



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112016013584-9 B1



(22) Data do Depósito: 18/12/2014

(45) Data de Concessão: 17/01/2023

(54) Título: SISTEMA DE COMPUTAÇÃO E MÉTODO EXECUTADO POR PROCESSADOR

(51) Int.Cl.: G06F 17/30.

(30) Prioridade Unionista: 08/07/2014 US 14/325,642; 20/12/2013 US 61/919,349.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): GREGORY HUGHES; MICHAEL J. COULSON; ALEXANDRE TRISTAN ST -CYR; FAISAL MOHAMOOD; THERESA PALMER-BOROSKI; CLEMENS SZYPERSKI; MARIUS DUMITRU.

(86) Pedido PCT: PCT US2014071004 de 18/12/2014

(87) Publicação PCT: WO 2015/095429 de 25/06/2015

(85) Data do Início da Fase Nacional: 13/06/2016

(57) Resumo: CONSTRUÇÃO DE CONSULTAS PARA EXECUÇÃO ATRAVÉS DE ESTRUTURAS DE DADOS MULTIDIMENSIONAIS. A presente invenção refere-se a várias tecnologias pertencendo à construção de uma consulta para execução através de um cubo. Dados tabulares são apresentados em uma tela de vídeo, onde os dados tabulares representam pelo menos parte de um cubo de dados. É recebida entrada com respeito aos dados tabulares, e em resposta à entrada ser recebida, é construída uma consulta baseada na entrada. A consulta é executada através do cubo de dados, resultando em provisionamento de uma nova tabela.

Relatório Descritivo da Patente de Invenção para
**"SISTEMA DE COMPUTAÇÃO E MÉTODO EXECUTADO POR
PROCESSADOR".**

ANTECEDENTES

[001] Os aplicativos de inteligência comercial (BI) implementados por computador têm sido desenvolvidos para facilitar a descoberta de conhecimento (por exemplo, um fato comercial) que pode ajudar um negócio a alcançar sua meta comercial. Com mais particularidade, um usuário de um aplicativo de inteligência comercial pode formular uma consulta que é para ser executada através de dados pertencendo a um negócio particular. Convencionalmente, os dados pertencendo ao negócio são estruturados como pelo menos uma base de dados relacional que compreende pelo menos uma tabela bidimensional.

[002] Apesar de os aplicativos BI convencionais proporcionarem uma interface adequada para ajudar os usuários ao adquirir conhecimento do negócio quando os dados pertencendo ao negócio são estruturados como bases de dados relacionais, os aplicativos de Inteligência comercial convencionais não são bem adequados para descobrir conhecimento do negócio a partir de uma estrutura de dados multidimensional (por exemplo, um cubo de dados, algumas vezes referido como um "hipercubo"). Em um exemplo, quando dados pertencendo a um negócio são estruturados como um cubo de dados, um usuário que deseja obter conhecimento do negócio por consultar o cubo de dados possui *a priori* conhecimento do conteúdo do cubo. Adicionalmente, o usuário deve estar familiarizado com uma linguagem de consulta que possa ser utilizada para executar consultas através do cubo de dados. Além disso, o usuário deve ter conhecimento de partes e/ou dados do cubo de dados que são de interesse quando formulando a consulta. Assim, o usuário deve formular uma consulta final, a qual não pode resultar em apresentação

de conhecimento do negócio desejado.

SUMÁRIO

[003] O dito a seguir é um breve sumário do assunto que é descrito em maiores detalhes neste documento. Este sumário não é pretendido para ser limitativo quanto ao escopo das concretizações.

[004] Um sistema de computação é descrito neste documento, onde o sistema de computação inclui um processador e uma memória. A memória inclui um aplicativo de inteligência comercial (BI) que é executado pelo processador. O aplicativo BI é configurado para construir uma consulta baseado nas modificações incrementais para versões anteriores da consulta. A consulta é representada como uma sequência de etapas da consulta, cada etapa na sequência das etapas da consulta correspondendo a uma respectiva modificação incremental nas modificações incrementais. O aplicativo BI é adicionalmente configurado para recuperar dados tabulares a partir de um cubo de dados baseado na consulta.

BREVE DESCRIÇÃO DOS DESENHOS

[005] A figura 1 é um diagrama de blocos funcional de um sistema ilustrativo que facilita construir uma consulta para execução através de uma estrutura de dados multidimensional.

[006] A figura 2 é um fluxograma ilustrando uma metodologia ilustrativa para construir uma consulta para execução através de uma estrutura de dados multidimensional.

[007] A figura 3 é um fluxograma ilustrando uma metodologia ilustrativa para refinar uma consulta baseado em uma solicitação para contrair ou expandir atributos de pelo menos uma dimensão de uma estrutura de dados multidimensional.

[008] A figura 4 é um fluxograma que ilustra uma metodologia ilustrativa para mesclar pelo menos duas estruturas de dados multidimensionais para criar uma estrutura de dados mesclada.

[009] As figuras 5 até 16 são interfaces gráficas com o usuário ilustrativas que facilitam construir uma consulta que pode ser executada através de uma estrutura de dados multidimensional.

[0010] A figura 17 é um sistema de computador ilustrativo.

DESCRIÇÃO DETALHADA

[0011] Várias tecnologias pertencendo a construir uma consulta através de uma série de estágios, onde a consulta é configurada para executar através e uma estrutura de dados multidimensional, são agora descritas com referência aos desenhos, onde números de referência iguais são utilizados para fazer referência a elementos iguais por todas as partes. Na descrição seguinte, para propósito de explicação, vários detalhes específicos são expostos de modo a proporcionar um entendimento completo de um ou mais aspectos. Entretanto, pode ser evidente que tal aspecto(s) pode ser praticado sem estes detalhes específicos. Em outros casos, estruturas e dispositivos bem conhecidos são apresentados em forma de diagrama de blocos de modo a facilitar descrever um ou mais aspectos. Adicionalmente, é para ser entendido que funcionalidade que é descrita como sendo realizada por um único componente do sistema pode ser executada por vários componentes. De forma similar, por exemplo, um componente pode ser configurado para executar funcionalidade que é descrita como sendo realizada por vários componentes.

[0012] Além disso, o termo "ou" é pretendido para significar um "ou" inclusivo ao invés do que um "ou" exclusivo. Ou seja, a não ser que de outro modo especificado, ou claro a partir do contexto, a frase "X emprega A ou B" é pretendida para significar qualquer uma das permutações naturais inclusivas. Ou seja, a frase "X emprega A ou B" é satisfeita por qualquer um dos seguintes casos: X emprega A; X emprega B; ou X emprega tanto A como B. Em adição, os artigos "um"

e "uma" como utilizados neste pedido e nas concretizações anexas geralmente devem ser construídos para significar "um ou mais" a não ser que de outro modo especificado ou claro a partir do contexto a ser direcionado para uma forma singular.

[0013] Adicionalmente, como utilizados neste documento, os termos "componentes" e "sistema" são pretendidos para abranger armazenamento de dados legível por computador que é configurado com instruções executáveis por computador que causam que alguma funcionalidade seja executada quando executada por um processador. As instruções executáveis por computador podem incluir uma rotina, uma função, ou coisa parecida. Também é para ser entendido que um componente ou sistema pode estar localizado em um único dispositivo ou distribuído através de vários dispositivos. Adicionalmente, como utilizado neste documento, o termo "ilustrativo" é pretendido para significar servir como uma ilustração ou exemplo de alguma coisa, e não é pretendido para indicar uma preferência.

[0014] Neste documento são descritas várias tecnologias pertencendo à construção de uma consulta que é configurada para ser executada através de uma estrutura de dados multidimensional (por exemplo, um cubo, o qual também pode ser referido como um hipercubo, um cubo de dados, etc.). Um cubo é definido através e um conjunto de tabelas relacionadas, por exemplo, utilizando esquema em estrela ou em flocos de neve que compreende pelo menos uma tabela de fatos e várias tabelas de dimensão (também referidas como "dimensões") relacionadas com a tabela de fatos. As fileiras de fatos (fileiras na tabela de fatos) podem ser agrupadas por "atributos" dimensão (por exemplo, colunas das tabelas de dimensão). "Medidas" são funções agregadas aplicadas em relação às colunas de um conjunto de fileiras de fato (por exemplo, uma função de soma através de valores em um conjunto de fileiras de fato).

[0015] Com mais particularidade, a tabela de fatos de um cubo compreende medidas, métricas, ou fatos de um processo comercial e está localizada no centro de um esquema em estrela ou em flocos de neve e rodeada pelas dimensões. As dimensões proporcionam informação rotulada estruturada para medidas numéricas de outro modo não ordenadas. Por consequência, as dimensões incluem os respectivos elementos de dados individuais que não se sobrepõem. As dimensões tipicamente são utilizadas em filtrar, agrupar e rotular dados. O termo "fracionamento" se refere a filtrar dados a partir do cubo, enquanto o termo "cortar em cubos" se refere a agrupar dados no cubo. Frequentemente, as dimensões possuem atributos dimensão, os quais são organizados hierarquicamente. Por exemplo, a dimensão pode representar tempo, com vários atributos hierárquicos possíveis. Por exemplo, a dimensão pode incluir os atributos dimensão "dias", "semanas", "meses", e "anos". O atributo "dias" pode ser agrupado (contraído) em "meses", o qual pode ser contraído em "anos". De forma similar, dias podem ser contraídos para "semanas", as quais podem ser contraídas para "anos", etc. Finalmente, uma medida do cubo é uma propriedade através da qual cálculos podem ser feitos, onde tais cálculos incluem soma, contagem, média, mínimo e máximo.

[0016] Os dados do cubo podem ser representados em uma única tabela plana que compreende atributos e aplicações de medida – esta tabela plana pode ser referida como uma "tabela de fatos". As operações do cubo são "reduzidas para uma representação expressa em termos de operadores relacionais. Uma "tabela de dimensões" se refere a uma tabela que inclui uma coluna para cada atributo da dimensão. Os membros de cada atributo dimensão podem ser enumerados, produzindo um produto vetorial de membros dos atributos de uma dimensão. Vários exemplos se relacionando com cubos serão expostos neste documento.

[0017] Com referência agora à figura 1, um sistema ilustrativo 100 que facilita construir uma consulta que pode ser executada através de uma estrutura de dados multidimensional (referida neste documento como um cubo) é ilustrado. Adicionalmente, o sistema 100 pode facilitar construir a consulta em uma abordagem em gradual, onde a consulta pode ser refinada à medida que um usuário (ou dispositivo de computação) é apresentado aos dados, de modo que a consulta é refinada baseado na análise de um usuário dos dados apresentados ou na análise de um computador dos dados apresentados. O sistema 100 inclui um depósito de dados 102 que compreende um cubo 104. Apesar de o cubo 104 ser apresentado como estando incluído no depósito de dados 102, é para ser entendido que o cubo 104 pode ser distribuído através de vários depósitos de dados. Além disso, o cubo 104 pode representar uma combinação de vários cubos potencialmente possuindo diferentes respectivas estruturas.

[0018] O sistema 100 pode adicionalmente compreender um dispositivo de computação servidor 106 que é configurado com instruções executáveis por computador que permitem que o dispositivo de computação servidor 106 execute consultas através do cubo 104 e emita dados em resposta a executar as consultas através do cubo.

[0019] O sistema 100 pode adicionalmente incluir um dispositivo de computação cliente 108 que pode ser configurado para receber e/ou construir uma consulta para execução através do cubo 104 baseada na entrada a partir de um usuário 110 ou do programa de computador. O dispositivo de computação cliente 108 está em comunicação com o dispositivo de computação servidor 106, e pode transmitir a consulta para o dispositivo de computador servidor 106. O dispositivo de computação cliente 108 pode ser qualquer dispositivo de computação adequado, incluindo, mas não limitado a um dispositivo de computação de mesa, um dispositivo de computação laptop, um

dispositivo de computação tablet (lousa), um telefone celular, um dispositivo de computação convertível, um dispositivo de computação portátil (por exemplo, um relógio de pulso, acessório para cabeça, dentre outros), um fablet, um console de jogo de vídeo, dentre outros.

[0020] O dispositivo de computação cliente 108 compreende um processador 111 e uma memória 112, onde o processador 111 pode executar instruções na memória 112. Como apresentado, a memória 112 inclui um aplicativo de inteligência comercial (BI) 114, o qual é executado pelo processador 111. O aplicativo BI 114 é configurado para extrair conhecimento comercial a partir do cubo 104 e apresentar o conhecimento comercial para o usuário 110 (por exemplo, visualizar o conhecimento comercial). Em uma concretização ilustrativa, o aplicativo BI 114 pode ser ou estar incluído em um aplicativo de planilha eletrônica. Apesar de o aplicativo BI 114 ser apresentado como sendo executado no dispositivo de computação cliente 108, é para ser entendido que o aplicativo BI 114 pode executar em um dispositivo de computação que seja acessível para o dispositivo de computação cliente 108 por meio de uma conexão de rede. Por exemplo, o dispositivo de computação cliente 108 pode possuir um navegador ou outro aplicativo adequado executando no mesmo, onde, por exemplo, o navegador pode ser direcionado para um dispositivo de computação remotamente situado que executa o aplicativo BI 114. Ou seja, o aplicativo BI 114 pode ser um aplicativo baseado na Rede ou oferecido como um serviço da Rede. Adicionalmente, é para ser entendido que o depósito de dados 102, o dispositivo de computação servidor 106, e/ou o dispositivo de computação cliente 108 podem estar situados em um único dispositivo de computação; assim, a arquitetura do sistema 100 é ilustrativa por natureza, e não pretendida para ser limitativa.

[0021] O aplicativo BI 114 inclui um componente receptor de

entrada 115 que recebe entrada a partir do usuário 110 (ou de um programa executável por computador) com respeito ao cubo de dados 104. Por exemplo, o componente receptor de entrada 115 pode receber entrada que índia um desejo de carregar conteúdo do cubo 104 (pelo menos uma parte do cubo 104) em uma parte da memória 112 do dispositivo de computação cliente 108 que está alocado para o aplicativo BI 114. Adicionalmente ou alternativamente, para aumentar a performance, em resposta ao componente receptor de entrada 15 recebendo uma indicação de que o cubo 104 é desejavelmente acessado, o aplicativo BI 114 pode ser configurado para abrir um canal de comunicações com o dispositivo de computação servidor 106, de modo que uma totalidade do cubo 104 não precise ser carregada dentro da memória 112 do dispositivo de computação cliente 108.

[0022] O aplicativo BI 114 adicionalmente inclui um componente construtor de consulta 116 que constrói uma consulta baseado na entrada recebida pelo componente receptor de entrada 115. Por exemplo, em resposta ao componente receptor de entrada 115 recebendo uma solicitação para carregar o cubo 104 na memória 112, o componente construtor de consulta 116 pode gerar uma consulta que, quando executada pelo dispositivo de computação servidor 106 através do cubo 104, causa que uma totalidade do cubo 104 seja recuperada a partir do depósito de dados 102 e carregada na memória 112 do dispositivo de computação cliente 08 (por exemplo, onde isto é representado como os dados 117). Como anteriormente indicado, o aplicativo BI 114 pode ser configurado para gerar consultas que podem ser executadas através de vários diferentes tipos de estruturas multidimensionais. Por consequência, como será descrito em maiores detalhes neste documento, o componente construtor de consulta 116 pode inicialmente gerar uma consulta em uma linguagem de nível relativamente alto que seja comum através de todos os tipos de cubo

que o aplicativo BI 114 está configurado para suportar. Tal consulta pode ser referida como uma consulta de "alto nível". O componente construtor de consulta 116 pode incluir um componente tradutor de consulta 118 que pode traduzir a consulta de alto nível para uma consulta que seja suportada pelo dispositivo de computação servidor 106. Portanto, em um exemplo, quando o aplicativo BI 114 é configurado para suportar consulta de três tipos diferentes de cubos, o usuário 110 não precisa aprender três diferentes linguagens de consulta para consultar através dos três tipos diferentes de cubo. Ao invés disso, o componente construtor de consulta 116 e o componente tradutor de consulta 118 estão configurados para manipular a construção de consultas em uma linguagem de consulta que corresponde ao cubo sendo acessado.

