT(' ',
CONCAT(CAST(ICOLADM.PRODUCT.SIZE AS CHAR(2)), CONCAT(' OZ ',
RTRIM(ICOLADM.PRODUCT.PACKAGETYPE)))))), CONCAT(RTRIM(ICOLADM.MARKET.CITY),
CONCAT(' ', ICOLADM.MARKET.STATE)), 'SALES', SUM(ICOLADM.FACTTABLE.TOTALPRICE)
FROM ICOLADM.FACTTABLE, ICOLADM.CUSTOMER, ICOLADM.TIME AS MYTIME,
ICOLADM.PRODUCT, ICOLADM.MARKET
WHERE ICOLADM.FACTTABLE.BRANCHID = ICOLADM.CUSTOMER.BRANCHID AND
ICOLADM.FACTTABLE.TIMEID = MYTIME.TIMEID AND ICOLADM.FACTTABLE.UPC =
ICOLADM.PRODUCT.UPC AND ICOLADM.FACTTABLE.MARKETID =
ICOLADM.MARKET.MARKETID AND MYTIME.YEAR = 2003
GROUP BY CONCAT(RTRIM(ICOLADM.CUSTOMER.STORENAME), CONCAT(' ',
RTRIM(ICOLADM.CUSTOMER.BRANCHNAME))), MYTIME.MONTH,
CONCAT(RTRIM(ICOLADM.PRODUCT.NAME), CONCAT(' ',
CONCAT(CAST(ICOLADM.PRODUCT.SIZE AS CHAR(2)), CONCAT(' OZ ',
RTRIM(ICOLADM.PRODUCT.PACKAGETYPE)))))), CONCAT(RTRIM(ICOLADM.MARKET.CITY),
CONCAT(' ', ICOLADM.MARKET.STATE))
```

A única alteração que foi efectuada entre as consultas EXP-2003 e EXP-2004 (bem como entre as consultas SAL-2003 e SAL-2004) consistiu em alterar o ano na última cláusula where:

- AND MYTIME.YEAR = 2004

Capítulo 10. Exclusões

As informações são fornecidas “TAL COMO ESTÃO” e sem garantias de qualquer espécie. A menção ou referência a produtos não IBM destina-se apenas a informar e não constitui uma aprovação desses produtos por parte da IBM. O desempenho baseia-se em medições e projecções utilizando os pontos de referência padrão da IBM num ambiente controlado. O débito ou desempenho real que qualquer utilizador obtenha varia em função de considerações, tais como a quantidade de multiprogramação na fila de trabalhos do utilizador, a configuração de E/S, a configuração do armazenamento e o volume de trabalho processado. Deste modo, não pode ser fornecida qualquer garantia de que o utilizador individual obtenha melhorias no débito ou desempenho equivalentes aos rácios aqui indicados.

Capítulo 11. Referências

- Conceber o esquema em estrela

(http://www.ciobriefings.com/whitepapers/starschema.asp)
- IBM(R) DB2 OLAP Server(TM) 8.1 Database Administrators Guide
- IBM DB2 Universal Database Business Intelligence Tutorial

(http://www-306.ibm.com/software/data/db2/db2olap/docs/V71docs/db2tu/frame3.htm#db2tussw)
- Hyperion(R) Analyzer Release 6.1 Installation Guide