[0023] O aplicativo BI 114 adicionalmente inclui um componente apresentador 120 que é configurado para apresentar os dados 117 na memória 112, por exemplo, em um exibidor do dispositivo de computação cliente 108. De acordo com um exemplo, quando o componente construtor de consulta 116 constrói a consulta baseado na entrada recebida pelo componente receptor de entrada 115, o componente construtor de consulta 116 pode ser configurado para transmitir a consulta para o dispositivo de computação servidor 106.

[0024] O dispositivo de computação servidor 106 inclui um processador do servidor 122 e uma memória do servidor 124, onde a memória do servidor 124 compreende componentes que podem ser executados pelo processador do servidor 122. A memória do servidor 124 inclui um componente executor de consulta 126 que recebe a consulta a partir do dispositivo de computação cliente 108 e executa a consulta através do cubo 104 no depósito de dados 102. A memória do servidor 124 adicionalmente inclui um componente provedor de dados 128 que recebe os dados a partir do cubo 104 baseado na

consulta executada pelo componente executor de consulta 122, e o componente provedor de dados 124 proporciona tais dados 117 para o dispositivo de computação cliente 108, onde eles são colocados em uma parte da memória 112 acessível para o aplicativo BI 114. O componente apresentador 120 recupera os dados 117 a partir da memória 115 e apresenta tais dados para o usuário 110 (por exemplo, em formato de tabela).

[0025] Uma vez apresentado para os dados, o usuário 110 pode proporcionar entrada adicional com respeito aos dados apresentados. Por exemplo, o usuário 110 pode desejar receber medidas correspondendo a um atributo dimensão particular (por exemplo, o atributo dimensão "dia" para a dimensão que é representativa de tempo). Em outro exemplo, o usuário 110 pode desejar filtrar medidas baseado nos atributos dimensão particulares (por exemplo, o usuário 110 pode desejar excluir alguns dias representados nos dados 117). Em outro exemplo, o usuário pode desejar expandir um atributo dimensão, de modo que um atributo mais granular seja apresentado na tela de exibidor (desse modo aumentando o numero de fileiras nos dados apresentados para o usuário 110). Além disso, o usuário 110 pode desejar contrair um atributo dimensão, de modo que um atributo mais grosseiro seja apresentado na tela de exibidor (desse modo diminuindo o número de fileiras nos dados apresentados para o usuário 110). O componente receptor de entrada 114 pode receber tal entrada, e o componente construtor de consulta 116 pode refinar a consulta mencionada acima baseado na entrada. A consulta é transmitida pelo dispositivo de computador cliente 108 para o dispositivo de computador servidor 106, e o componente executor de consulta 126 do dispositivo de computação servidor 106 executa a consulta refinada através do cubo 104. O componente provedor de dados 128 proporciona dados retornados baseado na execução da

consulta através do cubo 104 para o dispositivo de computação cliente 108, onde eles são colocados na memória 112 como os dados 117. O componente apresentador 120 apresenta os dados 117 na forma tabular para o usuário 110. O usuário 110 então opcionalmente pode proporcionar entrada adicional com respeito aos dados apresentados ou aos estágios anteriores da consulta para adicionalmente refinar a consulta.

[0026] Portanto, pode ser verificado que o usuário 110 pode causar que uma consulta seja construída através de vários estágios (ou etapas), onde o usuário 110 pode ser apresentado para dados tabulares recuperados por consultar um cubo, e então, pode escolher operações adicionais que são para ser executadas através dos dados tabulares para refinar a consulta. Assim, O usuário 110 pode construir ou refinar a consulta por inspecionar os dados e então especificar operações adicionais para executar em relação aos dados. Isto é em contraste com as abordagens convencionais, onde o usuário 110 deve ter conhecimento de uma linguagem de consulta suportada pelo componente executor de consulta 126, deve ter conhecimento de uma parte do cubo que é de interesse para o usuário 110, e deve então construir uma consulta para recuperar a parte do cubo que é de interesse, onde a consulta é na linguagem de consulta referida acima. Nas abordagens convencionais, quando a consulta construída não resulta em apresentação de dados desejados para o usuário 110, o usuário 110 deve construir uma nova consulta (por exemplo, a partir de rascunho), ao invés de que através da abordagem de "consulta por exemplo" exposta neste documento. Assim, utilizando os aspectos descritos neste documento, o usuário 110 pode prontamente explorar conteúdos do cubo 104 para adquirir conhecimento comercial.

[0027] Vários exemplos pertencendo à operação do sistema 100 são agora expostos. Como indicado anteriormente, o cubo 104 pode

ser representado como uma tabela plana que compreende atributos dimensão e aplicações de medida, onde a tabela plana pode ser referida como uma tabela de "fatos". Uma consulta a ser executada através do cubo 104 pode ser "reduzida" a uma representação expressa em termos de operadores relacionais, assim permitindo que o usuário 110 interaja com o cubo 104 sem ser forçado a aprender linguagens de consulta específicas. Geralmente, um operador relacional é uma construção que testa ou define uma relação entre duas entidades. Operadores relacionais ilustrativos incluem, mas não estão limitados a filtros de fileira, seleção de coluna, classificação, dentre outros.

[0028] Uma tabela de dimensões é uma tabela que inclui uma coluna para cada atributo da dimensão. Assim, por exemplo, uma tabela de dimensões "tempo" pode incluir colunas separadas para ano, mês, semana, dia, e assim por diante. Em uma tabela de dimensões, os membros de cada atributo dimensão são enumerados, produzindo um produto vetorial dos membros dos atributos de uma dimensão. Por exemplo, uma tabela de clientes para uma dimensão "Geografia do Cliente" é exposta abaixo na Tabela 1:

TABELA 1

	Geografia do Cliente.País (Customer Geography.Country)	Geografia do Cliente.Estado- Província (Customer Geography.State- Province)	Geografia do Cliente.Cidade (Customer Geography. City)
1	Australia	New South Wales	Alexandria
2	Australia	New South Wales	Coffs Harbour
3	Australia	New South Wales	Darlinghurst
4	Australia	New South Wales	Goulburn
5	Australia	New South Wales	Lane Cove

	Geografia do Cliente.País (Customer Geography.Country)	Geografia do Cliente.Estado- Província (Customer Geography.State- Province)	Geografia do Cliente.Cidade (Customer Geography. City)
6	Australia	New South Wales	Lavender Bay
7	Australia	New South Wales	Malabar
8	Australia	New South Wales	Mettraville
9	Australia	New South Wales	Milsons Point
10	Australia	New South Wales	Newcastle

[0029] Conceitualmente, uma tabela de fatos possui tabelas de dimensões relacionadas com a mesma por meio de junções (dimensões "expandidas") ou junções aninhadas (dimensões "contraídas"). Por exemplo, uma tabela de fatos com "Geografia do Cliente" e "Produto" como dimensões contraídas é apresentada na Tabela 2:

TABELA 2

	Geografia do Cliente (Customer Geography) ← →	Produto (Product) ← →
1	Tabela (Table)	Tabela (Table)

[0030] Esta tabela resulta em uma junção aninhada da tabela de fatos com cada uma destas tabelas de dimensões, por exemplo:

`Table.NestedJoin(factTable, {}, #"Customer Geography", {}, "Customer Geography")`

onde, referindo-se aos argumentos na ordem, "factTable" é a tabela de fatos, "{ }" são as chaves do lado esquerdo, "#Customer Geography" é a tabela de dimensões, "{ }" são as chaves do lado direito, e "Customer Geography" é o nome da coluna para criar na qual colocar as tabelas aninhadas. Cada fileira na tabela de fatos representa uma coordenada que seleciona algum subconjunto dos dados no cubo.

Neste caso, com um conjunto não filtrado e não expandido de dimensões, todos os dados no cubo 104 seriam selecionados para esta única (e somente) fileira na tabela de fatos. O Aplicativo B 14 pode atuar para filtrar conteúdo para exibição, assim emitindo uma camada de exibição. A camada de exibição pode ocultar as dimensões contraídas (por exemplo, para proporcionar uma melhor e menor experiência confusa do usuário). É para ser entendido que cada única dimensão não precisa ser apresentada desta maneira, mas ao invés disso, dimensões que foram "tocadas" pelo usuário 110 pode ser apresentadas à medida que o usuário 110 está trabalhando com o cubo 104.

[0031] Como anteriormente indicado, uma tabela de dimensões pode ser expandida para produzir coordenadas com granulação mais fina, e o componente construtor de consulta 116 pode construir uma consulta que causa que tal expansão ocorra. Por exemplo, expandir a dimensão "Customer Geography" e selecionar os atributos Country, Province e City resulta no produto vetorial destes atributos expandidos e das fileiras da tabela de fatos, como apresentado na Tabela 3:

TABELA 3

	Geografia do Cliente.País (Customer Geography.Co untry)	Geografia do Cliente.Estado- Província (Customer Geography.State -Province)	Geografia do Cliente.Cidade (Customer Geography.Cit y)	Produto (Product)
1	Australia	New South Wales	Alexandria	Table
2	Australia	New South Wales	Coffs Harbour	Table
3	Australia	New South	Darlinghurst	Table

		Wales		
4	Australia	New South Wales	Goulburn	Table
5	Australia	New South Wales	Lane Cove	Table
6	Australia	New South Wales	Lavender Bay	Table
7	Australia	New South Wales	Malabar	Table
8	Australia	New South Wales	Mettraville	Table
9	Australia	New South Wales	Milsons Point	Table
10	Australia	New South Wales	Newcastle	Table

[0032] Em um exemplo, isto pode ser modelado como expandindo a coluna "Customer Geography" resultando da junção aninhada representada acima, por exemplo:

```
Table.ExpandTableColumn(
    Table.NestedJoin(factTable, {}, #"Customer Geography", {}, "Customer Geography"),
    "Customer Geography",
    {"Country", "State-Province", "City"}
)
```

[0033] Aqui, o argumento "Customer Geography" identifica a coluna a expandir; a lista de colunas a extrair da tabela também é proporcionada. Isto também é equivalente a uma junção (plana) entre a tabela de fatos e as tabelas de dimensões, por exemplo:

```
Table.Join(factTable, {}, #"Customer Geography", {})
```

onde "factTable" identifica a tabela de fatos, { } representa as chaves do lado esquerdo, #"Customer Geography" é a tabela de dimensões,

e { } são as chaves do lado direito. Pode ser observado que a dimensão "Product" permanece contraída, e pode ser oculta da visualização. Cada fileira pode selecionar o subconjunto de dados no cubo 104 correspondendo à coordenada observada, por exemplo, a fileira 3 pode selecionar todos os dados de Darlinghurst em New South Wales, Australia, para todos os Products.

[0034] Como indicador anteriormente, uma medida pode ser modelada como uma coluna calculada através da coordenada dimensão de uma fileira. Ela é uma coluna que é o resultado da mesma função sendo aplicada para cada fileira da tabela. Por exemplo, a aplicação da medida "Internet Sales Amt." Resulta na adição de uma coluna calculada, como apresentado na Tabela 4:

TABELA 4

	Geografia do Cliente.País (Customer Geography. Country)	Geografia do Cliente.Estado- Província (Customer Geography.State- Province)	Geografia do Cliente.Cida de (Customer Geography. City)	Produto (Product)	Quantidade de Vendas pela Internet (Internet Sales Amt.)
1	Australia	New South Wales	Alexandria	Tabela (Table)	Null
2	Australia	New South Wales	Coffs Harbour	Tabela (Table)	235454
3	Australia	New South Wales	Darlinghurst	Tabela (Table)	155010
4	Australia	New South Wales	Goulburn	Tabela (Table)	30875
5	Australia	New South Wales	Lane Cove	Tabela (Table)	220083
6	Australia	New South Wales	Lavender Bay	Tabela (Table)	195122
7	Australia	New South Wales	Malabar	Tabela (Table)	176905

	Geografia do Cliente.País (Customer Geography. Country)	Geografia do Cliente.Estado- Província (Customer Geography.State- Province)	Geografia do Cliente.Cida de (Customer Geography. City)	Produto (Product)	Quantidade de Vendas pela Internet (Internet Sales Amt.)
8	Australia	New South Wales	Mettraville	Tabela (Table)	216564
9	Australia	New South Wales	Milsons Point	Tabela (Table)	187075
10	Australia	New South Wales	Newcastle	Tabela (Table)	245936

[0035] Em resposta ao componente receptor de entrada 115 receber uma solicitação para calcular a medida mencionada cima, o componente construtor de consulta 116 pode construir uma consulta tal como a seguinte que causa que a medida seja calculada:

```
Table.AddColumn(
    factTable,
    "Internet Sales Amount",
    (row) => ApplyMeasure(
        #"Internet Sales Amount",
        [
            Country = row[Country],
            #"State-Province" = row[State-Province],
            City = row[City],
            Product = row[Product]
        ]
    )
)
```

[0036] Neste exemplo, a medida "Internet Sales Amount" é aplicada em relação a um registro construído a partir das coordenadas de dimensão disponíveis. No exemplo exposto acima, os membros

individuais (dependendo da fileira) dos atributos da dimensão "Customer Geography" podem ser passados para cada invocação da medida enquanto todo o conjunto contraído de membros "Product" podem ser passados para cada invocação da medida. Deste modo, a medida é aplicada para o subconjunto do cubo 104 descrito por esta fileira. Por exemplo, a fileira 3 na Tabela 4 acima aplica a medida Internet Sales Amount para o subconjunto do cubo 104 para todos os produtos na região geográfica Australia, New South Wales, Darlinghurst.

[0037] Uma camada de projeção final pode ser colocada na parte de cima do cubo 104 que oculta as colunas de dimensão contraídas de modo que ela não interfere com o usuário 110 ou confunde o usuário 110. A projeção pode ser executada utilizando um operador que é configura para remover colunas de uma tabela (por exemplo, Table.RemoveColumns).

[0038] Adicionalmente, o componente construtor de consulta 116 pode construir uma consulta que causa que medidas sejam "flutuadas" quando uma indicação que expansão ou contração de uma dimensão é solicitada. Por exemplo, usuários esperam que as medidas selecionadas sejam aplicadas para um conjunto atual de coordenadas de dimensão, e que as medidas se atualizem quando as coordenadas de dimensão são alteradas. Por exemplo, se a medida "Internet Sales Amount" for aplicada para um cubo totalmente contraído, a seguinte tabela ilustrativa pode ser apresentada:

TABELA 5

	Quantidade de Vendas pela Internet (Internet Sales Amount)
1	29358677

[0039] O componente receptor de entrada 115 pode receber uma solicitação para expandir o atributo "Country" da dimensão "Customer Geography", e o componente construtor de consulta 116 pode

construir uma consulta que causa que uma tabela que o usuário 110 espera ver seja apresentada:

TABELA 6

	Quantidade de Vendas pela Internet (Internet Sales Amount)	Geografia do Cliente.País (Customer Geography.Country)
1	9061000	Australia
2	1977844	Canada
3	2644017	França (France)
4	2894312	Alemanha (Germany)
5	3391712	Reino Unido (United Kingdom)
6	938789	Estados Unidos (United States)

[0040] Isto é referido como medidas "flutuantes". Como indicado anteriormente, o componente construtor de consulta 116 pode identificar uma solicitação para calcular uma medida, e pode posicionar o cálculo em uma consulta construída para um ponto após o atributo dimensão relevante ter sido selecionado.

[0041] Em outro exemplo, uma coluna pode ser uma coluna calculada, onde uma coluna calculada transporta uma função que foi utilizada para construir a coluna. Por exemplo, um usuário pode inspecionar a Tabela 6 e solicitar a função utilizada para calcular a coluna "Internet Sales Amount":

[0042] `ComputedColumns(factTable, "Internet Sales Amount"),`

[0043] Resultando no retorno da seguinte expressão:

```
(row) => ApplyMeasure(
    #"Internet Sales Amount",
    [
        Country = row[Country]
    ]
)
```

[0044] Se uma nova coluna fosse para ser criada com esta função, uma cópia exata da coluna original será proporcionada.

[0045] As colunas calculadas podem ser utilizadas para flutuar uma medida. Sempre que uma operação de cubo que altera a dimensionalidade de um cubo (por exemplo, uma dimensão é adicionada, uma dimensão é expandida, ou uma dimensão é contraída) é aplicada, todas as medidas põem ser flutuadas, de modo que as medidas são recalculadas em relação ao conjunto correto de coordenadas de dimensão. O sistema 100 pode executar um processo com várias etapas para flutuar a medida apropriadamente. Com mais especificidade, primeiro, o componente construtor de consulta 116 pode coletar expressões para colunas calculadas da tabela que representam aplicações de medida. Depois disso, o componente construtor de consulta 116 pode remover as colunas calculadas para aplicações de medida da tabela. A dimensionalidade da tabela então pode ser atualizada, por exemplo, por adicionar uma medida, expandir uma dimensão, ou contrair uma dimensão. As expressões de aplicação de medida são ajustadas para incorporarem a nova dimensionalidade, e as novas expressões de aplicação de medida são aplicadas para a tabela. Desta maneira, as aplicações de medida põem ser reordenadas como se elas tivessem ocorrido após a dimensionalidade do cubo ser ajustada.

[0046] Por exemplo, se o usuário 119 iniciar com a medida "Internet Sales Amount" aplicada para apenas o atributo dimensão "Country", e expandir a dimensão "Customer Geography" para incluir o atributo "City", o seguinte processo pode ocorrer (referindo-se à Tabela 6). Primeiro, o componente construtor de consulta 116 pode extrair a expressão para a coluna calculada:

```
(row) => ApplyMeasure(
    #"Internet Sales Amount",
    [
        Country = row[Country]
    ]
)
```

[0047] Depois disso, o componente construtor de consulta 16 pode construir a consulta de modo que a coluna calculada seja removida:

```
Table.RemoveColumns(factTable, {"Internet Sales Amount"})
```

[0048] A Tabela 7 ilustra a tabela resultante (onde a coluna calculada na Tabela 6 foi removida).

TABELA 7

	Geografia do Cliente.País (Customer Geography.Country)
1	Australia
2	Canada
3	França (France)
4	Alemanha (Germany)
5	Reino Unido (United Kingdom)
6	Estados Unidos (United States)

[0049] A alteração de dimensionalidade pode subsequentemente ser aplicada: por exemplo:

```
ExpandDimension(factTable, #"Customer Geography", {"City"})
```

[0050] Esta expansão é apresentada na Tabela 1.

[0051] O componente construtor de consulta 116 ajusta a expressão para a medida incluir a nova dimensionalidade:

```
(row) => ApplyMeasure(
    #"Internet Sales Amount",
    [
        Country = row[Country],
        City = row[City]
    ]
)
```

[0052] Aqui, então, a expressão para a medida contempla o atributo "City" da dimensão "Customer Geography"

[0053] Finalmente, o componente construtor de pesquisa 116 pode reaplicar a nova medida:

```

Table.AddColumn(
    factTable,
    "Internet Sales Amount",
    (row) => ApplyMeasure(
        #"Internet Sales Amount",
        [
            Country = row[Country],
            City = row[City]
        ]
    )
)

```

[0054] Por exemplo, isto pode resultar na Tabela 8 sendo gerada pelo componente construtor de consulta 126.

TABELA 8

	Quantidad e de Vendas pela Internet (Internet Sales Amt.)	Geografia do Cliente.País (Customer Geography.C ountry)	Geografia do Cliente.Estado- Província (Customer Geography.State- Province)	Geografia do Cliente.Cidade (Customer Geography.Cit y)
1	Null	Australia	New South Wales	Alexandria
2	235454	Australia	New South Wales	Coffs Harbour
3	155010	Australia	New South Wales	Darlinghurst
4	30875	Australia	New South Wales	Goulburn
5	220083	Australia	New South Wales	Lane Cove
6	195122	Australia	New South Wales	Lavender Bay
7	176905	Australia	New South Wales	Malabar
8	216564	Australia	New South Wales	Mettraville

	Quantidade de Vendas pela Internet (Internet Sales Amt.)	Geografia do Cliente.País (Customer Geography.Country)	Geografia do Cliente.Estado-Província (Customer Geography.State-Province)	Geografia do Cliente.Cidade (Customer Geography.City)
9	187075	Australia	New South Wales	Milsons Point
10	245936	Australia	New South Wales	Newcastle

[0055] Esta "flutuação de colunas de medida é examinada sob composição funcional e interativa. Em outras palavras, etapas adicionais na consulta que alteram dimensionalidade da tabela em cubo (e assim, medidas flutuantes) podem continuar a ser acumuladas sem ter que voltar e ajustar qualquer uma das etapas anteriores na cadeia.

[0056] O sistema 100 adicionalmente suporta operadores de cubo que podem ser utilizados para manipular dados do cubo. Tais operadores podem ser envoltório ao redor de primitivas relacionais existentes, com a exceção de que os operadores adicionam outras operações para medidas flutuantes ao redor da operação relacional principal. Para inteireza, uma função FactRowsExist é introduzida abaixo, a qual seleciona somente as combinações de coordenadas a partir do produto vetorial de todas as dimensões que realmente selecionam algumas fileiras de fato a partir dos dados no cubo 104 subjacente. Um filtro similar pode ser utilizado para preservar as semânticas corretas das operações do cubo quando elas são reduzidas para o espaço relacional.

[0057] Primeiro, um operador "AddDimension" é descrito, o qual adiciona uma dimensão para uma tabela. Este operador recebe uma identidade da tabela de fatos, uma identidade de um nome de coluna

de uma tabela de dimensões, e uma identidade da tabela de dimensões como entrada. Por exemplo: `AddDimension(factTable, dimensionColumn, dimensionTable)` pode incorporar uma tabela de dimensões (possivelmente filtrada) dentro da tabela de fatos utilizando uma junção aninhada. Este operador pode alterar a dimensionalidade do cubo se a tabela de dimensões foi filtrada. Pode ser observado que a filtragem da tabela de dimensões pode ser realizada por meio de um operador que seleciona fileiras em uma tabela (por exemplo, `Table.SelectRows`). Uma forma relacional reduzida do operador ilustrativo é exposta abaixo:

```
Table.AddColumn(
  Table.SelectRows(
    Table.NestedJoin(
      Table.RemoveColumns(factTable, {measure columns, ...}),
      {},
      dimensionTable,
      {},
      dimensionColumnName),
    (row) => FactRowsExist(row coordinate)),
  columns ctor for measure columns)
```

[0058] Um operador "ExpandDimension" expande atributos de uma tabela de dimensões anteriormente anexada (por exemplo, por meio do operador `AddDimension`), e altera a dimensionalidade do cubo. Este operador ilustrativo obtém a tabela de fatos, uma identidade de um nome de coluna de dimensão, e uma identidade de pelo menos um nome de atributo como entrada, por exemplo:

[0059] `ExpandDimension(factTable, dimensionColumnName, {dimensionAttributeName1, ...})`.

[0060] Uma forma relacional reduzida ilustrativa deste operador é exposta abaixo:

```

Table.AddColumn(
  Table.ExpandTableColumn(
    Table.RemoveColumns(factTable, {measure columns}),
    dimensionColumnName,
    {dimension attributes, ...})
  columns ctor for measure columns)

```

[0061] Um operador "CollapseDimension" contrai atributos de uma dimensão anteriormente expandida, e altera a dimensionalidade do cubo. O operador CollapseDimension obtém a tabela de fatos e uma identidade de pelo menos um nome de atributo; por exemplo, CollapseDimension(*factTable*, {*dimensionAttributeName1*, ...}). Pode ser observado que é possível adicionar uma filtro em relação aos membros dimensão (e medidas) por meio de um operador que seleciona fileiras antes de contrair uma dimensão. Por exemplo, isto pode criar uma fração em relação a uma dimensão, e pode ser utilizado para implementar filtragem cruzada de dimensões e medidas (por exemplo, um filtro que faz referência a várias dimensões de uma vez em uma cláusula "ou"). Uma forma relacional reduzida ilustrativa do operador CollapseDimension é apresentada abaixo:

```

Table.AddColumn(
  Table.Group(
    Table.RemoveColumns(factTable, {measure columns}),
    {all other dimension attributes},
    {ctor for table of members of collapsed dimension attribute})

```

[0062] Um operador "AddMeasure" aplica a medida para a tabela de fatos, onde o registro de coordenada de dimensão pode ser da forma apresentada em exemplos anteriores. O operador AddMeasure obtém como entrada a *factTable*, uma identidade de um nome de

coluna, e uma identidade de uma função medida que é para ser aplicada; por exemplo, `AddMeasure(factTable, columnName, measureFunction)`. Uma forma relacional reduzida ilustrativa do operador `AddMeasure` é apresentada abaixo:

```
Table.AddColumn(
    factTable,
    columnName,
    (row) => ApplyMeasure(
        measureFunction,
        [dimension coordinate record for row]))
```

[0063] Consultas expressas nas forma citadas acima podem ser traduzidas e proporcionadas em qualquer ordem (por exemplo, pelo componente construtor de consulta 116 e/ou pelo componente executor de consulta 126) para uma consulta que pode ser executada em relação ao cubo desejado 104. A descrição exposta abaixo assume familiaridade com as árvores de sintaxe abstrata (ASTs) e/ou com as árvores de expressão, e adicionalmente assume familiaridade com normalização de tais árvores em formas padrão. Conceitualmente, o processo de tradução de consulta é exposto abaixo. Primeiro, o usuário 110 (ou programa) expressa operações em termos de operadores de cubo e relacionais. Depois disso, as operações de cubo são "reduzidas" para operadores relacionais e medidas flutuantes. Subsequentemente, as expressões são normalizadas para reordenar os operadores e estabelecidas para padrões de árvore de expressão normalizada. Então, os padrões de árvore de expressão normalizada podem ser "aumentados" para uma "expressão de cubo" para corresponder às expectativas de servidores de cubo – por exemplo, de modo que o componente executor de

consulta 126 possa executar a consulta. A expressão de cubo pode então ser traduzida para uma sintaxe específica do servidor.

[0064] Regras ilustrativas de normalização de consulta que podem ser aplicadas pelo componente construtor de consulta 115 são agora expostas. Por exemplo, a seleção de coluna e os filtros de fileira podem ser reduzidas o máximo possível para a árvore de expressão. Uma junção aninhada que é expandida pode ser convertida para uma junção plana; por exemplo, `ExpandtableColumn(NetedJoin(x,y),{all cols of y})` \rightarrow `Join(x,y)`. Uma remoção de uma coluna adicionada é como se a coluna nunca tivesse sido adicionada: por exemplo, `RemoveColumn(AddColumn(x),x)` \rightarrow `no-op`. Um agrupamento na parte de cima de uma junção plana pode ser convertida para uma junção aninhada em alguns casos: por exemplo, `Group(Join(x,y),{all cols of x},{table of y})` \rightarrow `NestedJoin(x,y)`.

[0065] Ações ilustrativas pertencendo à tradução para uma "expressão de cubo", as quais podem ser executadas pelo componente construtor de consulta 116, são agora expostas. Padrões normalizados específicos podem ser detectados dentro da expressão de consulta reduzida e traduzidos para uma "expressão de cubo" – uma árvore de expressão que corresponde rigorosamente à gramática de linguagens de consulta multidimensionais. Em um exemplo, uma junção plana com uma tabela de dimensões adiciona os atributos da dimensão para a expressão de cubo: por exemplo, `Join(factTable, {}, dimensionTable, {})` se traduz para o seguinte:

Query

From: *factTable cube*

Dimensions: [dimAttr1], [dimAttr2], ...

Measures:

Filter: (null)

[0066] Em outro exemplo, uma junção aninhada com uma tabela

de dimensão filtrada empurra o filtro contra os atributos da dimensão (por exemplo, um fracionador) dentro de uma subconsulta da expressão de cubo. Por exemplo, `NestedJoin(factTable, {}, SelectRows(dimensionTable, (r) => r[City] = Seattle))` se traduz para o seguinte:

Query

From:

Query

From: *factTable cube*

Dimensions: [City]

Measures:

Filter: equals([City], "Seattle")

Dimensions:

Measures:

Filter:

[0067] Ainda em outro exemplo, uma aplicação de medida adiciona uma referência de medida para a expressão de cubo. Por exemplo,

`AddColumn(`

factTable,

"Internet Sales Amount",

(row) => ApplyMeasure("#Internet Sales Amount", []))

traduz-se para:

Query

From: *factTable cube*

Dimensions:

Measures: [Internet Sales Amount]

Filter:

[0068] Ainda em outro exemplo, um filtro de fileira (por exemplo, `SelectRows`) pode ser traduzido para uma expressão de filtro em relação às medidas e dimensões. Assim, por exemplo, `Table.SelectRows(factTable, each {Internet Sales Amount} > 500)` se traduz para o seguinte:

Query

From: *factTable cube*

Dimensions:

Measures: ...

Filter: `greater-than([Internet Sales Amount], 500)`

[0069] Ainda em outro exemplo, um grupo de todas outras dimensões é traduzido para uma operação de "contração" da dimensão restante. A dimensão contraída pode ser enviada para uma subconsulta na expressão de cubo, e as outras dimensões permanecem na expressão de cubo externa. Portanto, por exemplo, `Table.Group(factTable, {"dim2", "dim3"}, {"collapsed dim1", (rows) => rows[dim1]})` se traduz para o seguinte:

Query

From:

Query

From: *factTable cube*

Dimensions: [dim1]

Measures:

Filter:

Dimensions:

Measures: [dim2], [dim3]

Filter:

[0070] O componente construtor de consulta 116 pode traduzir a

expressão de cubo para uma linguagem e sintaxe esperada de um servidor de cubo desejado. Vários exemplos são expostos abaixo para esclarecer as ações descritas acima.

Exemplo 1 – Adiciona uma Dimensão

[0071] A consulta seguinte pode ser apresentada por um usuário ou programa:

```
AddDimension(emptyFactTable, "Customer Geography", #"Customer Geography")
```

[0072] O componente construtor de consulta 116 pode reduzir a consulta para o seguinte:

```
Table.SelectRows(
    Table.NestedJoin(
        emptyFactTable,
        {},
        #"Customer Geography",
        {},
        "Customer Geography"),
    (row) => FactRowsExist([#"Customer Geography" = row[Customer
Geography]))
```

[0073] Para esta consulta ilustrativa, nenhuma normalização é executada, e o dito acima é traduzido para uma expressão de cubo vazia (desde que não existe operação de cubo a executar). Adicionalmente, a consulta não precisa ser traduzida para a linguagem de um servidor de cubo, desde que não existe consulta a ser executada através do cubo 104. O resultado é o retorno de uma tabela vazia.

Exemplo 2 – Expandir uma Dimensão

[0074] Nos exemplos seguintes, o formato da expressão de cubo é como dito a seguir: "From" se refere a um cubo de entrada; "Dimensions" se refere às dimensões a serem expandidas; "Measures" são medidas a se aplicar; e o filtro referenciado nos exemplos filtra fileiras resultantes baseado em seus predicados. A consulta seguinte

pode ser apresentada por um usuário ou programa:

```
ExpandDimension(
  AddDimension(emptyFactTable, "Customer Geography", #"Customer
Geography"),
  {"Country", "State-Province", "City"})
```

[0075] O componente construtor de consulta 16 pode reduzir a consulta como a seguir:

```
Table.ExpandTableColumn(
  Table.SelectRows(
    Table.NestedJoin(
      emptyFactTable,
      {},
      #"Customer Geography",
      {},
      "Customer Geography"),
    (row) => FactRowsExist([#"Customer Geography" = row[Customer Geography]]),
    {"Country", "State-Province", "City"})
```

[0076] O componente construtor de consulta 116 pode então normalizar a expressão reduzida por enviar o ExpandtableColumn através de SelectRows, e então, converter o padrão ExpandTableColumn(NestedJoin) para uma junção plana como a seguir:

```
Table.SelectRows(
  Table.Join(
    emptyFactTable,
    {},
    #"Customer Geography",
    {}),
  (row) => FactRowsExist([#"Customer Geography" = row[Customer Geography]]))
```


[0077] O componente construtor de consulta 116 pode então traduzir a expressão normalizada para uma expressão de cubo:

Query

RowRange:

skip:0, take:Infinite

From:

Identifier([Adventure Works])

Dimensions:

Identifier([Customer].[Customer Geography].[Country])

Identifier([Customer].[Customer Geography].[State-Province])

Identifier([Customer].[Customer Geography].[City])

Measures:

Identifier([Measures].[Internet Sales Amount])

Filter:

(null)

Sort:

[0078] O componente construtor de consulta 116 e/ou o componente executor de consulta 126 podem traduzir a expressão de cubo para uma linguagem suportada para consultar o cubo 104. O componente provedor de dados 128 pode então proporcionar a Tabela 1 para o dispositivo de computação cliente 108 para exibição.

Exemplo 3 – Adiciona uma Medida

[0079] A seguinte consulta pode ser apresentada por um usuário ou programa:

AddMeasure(

ExpandDimension(

AddDimension(*emptyFactTable*, "Customer Geography", #"Customer Geography"),

{"Country", "State-Province", "City"}),

"Internet Sales Amount",

#"Internet Sales Amount")

[0080] O componente construtor de consulta 116 pode reduzir tal consulta para o seguinte:

```

Table.AddColumn(
  Table.ExpandTableColumn(
    Table.SelectRows(
      Table.NestedJoin(
        emptyFactTable,
        {},
        #"Customer Geography",
        {},
        "Customer Geography"),
        (row) => FactRowsExist([#"Customer Geography" = row[Customer
Geography]])),
        {"Country", "State-Province", "City"}),
    "Internet Sales Amount",
    (row) => ApplyMeasure(
      #"Internet Sales Amount",
      [
        Country = row[Country],
        #"State-Province" = row[State-Province],
        City = row[City]
      ]
    )
  )
)

```

[0081] O componente construtor de consulta 116 pode então converter a expressão para ter uma Junção plana como no exemplo anterior:

```

Table.AddColumn(
  Table.SelectRows(
    Table.Join(
      emptyFactTable,
      {},
      #"Customer Geography",
      {}),
    (row) => FactRowsExist([#"Customer Geography" = row[Customer Geography]]),
    "Internet Sales Amount",
    (row) => ApplyMeasure(
      #"Internet Sales Amount",
      [
        Country = row[Country],
        #"State-Province" = row[State-Province],
        City = row[City]
      ]
    )
  )
)

```

[0082] O componente construtor de consulta 116 pode então traduzir a expressão normalizada para uma expressão de cubo;

```

Query
  RowRange:
    skip:0, take:Infinite
  From:
    Identifier([Adventure Works])
  Dimensions:
    Identifier([Customer].[Customer Geography].[Country])
    Identifier([Customer].[Customer Geography].[State-Province])
    Identifier([Customer].[Customer Geography].[City])
  Measures:
    Identifier([Measures].[Internet Sales Amount])
  Filter:
    (null)
  Sort:

```

[0083] O componente construtor de consulta 116 e/ou o componente executor de consulta 126 podem traduzir a expressão de cubo para uma linguagem e/ou sintaxe correspondendo ao cubo 104. Uma tabela resultante da execução de tal consulta é apresentada na Tabela 4.

Exemplo 4 - Filtros

[0084] A seguinte consulta pode ser apresentada por um usuário ou programa:

```
Table.SelectRows(
  Table.SelectRows(
    AddMeasure(
      ExpandDimension(
        AddDimension(emptyFT, "Customer Geography", #"Customer
Geography"),
        {"Country", "State-Province", "City"}),
      "Internet Sales Amount",
      #"Internet Sales Amount"),
    (row) => row[City] = "Seattle"),
  (row) => row[Internet Sales Amount] > 500)
```

[0085] O componente construtor de consulta 116 pode reduzir tal consulta para o seguinte:

```
Table.SelectRows(
  Table.SelectRows(
    Table.AddColumn(
      Table.ExpandTableColumn(
        Table.SelectRows(
          Table.NestedJoin(
            emptyFactTable,
            {},
            #"Customer Geography",
            {},
            "Customer Geography"),
            (row) => FactRowsExist([#"Customer Geography" = row[...]]),
            {"Country", "State-Province", "City"}),
        "Internet Sales Amount",
        (row) => ApplyMeasure(
          #"Internet Sales Amount",
          [
            Country = row[Country],
            #"State-Province" = row[State-Province].
```

```

        City = row[City]
    ]
)
),
(row) => row[City] = "Seattle"),
(row) => row[Internet Sales Amount] > 500)

```

[0086] O componente construtor de consulta 116 pode então normalizar a expressão acima para ter uma junção plana. O componente construtor de consulta 116 também pode enviar o filtro "fileira" para o atributo dimensão City abaixo da junção com a tabela de dimensões "Customer Geography", como apresenta aqui:

```

Table.SelectRows(
    Table.AddColumn(
        Table.SelectRows(
            Table.Join(
                emptyFactTable,
                {},
                Table.SelectRows(
                    #"Customer Geography",
                    (row) => row[City] = "Seattle"),
                {}),
            (row) => FactRowsExist([#"Customer Geography" = row[Customer
Geography]])),
        "Internet Sales Amount",
        (row) => ApplyMeasure(
            #"Internet Sales Amount",
            [
                Country = row[Country],
                #"State-Province" = row[State-Province],
                City = row[City]
            ]
        )
    )
)

```

```

    ]
  )
),
(row) => row[Internet Sales Amount] > 500)

```

[0087] O componente construtor de consulta 116 então pode traduzir a expressão normalizada para uma expressão de cubo:

Query

RowRange:

skip:0, take:Infinite

From:

Identifier([Adventure Works])

Dimensions:

Identifier([Customer].[Customer Geography].[Country])

Identifier([Customer].[Customer Geography].[State-Province])

Identifier([Customer].[Customer Geography].[City])

Measures:

Identifier([Measures].[Internet Sales Amount])

Filter:

And

Equals

Identifier([Customer].[Customer Geography].[City])

Constant(Seattle)

GreaterThanOrEquals

Identifier([Measures].[Internet Sales Amount])

Constant(500)

Sort:

[0088] O componente construtor de cubo 16 e/ou o componente executor de consulta 126 pode traduzir a expressão acima para uma linguagem ou sintaxe que possa ser utilizada para consultar o cubo 04. Executar esta consulta pode resultar em obter a seguinte tabela:

TABELA 9

	Geografia do Cliente.País (Customer Geography.Co untry)	Geografia do Cliente.Estado- Província (Customer Geography.Stat e-Province)	Geografia do Cliente.Cida de (Customer Geography. City)	Quantidade de Vendas pela Internet (Internet Sales Amount)
1	Estados Unidos (United States)	Washington	Seattle	75164

Exemplo 5: Contração

[0089] A seguinte consulta pode ser apresentada por um usuário ou programa:

```
CollapseDimension(
  Table.SelectRows(
    AddMeasure(
      ExpandDimension(
        AddDimension(emptyFT, "Customer Geography", #"Customer
Geography"),
        {"Country", "State-Province", "City"}),
        "Internet Sales Amount",
        #"Internet Sales Amount"),
      (row) => row[City] = "Seattle"),
    {"City", "State-Province"})
```

[0090] E para ser observado que o filtro medida apresente nos exemplos anteriores neste documento tinham sido omitidos para propósito de brevidade. O componente construtor de consulta 116 pode reduzir esta consulta para o seguinte:

```

Table.AddColumn(
  Table.Group(
    Table.RemoveColumns(
      Table.SelectRows(
        Table.AddColumn(
          Table.ExpandTableColumn(
            Table.SelectRows(
              Table.NestedJoin(
                emptyFactTable,
                {},
                #"Customer Geography",
                {},
                "Customer Geography"),
                (row) => FactRowsExist([#"Customer Geography" = row[...]]),
                {"Country", "State-Province", "City"}),
              "Internet Sales Amount",
              (row) => ApplyMeasure(
                #"Internet Sales Amount",
                [
                  Country = row[Country],
                  #"State-Province" = row[State-Province],
                  City = row[City]
                ]
              )
            ),
            (row) => row[City] = "Seattle"),
            {"Internet Sales Amount"}),
            {"Country"}},
            {
              {"City", (rows) => rows[City]},
              {"State", (rows) => rows[State]}
            }
          ),
          "Internet Sales Amount",
          (row) => ApplyMeasure(
            #"Internet Sales Amount",
            [
              Country = row[Country]
            ]
          )
        )
      )
    )
  )

```

[0091] A adição das operações para "flutuar" a medida são para ser observadas, desde que a dimensionalidade da tabela está

alterando. Adicionalmente, pode ser observado que a aplicação de medida foi ajustada para contar com o novo conjunto de dimensões (Country). Pode ser adicionalmente observador que a operação Group é empregada, a qual agrupa pelos atributos dimensão restantes que não estão sendo contraídos. Primeiro, ExpandTableColumn(NestedJoin) é substituído por uma Junção plana como nos exemplos anteriores:

```
Table.AddColumn(
  Table.Group(
    Table.RemoveColumns(
      Table.SelectRows(
        Table.AddColumn(
          Table.SelectRows(
            Table.Join(
              emptyFactTable,
              {},
              #"Customer Geography",
              {}),
            (row) => FactRowsExist([#"Customer Geography" = row[...]]),
          ),
            "Internet Sales Amount",
            (row) => ApplyMeasure(
              #"Internet Sales Amount",
              [
                Country = row[Country],
                #"State-Province" = row[State-Province],
                City = row[City]
              ]
            ),
            (row) => row[City] = "Seattle"),
            {"Internet Sales Amount"}),
            {"Country"},
            {
              {"City", (rows) => rows[City]},
              {"State", (rows) => rows[State]}
            },
            "Internet Sales Amount",
            (row) => ApplyMeasure(
              #"Internet Sales Amount",
              [
                Country = row[Country]
              ]
            )
          )
        )
      )
    )
  )
```

[0092] O filtro em relação à dimensão City é reduzido para a tabela dimensão:

```
Table.AddColumn(
  Table.Group(
    Table.RemoveColumns(
      Table.AddColumn(
        Table.SelectRows(
          Table.Join(
            emptyFactTable,
            {},

            Table.SelectRows(
              #"Customer Geography",
              (row) => row[City] = "Seattle"),
            {}),
          (row) => FactRowsExist([#"Customer Geography" = row[...]]),
        ),
        "Internet Sales Amount",
        (row) => ApplyMeasure(
          #"Internet Sales Amount",
          [
            Country = row[Country],
            #"State-Province" = row[State-Province],
            City = row[City]
          ]
        ),
        {"Internet Sales Amount"}),
        {"Country"},
        {
          {"City", (rows) => rows[City]},
          {"State", (rows) => rows[State]}
        },
        "Internet Sales Amount",
        (row) => ApplyMeasure(
          #"Internet Sales Amount",
          [
            Country = row[Country]
          ]
        )
      )
    )
  )
```

[0093] O par RemoveColumns(AddColumn) também é eliminado desde que a coluna removida não é mais necessária. Por ser observado que isto é uma parte chave da "flutuação" de medidas e torna esta consulta eficiente para avaliar após a normalização:

```
Table.AddColumn(
  Table.Group(
    Table.SelectRows(
      Table.Join(
        emptyFactTable,
        {},
        Table.SelectRows(
          #"Customer Geography",
          (row) => row[City] = "Seattle"),
        {}),
      (row) => FactRowsExist([#"Customer Geography" = row[...]]),
      {"County"}),
    {
      {"City", (rows) => rows[City]},
      {"State", (rows) => rows[State]}
    }
  ),
  "Internet Sales Amount",
  (row) => ApplyMeasure(
    #"Internet Sales Amount",
    [
      Country = row[Country]
    ]
  )
)
```

[0094] Finalmente, o componente construtor de consulta 116 pode normalizar a combinação Group(Join) em um NestedJoin:

```

Table.AddColumn(
  Table.SelectRows(
    Table.NestedJoin(
      emptyFactTable,
      {},
      Table.SelectRows(
        #"Customer Geography",
        (row) => row[City] = "Seattle"),
      {},
      "Customer Geography"),
    (row) => FactRowsExist([#"Customer Geography" = row[...]]),
    "Internet Sales Amount",
    (row) => ApplyMeasure(
      #"Internet Sales Amount",
      [
        Country = row[Country]
      ]
    )
  )

```

[0095] O componente construtor 116 pode então traduzir a expressão para uma expressão de cubo:

Query

RowRange:

skip:0, take:Infinite

From:

Identifier([Adventure Works])

Dimensions:

Identifier([Customer].[Customer Geography].[Country])

Measures:

Identifier([Measures].[Internet Sales Amount])

Filter:

Equals

Identifier([Customer].[Customer Geography].[City])

Constant(Seattle)

Sort:

[0096] O componente construtor 116 e/ou o componente executor 126 podem traduzir a consulta para uma linguagem e/ou sintaxe que possa ser utilizada para executar consultar através do cubo 104. A tabela seguinte pode ser recuperada baseada na consulta:

TABELA 10

	Geografia do Cliente.País (Customer Geography.Country)	Quantidade de Vendas pela Internet (Internet Sales Amount)
1	Estados Unidos (United States)	75164

[0097] As figuras 2 até 4 ilustram metodologias ilustrativas se relacionando com a construção de uma consulta para execução através de um cubo. Apesar de as metodologias serem apresentadas e descritas como sendo uma série de atos que são executadas em uma sequência, é para ser entendido e apreciado que as metodologias não estão limitadas pela ordem da sequência. Por exemplo, alguns atos podem ocorrer em uma ordem diferente do que é descrito neste documento. Em adição, um ato pode ocorrer simultaneamente com outro ato. Adicionalmente, em alguns casos, nem todos os atos podem ser requeridos para implementar uma metodologia descrita neste documento.

[0098] Além disso, os atos descritos neste documento podem ser instruções executáveis por computador que podem ser implementadas por um ou mais processadores e/ou armazenadas em um meio ou mídia legível por computador. As instruções executáveis por computador podem incluir uma rotina, uma sub-rotina, programas, um encadeamento de execução, e/ou dentre outros. Além disso, resultados dos atos da metodologias podem ser armazenados em um meio legível por computador, exibidos em um dispositivo de vídeo, e/ou coisa parecida.

[0099] Agora, com referência à figura 2, é ilustrado um fluxograma

ilustrando uma metodologia ilustrativa 200 para construir uma consulta que pode ser executada através de um cubo. A metodologia 200 inicia em 202, e em 204 um cubo é recebido. Em 206, uma seleção de pelo menos um atributo dimensão e de pelo menos uma medida no cubo é recebida. Em um exemplo, o cubo pode incluir dados que identificam vendas por localização e tempo. Em tal exemplo, vendas é a medida e a localização e o tempo são as dimensões. Os atributos da dimensão localização pode ser cidade, estado e país, enquanto atributos da dimensão tempo podem ser semanas, meses e anos.

[00100] Em 208, é construída uma consulta baseada na seleção do pelo menos um atributo dimensão e da pelo menos uma medida. Em um exemplo, se a seleção foi o atributo dimensão localização "city", e a medida selecionada for "Sales", a consulta construída pode ser configurada para recuperar vendas por cidade a partir do cubo.

[00101] Em 210, são recebidos dados baseados na consulta. Especificamente, a consulta é executada através do cubo, e dados gerados baseados na execução da consulta através do cubo são recebidos. Os dados podem ser apresentados para um usuário em forma tabular em um vídeo de um dispositivo de computação. Em 212, é recebida entrada com respeito aos dados apresentados. Tal entrada pode ser uma solicitação para filtrar os dados apresentados baseado em um valor de atributo particular, contração de um atributo dimensão, expansão de um atributo dimensão, adição de um atributo dimensão, adição de uma medida, etc. Exemplos deste tipo de entrada do usuário são agora expostos. Continuando com o exemplo onde o usuário adquiriu números de vendas por atributo dimensão "city", o usuário pode desejar receber números de venda a partir de cidades que iniciam com a letra "A". Assim, a entrada pode ser solicitada para filtrar cidades baseado na letra de entrada "A". Em outro exemplo, para contrair o atributo dimensão "location", o usuário pode desejar receber

números de venda por estado, ao invés do que por cidade. A entrada recebida em 212, assim, pode ser uma solicitação para contrair o atributo dimensão "location" de "city" para "state". Ainda em outro exemplo, para expandir o atributo dimensão "location", o usuário pode desejar receber números de venda por voting ward, ao invés do que por cidade. A entrada recebida em 212, portanto, pode ser uma solicitação para expandir o atributo dimensão "location" de "city" para "voting ward". Ainda em outro exemplo, o usuário pode desejar receber informação de lucro junto com os dados de vendas. O usuário pode solicitar que o lucro seja retornado, de modo que o usuário receba tanto vendas por cidade como lucro por cidade. Este é um exemplo de adicionar uma medida. Adicionalmente ou alternativamente, o usuário pode desejar remover uma medida.

[00102] Em 214, a consulta é atualizada baseada na entrada recebida em 212. Em 216, dados adicionais são recebidos baseados na consulta atualizada em 214. Ou seja, a consulta atualizada é executada através do cubo e os resultados de tal execução são recebidos e apresentados para o usuário na tela de exibidor. Portanto, o usuário constrói a consulta por visualizar dados e identificar pelo menos uma operação a ser executada em relação aos dados. Este processo pode continuar até que dados desejados pelo usuário sejam adquiridos. Em 218, é feita uma determinação com respeito a se o usuário está proporcionando entrada adicional. Se o usuário proporcionar entrada adicional, o método retorna para 212, caso contrário, o método completa em 220.

[00103] Agora, se referindo à figura 3, é ilustrado um método ilustrativo 300 que facilita construir uma consulta para execução através de um cubo. A metodologia 300 inicia em 302, e em 304, um cubo é recebido. Em 306, a seleção de pelo menos um atributo dimensão (por exemplo, o atributo "days" para a dimensão "time") e

pelo menos uma medida é recebida. Em 308, uma consulta é construída baseada na seleção do pelo menos um atributo dimensão e da pelo menos uma medida. Em 310, dados são recebidos baseados na consulta. Como descrito acima, a consulta construída em 308 pode ser executada através do cubo, resultando no provisionamento de dados que podem ser apresentados em forma tabular em um exibidor. Em 312, é recebida uma solicitação para contrair ou expandir a pelo menos uma dimensão representada nos dados tabulares. Contrair a dimensão se refere a produzir coordenadas granuladas de forma menos refinada para um atributo menos refinado, enquanto expandir uma dimensão se refere a produção de coordenadas granuladas de forma mais refinada para um atributo mais granular. Em 314, a consulta é refinada baseada na solicitação para contrair ou expandir a pelo menos uma dimensão.

[00104] Adicionalmente, o cálculo da pelo menos uma medida é "flutuado" na consulta, por exemplo, de modo que a medida não seja calculada até após os valores de atributos para atributos dimensão apropriados serem recuperados. Isto ocorre automaticamente, independente do fato de que o usuário, em um estágio anterior de construção de consulta, solicitou que a medida fosse calculada (por exemplo, antes de solicitar a expansão ou a contração do atributo dimensão). A flutuação do cálculo da medida pode ser empreendido por identificar uma solicitação para um cálculo de medida, e mover o comando do cálculo da medida na consulta para uma posição de modo que o cálculo da medida ocorra após um atributo dimensão ser identificado.

[00105] É para ser observado que este ato é diferente, por exemplo, de adicionar comandos para o fim da consulta. Por exemplo, uma consulta ilustrativa inicial pode possuir a seguinte forma: DIMENSION=TIME; ATTRIBUTE=DAYS; MEASURE=SALES. Os

dados recuperados baseados em tal consulta podem ser vendas por dia. Quando a consulta é reformulada, a consulta pode possuir a forma: `DIMENSION=TIME; ATTRIBUTE=DAYS; ATTRIBUTE=WEEKS; MEASURE=SALES`. Assim, a seleção do atributo é colocada antes do comando de cálculo da medida – o cálculo da medida é "flutuado". Novamente, isto está em contraste com anexar a seleção de atributo no fim da consulta, como a seguir: `DIMENSION=TIME; ATTRIBUTE=DAYS; MEASURE=SALES; ATTRIBUTE=WEEKS`. A metodologia 300 completa em 316.

[00106] Agora, se referindo à figura 4, é ilustrada uma metodologia ilustrativa 400 que facilita mesclar cubos. A metodologia 400 inicia em 402, e em 404, uma solicitação para mesclar um primeiro cubo a partir de uma primeira fonte (opcionalmente de um primeiro formato) com um segundo cubo a partir de uma segunda fonte (e opcionalmente de um segundo formato) é recebida. Em 406, o primeiro cubo é mesclado com o segundo cubo para gerar um cubo mesclado. Em 408, é executada uma consulta através do cubo mesclado. Assim, cubos a partir de diferentes fontes de dados podem ser mesclados e uma única consulta pode ser executada através do cubo mesclado. A metodologia 400 completa em 410.

[00107] Agora, com referência à figura 5, é ilustrada uma interface gráfica com o usuário 500 ilustrativa que facilita a construção de uma consulta a ser executada através de um cubo. A interface gráfica com o usuário 500 compreende um ícone gráfico 502 que é representativo de um cubo (por exemplo, o cubo 104). Em resposta ao ícone gráfico 502 sendo selecionado, vários ícones 504 até 512 podem ser apresentados, onde os ícones 504 até 512 são representativos de objetos no cubo representado pelo ícone 502. Em um exemplo, o ícone gráfico 506 pode ser representativo de objetos relacionados com clientes de um negócio. A seleção do ícone gráfico 506 pode causar

que os ícones gráficos 514 e 516 seja apresentados, onde os ícones gráficos 514 e 516 são representativos de dimensões no cubo 104. A seleção do ícone gráfico 516 pode causar que vários ícones gráficos 518 até 524 seja apresentados, onde tais ícones gráficos 518 até 524 são representativos de respectivos atributos da dimensão representada pelo ícone 516. Os números apresentados em correspondência com os ícones na interface gráfica com o usuário 500 podem indicar para um usuário um número de objetos por baixo do ícone. Por exemplo, a dimensão representada pelo ícone 516 possui quatro atributos, e o número (4) é denotado na relação gráfica com o ícone 516. O usuário 110 pode navegar através dos objetos e selecionar atributos dimensão e medidas do cubo 104 que são de interesse para o usuário 110.

[00108] Agora, se referindo à figura 6, outra interface gráfica com o usuário ilustrativa 600 é ilustrada, onde a interface gráfica com o usuário 600 representa a seleção de alguns atributos dimensão e medidas pelo usuário 110. Em resposta à recepção de uma seleção do ícone 512, vários ícones 602 e 604 são apresentados, os quais identificam dois grupamentos de medidas (medidas relacionadas com compras e medidas relacionadas com produtos, respectivamente). Em resposta ao ícone 602 ser selecionado, vários ícones selecionáveis 606 até 610 são apresentados, onde os ícones 606 até 610 são respectivamente representativos de medidas. Medidas ilustrativas apresentadas na figura 6 sob o grupamento "purchase" incluem "ordered quantity", "received quantity", e "cost". Adicionalmente, em resposta ao ícone 508 sendo selecionado, os ícones 612 e 614 são apresentados, e em resposta ao ícone 64 sendo, o ícone 616 é apresentado. Os ícones 612 e 614 representam dimensões se relacionando com um produto (por exemplo, "product ID" e "category name"), e o ícone 616 representa um atributo "category name" para a

dimensão "category name" representada pelo ícone 614.

[00109] Como apresentado na interface gráfica com o usuário ilustrativa 600, o ícone 606 foi selecionado e assim, uma medida identificando quantidades ordenadas de mercadoria (por exemplo, "ordered quantity") é selecionada. Adicionalmente, os atributos "city" e "country" da dimensão "customer name" foram selecionados, e o atributo "category name" da dimensão "category name" foi selecionado. Uma vez que o usuário 110 tenha selecionado os atributos dimensão e as medidas desejadas, o usuário 110 pode selecionar um botão "load" 618. O componente receptor de entrada 115 (figura 1) pode receber a seleção dos atributos dimensão e as medidas (por exemplo, em resposta ao botão "load" 618 sendo selecionado), e o componente construtor de consulta 116 constrói uma consulta baseado na entrada recebida pelo componente receptor de entrada 115. O dispositivo de computação servidor 106 recebe a consulta, e o componente executor de consulta 126 executa a consulta através do cubo 104. O componente provedor de dados 128 recebe dados retornados baseados na execução da consulta através do cubo 104, e transmite os dados para o dispositivo de computação cliente 108, onde eles são colocados na memória 112 como os dados 117.

[00110] Agora, se referindo à figura 7, uma interface gráfica com o usuário ilustrativa 700 que inclui uma planilha é apresentada, onde o componente apresentador 120 apresenta os dados 117 recuperados baseados na consulta construída em formato tabular. A interface gráfica com o usuário 700 inclui os dados tabulares 702 que incluem colunas para os respectivos atributos dimensão selecionados e fileiras para medidas de valores dos atributos dimensão. Assim, continuando com as seleções do usuário ilustrativas descritas acima com referência à interface gráfica com o usuário 600, os dados tabulares 702 incluem uma primeira coluna 704 que é representativa do atributo "city" para a

dimensão "customer name", uma segunda coluna 706 que é representativa do atributo "country" para a dimensão "customer name", e uma terceira coluna 708 que é representativa do atributo category name" para a dimensão "category name". Uma quarta coluna 710 representa a medida "ordered quantity", e valores na quarta coluna 710 representam quantidades ordenadas de um produto possuindo os respectivos atributos apresentados nas colunas 704, 706 e 708.

[00111] Agora, se referindo à figura 8, uma interface gráfica com o usuário ilustrativa 800 de uma ferramenta editor de consulta que facilita construir e editar uma consulta é ilustrada. Por exemplo, em resposta a ser apresentada com os dados na planilha apresentada na figura 7, o usuário 110 pode desejar construir uma consulta que causar que dados diferentes sejam apresentados para o usuário 110. A interface gráfica com o usuário 800 inclui um campo 802 que expõe uma lista de etapas que foram executadas para adquirir os dados apresentados na interface gráfica com o usuário 700. Um campo 804 representa dados recuperados a partir do cubo 104 baseado nas etapas apresentadas no campo 802. Por exemplo, a etapa "source" no campo representa a seleção do cubo 104 representado pelo ícone 502. A etapa "expand dim1" apresentada no campo 802 representa a seleção de atributos dimensão "city" e "country" da dimensão "customer name". A etapa "expand dim2" representa a seleção do atributo dimensão "category name" da dimensão "category name". Como será descrito neste documento, as etapas apresentadas no campo 802 são selecionáveis, e dados apresentados no campo 804 alteram à medida que diferentes etapas são selecionadas.

[00112] Agora, se referindo à figura 9, outra interface gráfica com o usuário ilustrativa 900 da ferramenta editor de consulta é apresentada. Neste exemplo, o usuário 110 selecionou a etapa "expand dim1" a partir da lista de etapas representadas no campo 802. Em resposta à

etapa "expand dim1" sendo selecionada, o aplicativo BI pode atualizar conteúdos apresentados no campo 804, de modo que valores de atributo para atributos dimensão "city" e "country" a partir da dimensão "customer name" são apresentados, mas os valores de atributo para o atributo dimensão "category name" e para as medidas não são representados no campo 804.

[00113] Agora, se referindo à figura 10, outra interface gráfica com o usuário ilustrativa 1000 da ferramenta editor de consulta é apresentada. Aqui, o usuário 110 selecionou a terceira etapa ("expand dim2") utilizada para construir a consulta a partir do campo 802, e os conteúdos do campo 804 são atualizados para apresentar dados quando a consulta é construída baseada nas primeiras três etapas (mas não a etapa "add measure1"). Ou seja, o componente construtor de consulta 16 constrói a consulta baseado nas primeiras três etapas, e o componente executor de consulta 126 executa tal consulta através do cubo 104. À medida que a terceira etapa se relaciona com expandir o nome da categoria, a terceira coluna 708 é apresentada. As figuras 9 e 10 foram apresentadas para ilustrar que o usuário 110, quando construindo e/ou editando uma consulta através da abordagem gradual neste documento, pode retornar para inserir uma nova etapa de consulta, modificar uma etapa de consulta anteriormente executada, etc.

[00114] Com referência agora à figura 11, outra interface gráfica com o usuário ilustrativa 1100 da ferramenta editor de consulta é apresentada, onde o usuário 110 selecionou a etapa de construção de consulta "add measure1", a qual representa seleção da medida "ordered quantity". O conteúdo do campo 804 é atualizado em resposta à etapa mencionada acima sendo selecionada para apresentar valores de medida para os valores de atributo dos atributos dimensão. Novamente, o usuário pode inserir etapas de construção de

consulta entre as etapas apresentadas no campo 802, pode apagar etapas de construção de consulta das etapas apresentadas no campo 802, adicionar etapas de construção de consulta após a última etapa de construção de consulta apresentada no campo 802, etc.

[00115] Agora, se referindo à figura 12, uma interface gráfica com o usuário ilustrativa 1200 que pode ser utilizada para expor uma versão baseada em texto da consulta construída por meio da ferramenta editor de consulta é exibida. Por editar texto apresentado na figura 2, o usuário 110 pode escolher modificar a consulta construída por meio de inserir texto, remover texto, et. O editor baseado em texto pode ser particularmente bem adequado para usuários que estão familiarizados com a linguagem de consulta empregada pelo aplicativo de inteligência comercial 112.

[00116] Agora, com referência à figura 12, uma interface gráfica com o usuário ilustrativa 1300 que ilustrar contração de um atributo dimensão é representada. Como apresentado na figura 11, a medida "order quantity" foi calculada com respeito à dimensão "customer name" para os atributos dimensão "city" e "country", onde o atributo dimensão "city" é mais granular do que o atributo dimensão "country". Retornando para a figura 13, a contração da dimensão de "city" para "country" causa que os valores da medida "ordered quantity" sejam acumulado para o atributo dimensão "country". Isto é representado pela etapa "collapse dim1" no campo 802.

[00117] A construção de uma consulta utilizando a abordagem de etapa por etapa descrita neste documento quando atributos dimensão são contraídos ou expandidos é um processo não trivial, à medida que uma função para contrair ou expandir atributos dimensão não pode ser aplicada em sequência com as etapas de construção de consulta anteriores. O componente construtor de consulta 116 constrói a consulta, como descrito acima, por "flutuar" o cálculo da medida para a

traseira da expressão de consulta. Não existe análogo para esta abordagem em uma configuração de base de dados relacional. Ou seja, remover colunas a partir de uma tabela em uma configuração de base de dados relacional não causam que as fileiras sejam removidas,

[00118] Continuando com este exemplo, o componente consultor de consulta 116 constrói a consulta (antes da etapa collapse dim1") por primeiro definir uma primeira expressão que recupera os atributos dimensão identificados, e então definindo uma segunda expressão que calcula a medida identificada para os valores de atributo dos atributos dimensão identificados. Se o componente construtor de consulta 116 tentar definir uma terceira expressão que contrai uma dimensão atributo, onde a terceira função executa após a primeira e a segunda funções (por exemplo, mapear para a ordem das etapas no campo 802), a medida ainda seria calculada através do atributo dimensão de granulação mais refinada ("city"), ao invés do que através do atributo dimensão (mais grosseiro) desejado "country". No caso de contrações e expansões, ao invés de linearmente adicionar expressões, as expressões para calcular a medida são fluída para o exterior da consulta, de modo que a expressão é executada após as dimensões de atributo desejadas terem sido identificadas. A tabela resultante 1302 apresentada no campo 804 inclui duas colunas; uma primeira coluna 1304 correspondendo ao atributo dimensão "country" para a dimensão "customer name", e uma segunda coluna 1306 que identifica valores de medida calculados para os valores de atributo apresentados na primeira coluna 1304.

[00119] Com referência à figura 14, outra interface gráfica com o usuário ilustrativa 1400 é apresentada. Pode ser verificado que o usuário 110 selecionou uma etapa anterior de construção de consulta no campo 802 (por exemplo, a etapa "add measure1"). Assim, por exemplo, o usuário 110 pode desejar modificar a consulta antes da

consulta causar que o atributo dimensão "city" ser contraído para o atributo dimensão "country". Por exemplo, o usuário 110 pode selecionar uma célula particular 1402 que possui um primeiro valor na primeira coluna 704, o que pode causar que uma janela instantânea 1404 seja apresentada para o usuário 110. A janela instantânea 1404 pode incluir opções selecionáveis para filtrar resultados a partir dos dados tabulares 702. Por exemplo, o usuário 110 pode indicar que ele deseja filtrar quaisquer colunas nos dados tabulares 702 possuindo o primeiro valor a partir dos dados tabulares 702. Em outro exemplo, o usuário 110 pode indicar que ele deseja somente receber fileiras nos dados tabulares 702 que possuam o primeiro valor, dentre outras opções de filtro. A operação de filtro selecionada pelo usuário 110 a partir da janela instantânea 1404 pode então ser adicionada como uma etapa no processo de construção de consulta no campo 802, e pode ser apresentada imediatamente subsequente à etapa selecionada (por exemplo, após a etapa "add measure1", mas antes da etapa "collapse dim1"). O componente construtor de consulta 116, em resposta ao componente receptor de entrada 114 recebendo a indicação de que o usuário 110 selecionou o filtro, pode atualizar a consulta construída.

[00120] Referindo-se à figura 15, o usuário 110 então pode selecionar a última etapa nas etapas da consulta apresentadas no campo 802 (por exemplo, a etapa "collapse dim1"), o que causa que o componente construtor de consulta 116 transmita a consulta refinada para o componente executor de consulta 126, o qual executa a consulta refinada através do cubo 104. O componente provedor de dados 128 retorna os dados 117, os quais são retidos na memória 112 do dispositivo de computação cliente 108. O componente apresentado 120 apresenta os dados 117 em formato tabular no campo 804, onde os dados tabulares 1502 ilustram dados retornados por executar a consulta. Por exemplo, um valor em uma célula 1504 foi atualizado

(quando comparado com o valor correspondente na célula apresentada na figura 13) devido à etapa de filtragem sendo incluída na consulta.

[00121] Referindo-se à figura 16, outra interface gráfica com o usuário ilustrativa 1600 de uma ferramenta editor de consulta baseada em texto é apresentada. A interface gráfica com o usuário 1600 representa uma representação textual da consulta construída da maneira exposta acima.

[00122] Agora, vários exemplos são expostos.

[00123] Exemplo 1: Um sistema de computação compreendendo: um processador; e uma memória que compreende um aplicativo de inteligência comercial (BI) que é executado pelo processador, o aplicativo BI é configurado para: construir uma consulta baseada nas modificações incrementais para versões anteriores da consulta, a consulta representada como uma sequência de etapas da consulta, cada etapa na sequência de etapas da consulta corresponde a uma respectiva modificação incremental nas modificações incrementais; e recuperar dados tabulares a partir de um cubo de dados baseado na consulta.

[00124] Exemplo 2: O sistema de computação de acordo com o exemplo 1, o aplicativo BI compreende um componente construtor de consulta que recebe uma modificação incremental e constrói a consulta baseado: na modificação incremental e em uma sequência de modificações incrementais anteriormente recebidas, onde o componente construtor de consulta expressa a consulta como vários operadores relacionais.

[00125] Exemplo 3: Sistema de computação de acordo com o exemplo 2, o componente construtor de consulta normaliza os vários operadores para padrões de árvore de expressão predefinidos.

[00126] Exemplo 4: O sistema de computador de acordo com o

exemplo 1, o aplicativo BI compreende um componente construtor de consulta que recebe uma modificação incremental para a consulta, a modificação incremental para a consulta sendo uma solicitação para um dentre expandir uma dimensão no cubo ou contrair a dimensão no cubo, o componente construtor de consulta constrói a consulta baseado na modificação incremental.

[00127] Exemplo 5: O sistema de computador de acordo com o exemplo 4, a consulta compreende uma segunda modificação incremental, a segunda modificação incremental sendo uma solicitação para calcular uma medida para uma dimensão, o componente construtor de consulta constrói a consulta de modo que a medida seja calculada subsequente a um atributo da dimensão sendo selecionado.

[00128] Exemplo 6: O sistema de computação de acordo com o exemplo 5, a segunda modificação incremental ocorrendo antes da modificação incremental.

[00129] Exemplo 7: O sistema de computação de acordo com qualquer um dentre os exemplos 1 até 6, adicionalmente compreendendo um componente apresentado que apresentar os dados tabulares recuperados a partir do cubo de dados em um exibidor, o componente apresentado adicionalmente apresenta a consulta no exibidor.

[00130] Exemplo 8: O sistema de computação de acordo com o exemplo 7, o componente apresentador apresenta uma sequência de etapas da consulta no exibidor.

[00131] Exemplo 9: O sistema de computação de acordo com o exemplo 8, o aplicativo BI compreende um componente receptor de entrada que recebe uma seleção de uma etapa anterior da consulta nas etapas de consulta, onde em resposta ao componente receptor de entrada recebendo a seleção, o componente apresentado apresenta

segundos dados tabulares a partir do cubo de dados no exibidor, os segundos dados tabulares recuperados baseados na consulta na etapa anterior nas etapas da consulta.

[00132] Exemplo 10: O sistema de computação de acordo com o exemplo 9, o componente receptor de entrada recebe uma modificação incremental intermediária subsequente ao componente receptor de entrada recebendo a seleção da etapa anterior da consulta, o componente construtor de consulta constrói a consulta para adicionar outra etapa da consulta subsequente à etapa anterior da consulta e antes da última etapa da consulta na sequência de etapas da consulta.

[00133] Exemplo 11: O sistema de computação de acordo com qualquer um dos exemplos 1 até 10 compreendido por um dispositivo de computação servidor que é acessível por meio de um navegador da Rede.

[00134] Exemplo 12: Um método executado por um processador do computador, o método compreende: apresentar dados tabulares em um exibidor, os dados tabulares recuperados a partir de um cubo de dados baseados em uma etapa da consulta anteriormente emitida; receber uma etapa subsequente da consulta; construir uma consulta baseado na etapa da consulta anteriormente emitida e na etapa subsequente da consulta; recuperar dados tabulares atualizados a partir do cubo de dados baseado na consulta; apresentar os dados tabulares atualizados no exibidor em resposta a recuperar os dados tabulares atualizados.

[00135] Exemplo 13: O método de acordo com o exemplo 12, a etapa da consulta anteriormente emitida causa que uma medida seja calculada em relação a um primeiro atributo de uma dimensão, e a etapa subsequente da consulta causa que a medida seja calcula em relação a um segundo atributo da dimensão.

[00136] Exemplo 14: O método de acordo com o exemplo 13, adicionalmente compreendendo: em resposta a receber a etapa subsequente da consulta, construir a consulta de modo que a medida seja calculada em relação ao segundo atributo da dimensão após o segundo atributo da dimensão ser especificado na consulta.

[00137] Exemplo 15: O método de acordo com qualquer um dos exemplos 12 até 14, adicionalmente compreendendo apresentar uma sequência de etapas da consulta no exibidor, a sequência de etapas da consulta compreendendo a etapa da consulta anteriormente emitida e a etapa subsequente da consulta, cada etapa da consulta na sequência de etapas da consulta sendo selecionável.

[00138] Exemplo 16: O método de acordo com o exemplo 14, adicionalmente compreendendo: receber uma seleção de uma etapa da consulta na sequência de etapas da consulta; construir a consulta baseado na seleção da etapa da consulta na sequência de etapas da consulta; e apresentar dados tabulares que correspondem à consulta até a etapa da consulta selecionada.

[00139] Exemplo 17: O método de acordo com qualquer um dos exemplos 12 até 16, a etapa subsequente da consulta sendo uma solicitação para contrair ou expandir pelo menos uma dimensão, o método compreendendo refinar a consulta baseado na solicitação para contrair ou expandir a pelo menos uma dimensão.

[00140] Exemplo 18: O método de acordo com qualquer um dos exemplos 12 até 17, onde a construção da consulta compreende: transformar a etapa anterior da consulta e a etapa subsequente da consulta em vários operadores relacionais; e normalizar os vários operadores operacionais baseado em um padrão predefinido.

[00141] Exemplo 19. O método de acordo com qualquer um dos exemplos 12 até 18, adicionalmente compreendendo: receber várias modificações incrementais para a consulta; e para cada modificação

incremental: construir a consulta; e recuperar dados tabulares baseados em uma respectiva modificação incremental.

[00142] Exemplo 20: Um meio de armazenamento legível por computador compreendendo instruções que, quando executadas por um processador, causam que o processador execute atos compreendendo: receber uma consulta; em resposta a receber a consulta, recuperar dados tabulares a partir de um cubo de dados; em resposta a recuperar os dados tabulares a partir do cubo de dados, apresentar os dados tabulares e uma sequência de etapas da consulta em um exibidor, a sequência de etapas da consulta representativa da consulta, os dados tabulares compreendem uma medida calculada em relação a um primeiro atributo de uma dimensão no cubo de dados; receber uma modificação incremental para a consulta, a modificação incremental sendo uma solicitação para calcular a medida em relação a um segundo atributo da dimensão no cubo de dados; e em resposta a receber a modificação incremental para a consulta, recuperar segundos dados tabulares baseados na modificação incremental para a consulta, os segundos dados incrementais compreendem a medida calculada em relação ao segundo atributo da dimensão.

[00143] Exemplo 21: Um sistema implementado por computador, compreendendo: dispositivo para apresentar dados tabulares em um exibidor, os dados tabulares recuperados a partir de um cubo de dados baseados em uma etapa de consulta anteriormente emitida; dispositivo para receber uma etapa subsequente da consulta; dispositivo para construir uma consulta baseado na etapa da consulta anteriormente emitida e na etapa subsequente da consulta; dispositivo para recuperar dados tabulares atualizados a partir do cubo de dados baseado na consulta; e dispositivo para apresentar os dados tabulares atualizados no exibidor em resposta a recuperar os dados tabulares atualizados.

[00144] Agora, se referindo à figura 17, uma ilustração de alto nível de um dispositivo de computação ilustrativo 1700 que pode ser utilizado de acordo com os sistemas e metodologias revelados neste documento é ilustrado. Por exemplo, o dispositivo de computação 1700 pode ser utilizado em um sistema que suporte construção e refinamento de uma consulta para execução através de um cubo de dados. A título de outro exemplo, o dispositivo de computação 1700 pode ser utilizado em um sistema que suporte apresentação de dados extraídos a partir de um cubo. O dispositivo de computação 700 inclui pelo menos um processador 1702 que executa instruções que são armazenados em uma memória 1704. As instruções podem ser, por exemplo, instruções para implementar funcionalidade descrita como sendo realizada por um ou mais componentes discutidos acima ou instruções para implementar um ou mais dos métodos descritos acima. O processador 1702 pode acessar a memória 1704 por meio de um barramento do sistema 1706. Em adição a armazenar instruções executáveis, a memória 1704 também pode armazenar tabelas de fato, tabelas de dimensão, informação hierárquica, etc.

[00145] O dispositivo de computação 1700 adicionalmente inclui um depósito de dados 1708 que é acessível pelo processador 1702 por meio do barramento do sistema 1706. O depósito de dados 708 pode incluir instruções executáveis, um cubo, uma fração do cubo, etc. O dispositivo de computação 1700 também inclui uma interface de entrada 1710 que permite que dispositivos externos se comuniquem com o dispositivo de computação 1700. Por exemplo, a interface de entrada 1710 pode ser utilizada para receber instruções a partir de um dispositivo de computador externo, a partir de um usuário, etc. O dispositivo de computação 1700 também inclui uma interface de saída 1712 que faz interface do dispositivo de computação 1700 com um ou mais dispositivos externos. Por exemplo, o dispositivo de computação

1700 pode exibir texto, imagens, etc., por meio da interface de saída 1712.

[00146] É contemplado que os dispositivos externos que se comunicam com o dispositivo de computação 1700 via a interface de entrada 1710 e a interface de saída 1712 possam estar incluídos em um ambiente que proporciona substancialmente qualquer tipo de interface com o usuário com a qual um usuário pode interagir. Exemplos de tipos de interface com o usuário incluem interfaces gráficas com o usuário, interfaces naturais com o usuário, dentre outras. Por exemplo, uma interface gráfica com o usuário pode aceitar entrada a partir de um usuário empregando dispositivos de entrada tais como um teclado, mouse, controle remoto, dentre outros e proporciona saída em um dispositivo de saída tal como um exibidor. Adicionalmente, uma interface natural com o usuário pode permitir a um usuário interagir com o dispositivo de computação 700 de uma maneira livre de restrições impostas pelo dispositivo de entrada tais como teclados, mouses, controles remotos, dentre outros. Ao invés disso, uma interface natural com o usuário pode contar com o reconhecimento de fala, reconhecimento de toque e caneta stylus, reconhecimento de gesto tanto na tela como adjacente à tela, gestos no ar, rastreamento de cabeça e olho, voz e fala, visão, toque, gestos, inteligência artificial, dentre outros.

[00147] Adicionalmente, apesar de ilustrado como um sistema único, é para ser entendido que o dispositivo de computação 1700 pode ser um sistema distribuído. Assim, por exemplo, vários dispositivos podem estar em comunicação por meio de uma conexão de rede e podem coletivamente executar tarefas descritas como sendo executadas pelo dispositivo de computação 1700.

[00148] Várias funções descritas neste documento podem ser implementadas em hardware, software, ou em qualquer combinação

dos mesmos. Se implementadas em software, as funções podem ser armazenadas ou transmitidas como uma ou mais instruções ou código em um meio legível por computador. O meio legível por computador inclui meio de armazenamento legível por computador. Um meio de armazenamento legível por computador pode ser qualquer meio de armazenamento disponível que possa ser acessado por um computador. A título de exemplo e não de limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco ótico, armazenamento em disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser utilizado para transportar ou armazenar código de programa desejado na forma de instruções ou de estruturas de dados e que possa ser acessado por um computador. Disco rígido e disco, como utilizados neste documento, incluem disco compacto (CD), disco a laser, disco ótico, disco versátil digital (DVD), disco flexível, e disco Blu-ray (BD), onde discos rígidos normalmente reproduzem dados magneticamente e discos normalmente reproduzem dados de forma ótica com lasers. Adicionalmente, um sinal propagado não está incluído dentro do escopo de meio de armazenamento legível por computador. O meio legível por computador também inclui meio de comunicação incluindo qualquer meio que facilite a transferência de um programa de computador de um local para outro. Uma conexão, por exemplo, pode ser um meio de comunicação. Por exemplo, se o software for transmitido a partir de um site da Rede, servidor, ou de outra fonte remota utilizando um cabo coaxial, cabo de fibra ótica, par trançado, linha de assinante digital (DSL), ou tecnologias sem uso de fios tais como infravermelho e microondas, então o cabo coaxial, o cabo de fibra ótica, o par trançado, DSL, ou tecnologias sem uso de fios tais como infravermelho, rádio e microondas estão incluídas na definição

de meio de comunicação. Combinações do dito acima também devem estar incluídas dentro do escopo de meio legível por computador.

[00149] Alternativamente, ou em adição, a funcionalidade descrita neste documento pode ser executada, pelo menos em parte, por um ou mais componentes de lógica de hardware. Por exemplo, e sem limitação, tipos ilustrativos de componentes de lógica de hardware que podem ser utilizados incluem Arranjos de Portas Programáveis em Campo (FPGAs), Circuitos Integrados de Programa Específico (ASICs), Produtos Padrão de Programa Específico (ASSPs), sistemas de Sistema em um chip (SOCs), Dispositivos de Lógica Complexa Programável (CPDLs), etc.

[00150] O que foi descrito acima inclui exemplos de uma ou mais concretizações. Obviamente, não é possível descrever todas as modificações e alteração concebível dos dispositivos ou das metodologias para propósito de descrever os aspectos mencionados acima, mas, os versados na técnica podem reconhecer que várias modificações e permutações adicionais de vários aspectos são possíveis. Por consequência, os aspectos descritos são pretendidos para abranger todas tais alterações, modificações e variações que se situem dentro do espírito e do escopo das concretizações anexas. Adicionalmente, até a extensão em que o termo "inclui" é utilizado nos detalhes, descrição ou nas concretizações, tal termo é pretendido para ser inclusivo de uma maneira similar ao termo "compreendendo" como "compreendendo" é interpretado quando empregado como uma palavra de transição em uma concretização.

REIVINDICAÇÕES

1. Sistema de computação (100) **caracterizado pelo fato de que** compreende:

um processador (111); e

memória (112) que compreende um método que é executado pelo processador (111), em que o processador (111), quando executando o método, é configurado para:

construir uma consulta (116) com base em modificações incrementais para versões anteriores da consulta, a consulta (116) representada como uma sequência de etapas de consulta em um exibidor, cada etapa na sequência de etapas de consulta corresponde a uma modificação incremental nas modificações incrementais;

em resposta a recebimento de uma seleção de uma etapa de consulta intermediária na sequência de etapas de consulta, recuperar dados tabulares (117) a partir de um cubo de dados (104) e exibir os dados tabulares (117) no exibidor, os dados tabulares (117) recuperados com base na consulta na etapa de consulta intermediária;

receber uma modificação incremental para a consulta (116) enquanto os dados tabulares (117) são exibidos;

modificar a consulta (116) com base na modificação incremental para adicionar outra etapa de consulta à consulta, em que a etapa de consulta é adicionada subsequente à etapa de consulta intermediária e antes de uma última etapa de consulta da consulta (116); e

recuperar segundos dados tabulares a partir do cubo de dados (104) em resposta a modificar a consulta.

2. Sistema de computação (100), de acordo com a reivindicação 1, **caracterizado pelo fato de que** a consulta (116) é

modificada com base na modificação incremental e em uma sequência de modificações incrementais anteriormente recebidas, em que a consulta (116) é expressa como uma pluralidade de operadores relacionais.

3. Sistema de computação (100), de acordo com a reivindicação 2, **caracterizado pelo fato de que** o processador (111), quando executando o aplicativo, é ainda configurado para normalizar a pluralidade de operadores relacionais para padrões de árvore de expressão pré-definidos.

4. Sistema de computação (100), de acordo com a reivindicação 1, **caracterizado pelo fato de que** a modificação incremental para a consulta (116) sendo uma solicitação de expansão de uma dimensão no cubo de dados (104) ou contração da dimensão no cubo de dados (104).

5. Sistema de computação (100), de acordo com a reivindicação 4, **caracterizado pelo fato de que** a consulta (116) compreende uma segunda etapa de consulta que corresponde a uma segunda modificação incremental, a segunda modificação incremental sendo uma solicitação para calcular uma medida para a dimensão, em que a consulta (116) é construída de modo que a medida seja calculada subsequente a um atributo da dimensão sendo selecionada.

6. Sistema de computação (100), de acordo com a reivindicação 5, **caracterizado pelo fato de que** a segunda etapa de consulta sendo anterior à outra etapa de consulta na consulta (116).

7. Sistema de computação (100), de acordo com a reivindicação 1, **caracterizado pelo fato de que** o processador (111), quando executando o aplicativo, é configurado para exibir os segundos dados tabulares no exibidor em resposta a recuperar os segundos dados tabulares a partir do cubo de dados (104).

8. Sistema de computação (100), de acordo com a reivindicação 1, **caracterizado pelo fato de que** é compreendido por um dispositivo de computação de servidor (106) que é acessível por meio de um navegador da rede.

9. Método executado por um processador (111) de computador, o método **caracterizado pelo fato de que** compreende:

construir uma consulta (116) com base em modificações incrementais para versões anteriores da consulta, a consulta (116) representada como uma sequência de etapas de consulta em um exibidor, cada etapa na sequência de etapas de consulta corresponde a uma modificação incremental nas modificações incrementais;

receber uma seleção de uma etapa de consulta intermediária na sequência de etapas de consulta;

em resposta ao recebimento da seleção e com base na etapa de consulta intermediária, recuperar dados tabulares (117) a partir de um cubo de dados (104) e exibir os dados tabulares (117) no exibidor;

receber uma modificação incremental para a consulta (116) enquanto os dados tabulares (117) são exibidos;

modificar a consulta (116) com base na modificação incremental para adicionar outra etapa de consulta à consulta (116), em que a etapa de consulta é adicionada subsequente à etapa de consulta intermediária e antes de uma última etapa de consulta da consulta (116); e

recuperar segundos dados tabulares a partir do cubo de dados (104), os segundos dados tabulares recuperados com base na etapa de consulta intermediária e na outra etapa de consulta.

10. Método, de acordo com a reivindicação 9, **caracterizado pelo fato de que** a etapa de consulta intermediária, quando realizada pelo processador (111), faz com que uma medida

seja calculada em relação a um primeiro atributo de uma dimensão, e a outra etapa de consulta, quando realizada pelo processador (111) faz com que a medida seja calculada em relação a um segundo atributo da dimensão.

11. Método, de acordo com a reivindicação 10, **caracterizado pelo fato de que** ainda compreende:

em resposta a receber a outra etapa de consulta, modificar a consulta (116) de modo que o processador (111), quando executando a consulta (116) em relação ao cubo de dados (104), calcule a medida em relação ao segundo atributo da dimensão após o segundo atributo da dimensão ser especificado na consulta (116).

12. Método, de acordo com a reivindicação 9, **caracterizado pelo fato de que** cada etapa de consulta na sequência de etapas de consulta é selecionável.

13. Método, de acordo com a reivindicação 9, **caracterizado pelo fato de que** ainda compreende recuperar os segundos dados tabulares com base em cada etapa de consulta na consulta (116) até a outra etapa de consulta.

14. Método, de acordo com a reivindicação 13, **caracterizado pelo fato de que** etapas de consulta depois da outra etapa de consulta na consulta (116) não são consideradas quando recuperar os segundos dados tabulares.

15. Método, de acordo com a reivindicação 9, **caracterizado pelo fato de que** a outra etapa de consulta sendo uma solicitação para contrair ou expandir pelo menos uma dimensão, o método ainda compreendendo refinar a consulta (116) com base na solicitação para contrair ou expandir a pelo menos uma dimensão.

16. Método, de acordo com a reivindicação 9, **caracterizado pelo fato de que** ainda compreende:

transformar a etapa de consulta incremental e a outra etapa de consulta em uma pluralidade de operadores relacionais; e normalizar a pluralidade de operadores relacionais com base em um padrão pré-definido.

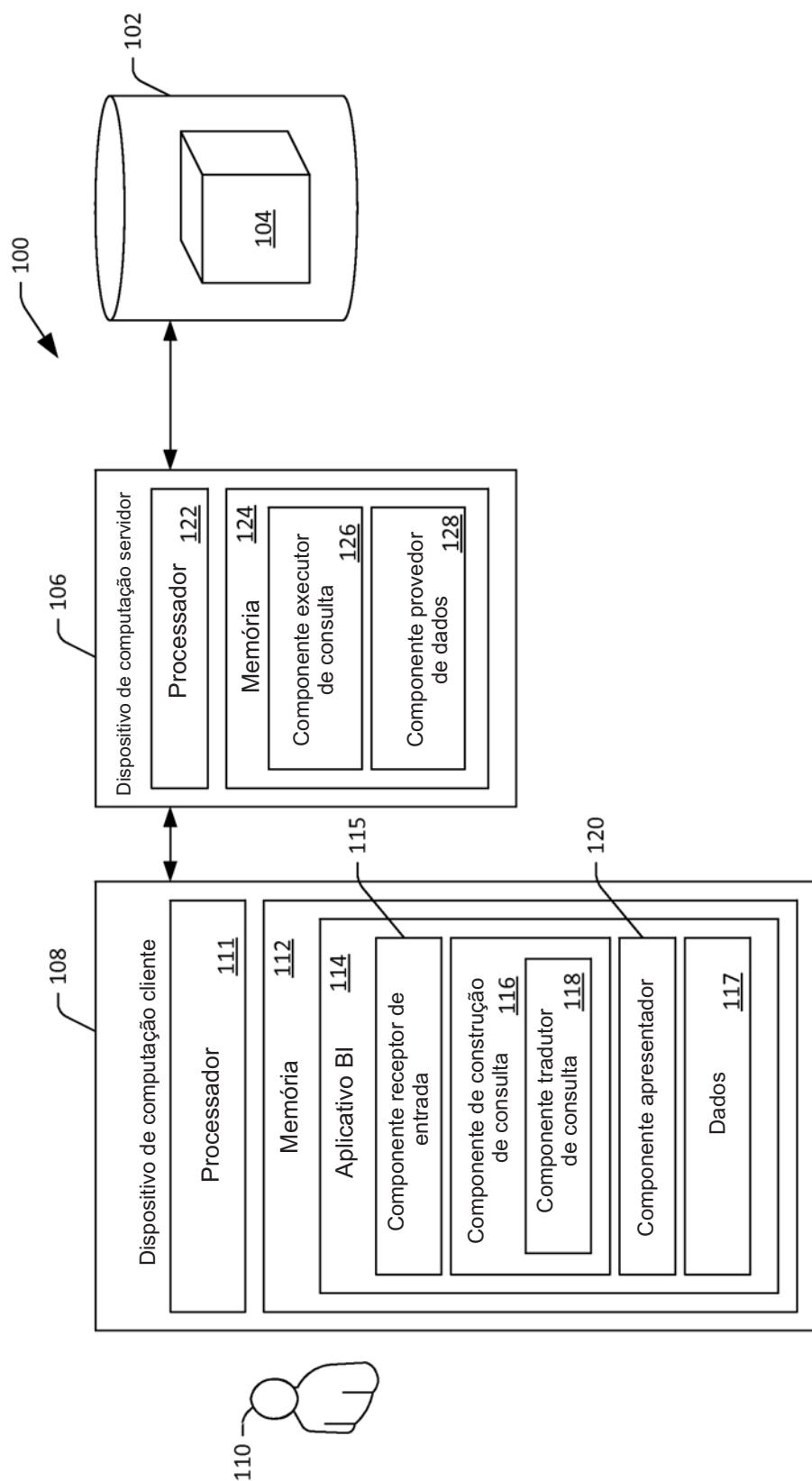
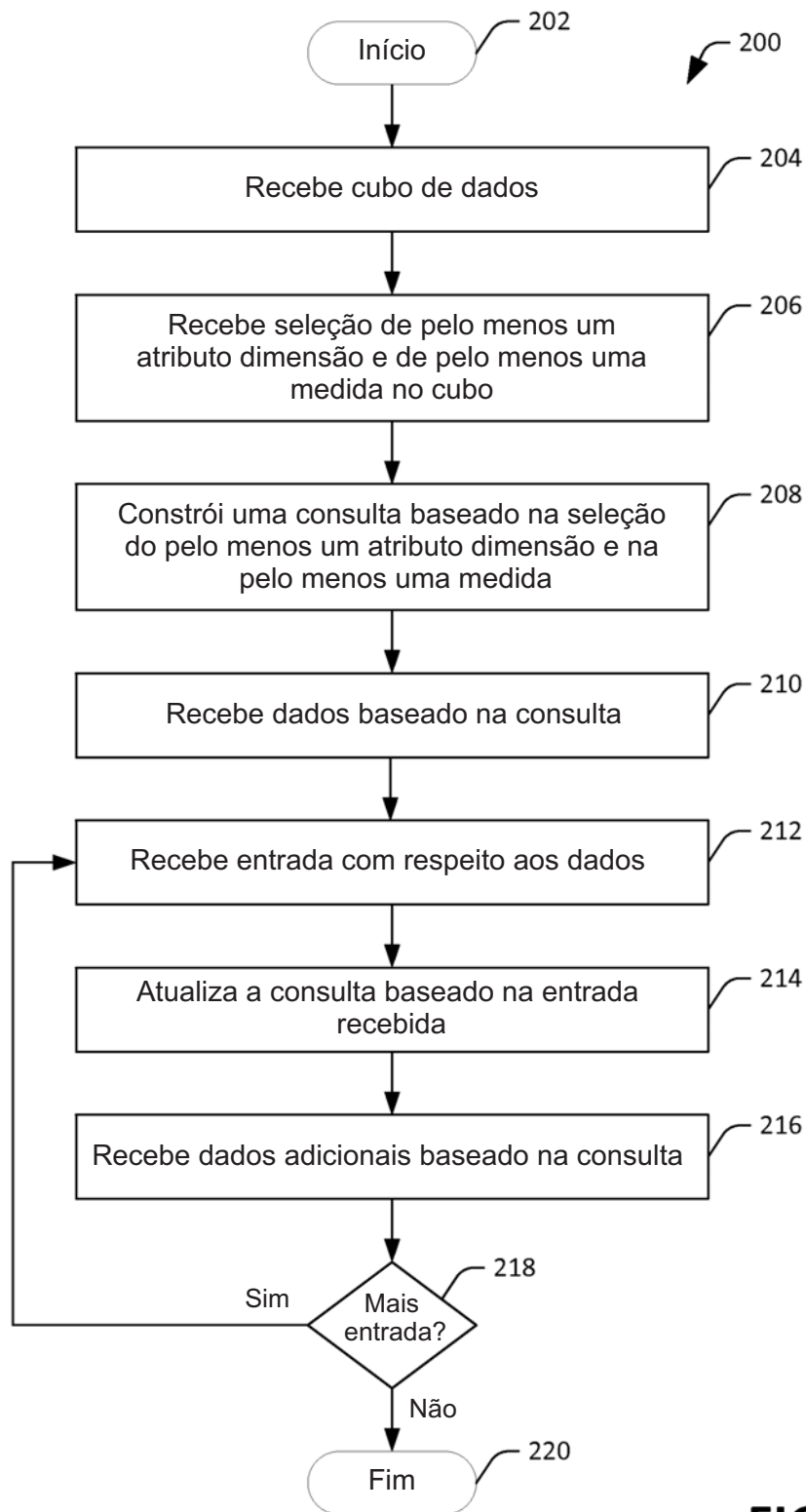
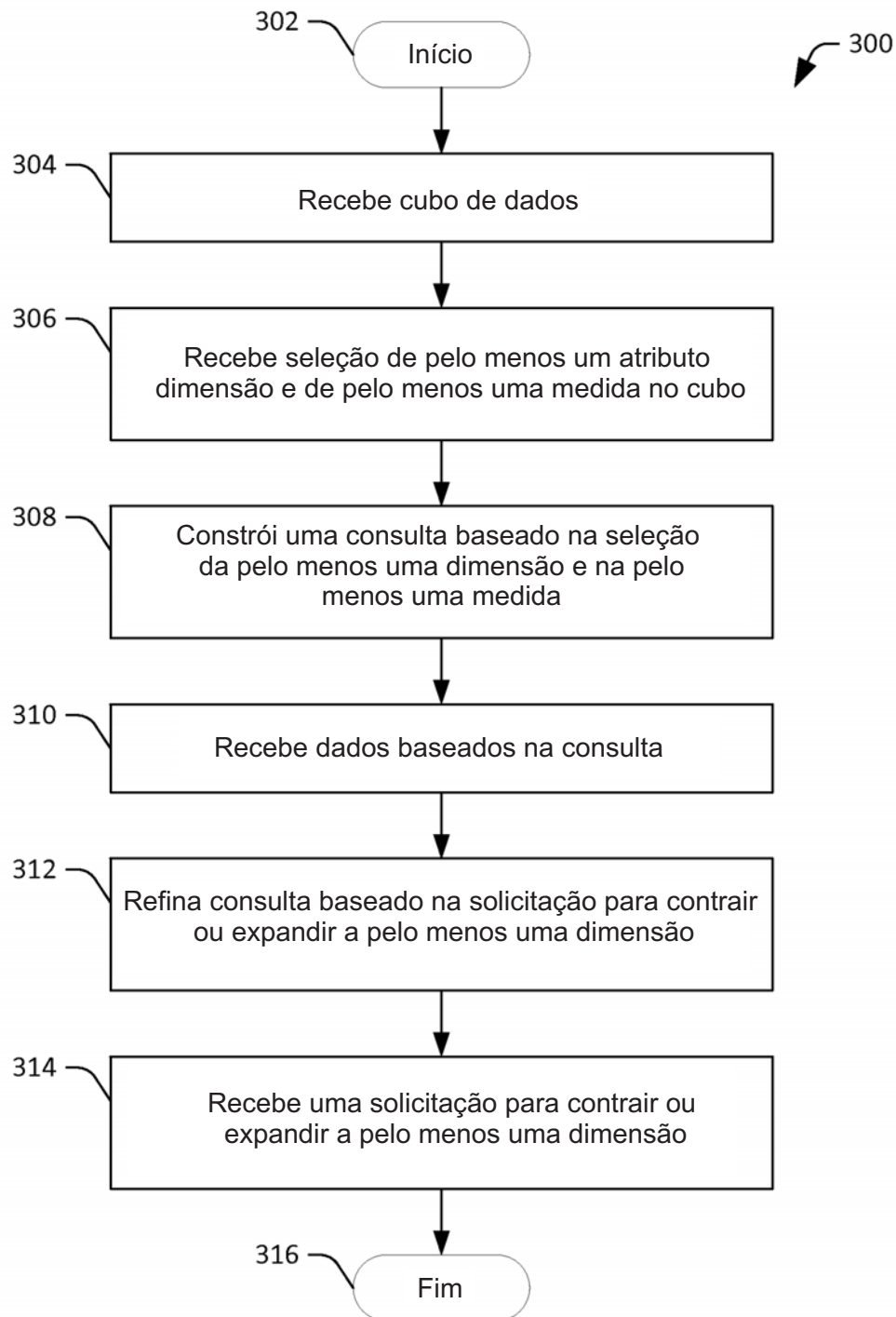
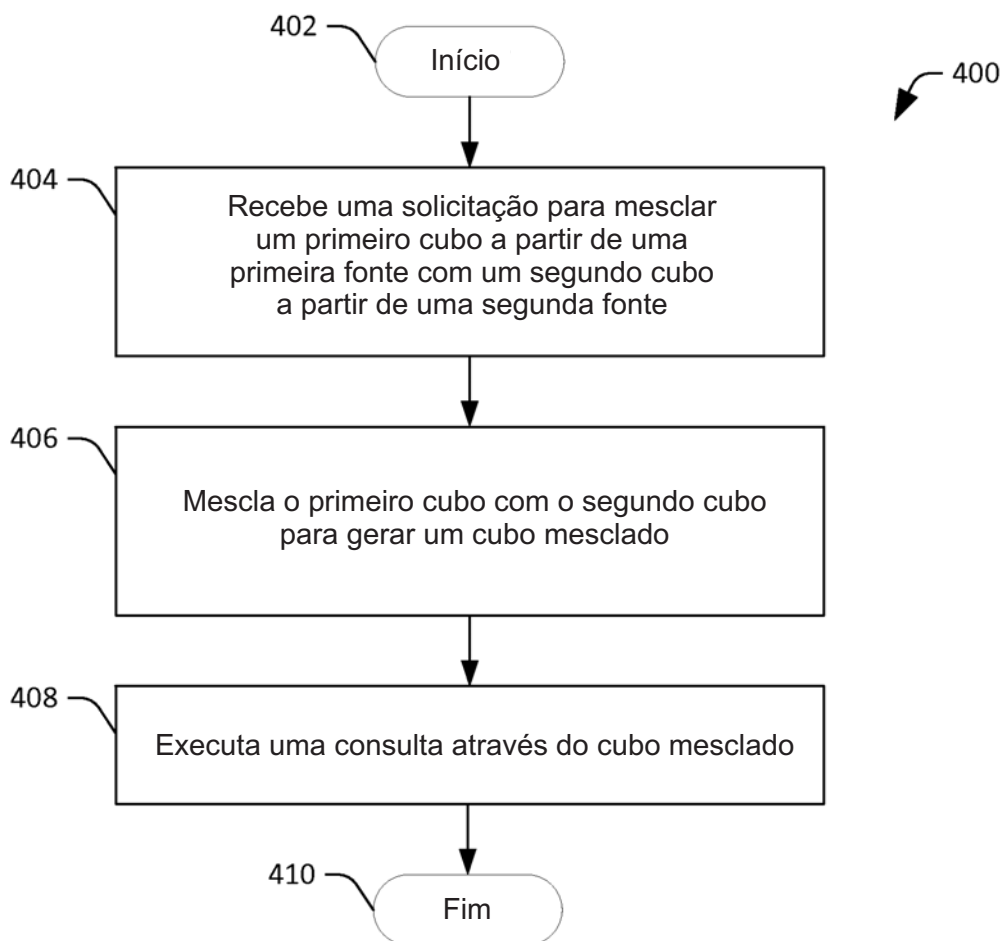
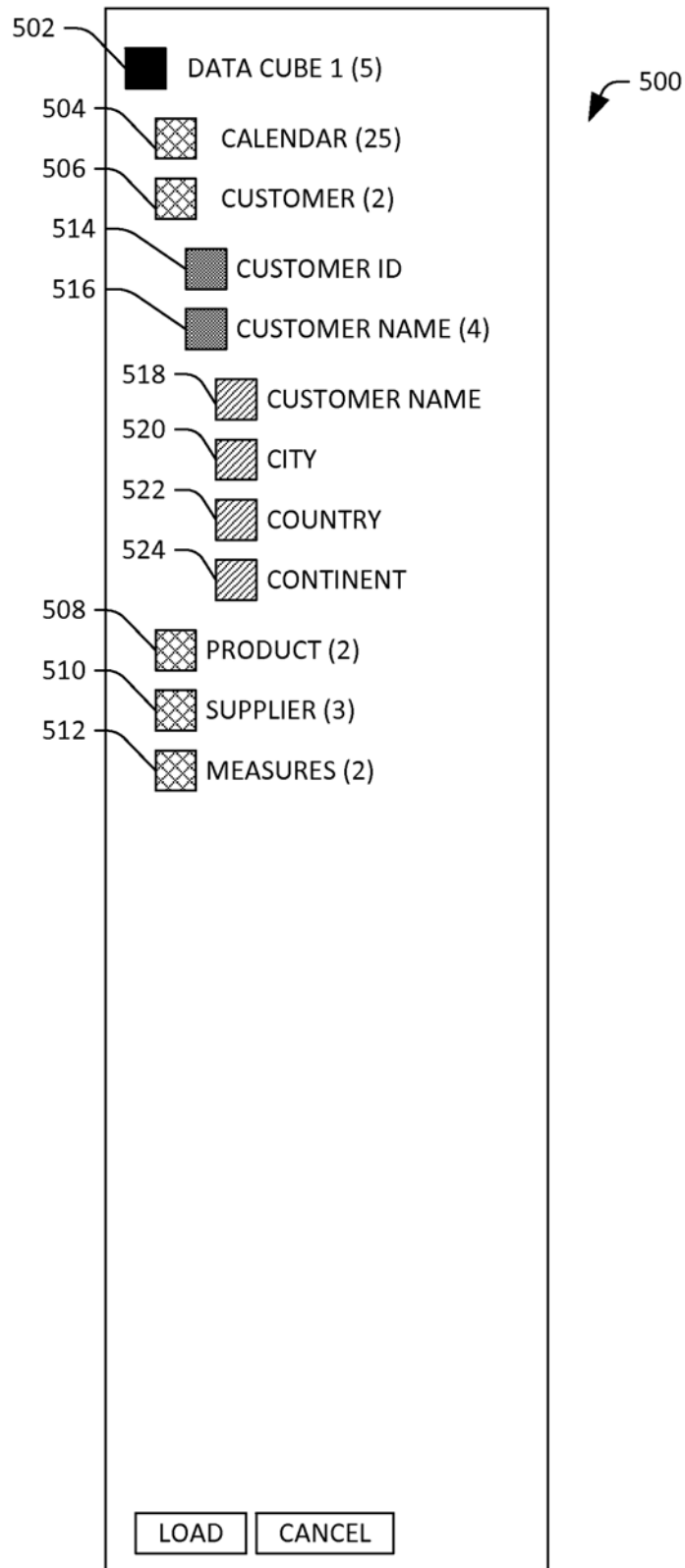


FIG. 1

**FIG. 2**

**FIG. 3**

**FIG. 4**

**FIG. 5**

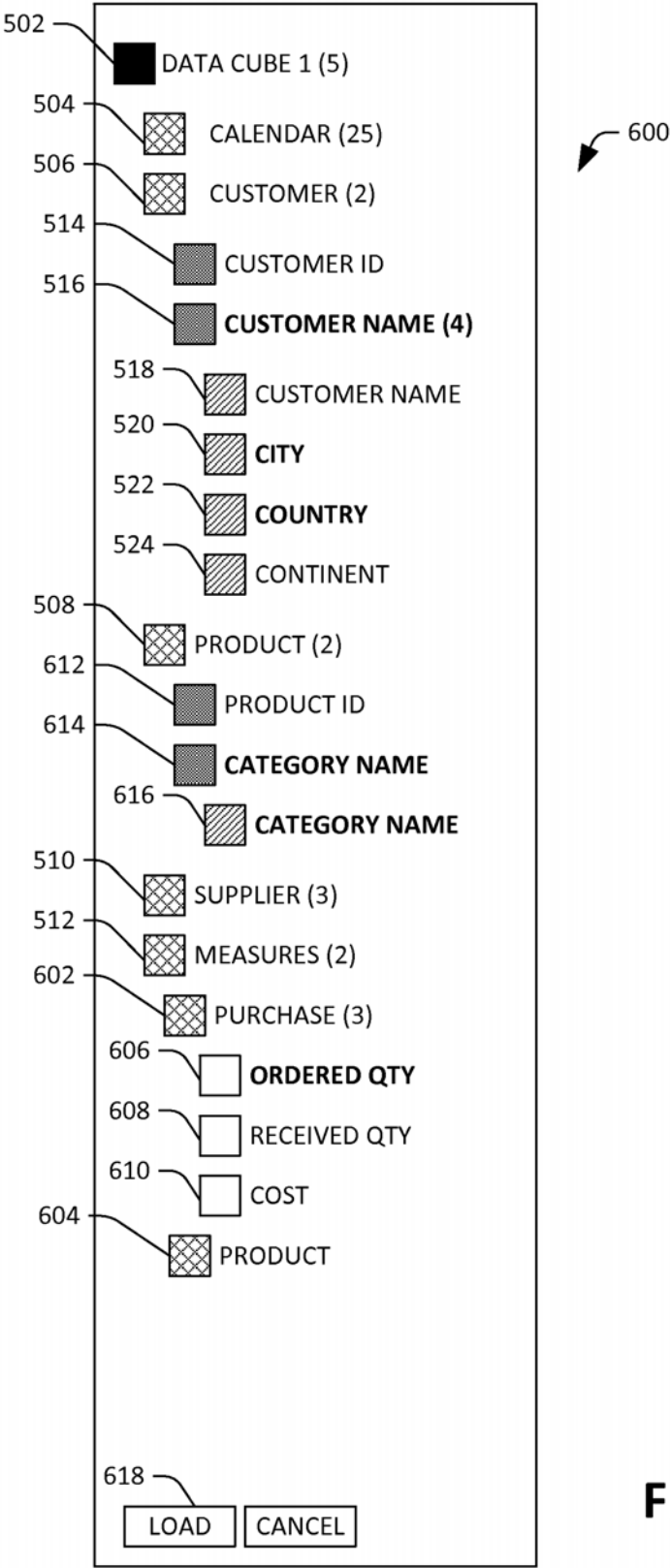


FIG. 6

700					
704					
706		708		710	
A		B	C	D	E
DIM1, ATT1		DIM1, ATT2	DIM2, ATT1	MEASURE	F
1	CHICAGO	USA	TENNIS	1000	
2	LONDON	UK	HANDSET	1200	
3	MADRID	SPAIN	APPLIANCES	1350	
4	MADRID	SPAIN	COMPUTER	200	
5	MARSEILLE	FRANCE	SWITCHES	450	
6	NEW YORK	USA	BIKES	975	
7	NEW YORK	USA	COMPUTER	1500	
8	NEW YORK	USA	CONSOLE	180	
9	PARIS	FRANCE	APPLIANCES	3000	
10	PARIS	FRANCE	COMPUTER	5500	
11	PARIS	FRANCE	CONSOLES	45	
12	PARIS	FRANCE	SWITCHES	1400	
13	QUEBEC	CANADA	HOME THEATER	1250	
14	QUEBEC	CANADA	HANDSET	700	
15	SEOUL	KOREA	APPLIANCES	2600	
16	TORONTO	CANADA	HANDSET	1450	
17	VANCOUVER	CANADA	GOLF	980	
18					

FIG. 7

702

	DIM1, ATT1	DIM1, ATT2	DIM2, ATT1	MEASURE1
1	CHICAGO	USA	TENNIS	1000
2	LONDON	UK	HANDSET	1200
3	MADRID	SPAIN	APPLIANCES	1350
4	MADRID	SPAIN	COMPUTER	200
5	MARSEILLE	FRANCE	SWITCHES	450
6	NEW YORK	USA	BIKES	975
7	NEW YORK	USA	COMPUTER	1500
8	NEW YORK	USA	CONSOLE	180
9	PARIS	FRANCE	APPLIANCES	3000
10	PARIS	FRANCE	COMPUTER	5500
11	PARIS	FRANCE	CONSOLES	45
12	PARIS	FRANCE	SWITCHES	1400
13	QUEBEC	CANADA	HOME THEATER	1250
14	QUEBEC	CANADA	HANDSET	700
15	SEOUL	KOREA	APPLIANCES	2600
16	TORONTO	CANADA	HANDSET	1450
17	VANCOUVER	CANADA	GOLF	980

800

804

802

SOURCE
EXPAND DIM1
EXPAND DIM2
ADD MEASURE1

FIG. 8

900

	DIM1, ATT1	DIM1, ATT2
1	CHICAGO	USA
2	LONDON	UK
3	MADRID	SPAIN
4	MADRID	SPAIN
5	MARSEILLE	FRANCE
6	NEW YORK	USA
7	NEW YORK	USA
8	NEW YORK	USA
9	PARIS	FRANCE
10	PARIS	FRANCE
11	PARIS	FRANCE
12	PARIS	FRANCE
13	QUEBEC	CANADA
14	QUEBEC	CANADA
15	SEOUL	KOREA
16	TORONTO	CANADA
17	VANCOUVER	CANADA

804

SOURCE
EXPAND DIM1
EXPAND DIM2
ADD MEASURE1

802

FIG. 9

	DIM1, ATT1	DIM1, ATT2	DIM2, ATT1
1	CHICAGO	USA	TENNIS
2	LONDON	UK	HANDSET
3	MADRID	SPAIN	APPLIANCES
4	MADRID	SPAIN	COMPUTER
5	MARSEILLE	FRANCE	SWITCHES
6	NEW YORK	USA	BIKES
7	NEW YORK	USA	COMPUTER
8	NEW YORK	USA	CONSOLE
9	PARIS	FRANCE	APPLIANCES
10	PARIS	FRANCE	COMPUTER
11	PARIS	FRANCE	CONSOLES
12	PARIS	FRANCE	SWITCHES
13	QUEBEC	CANADA	HOME THEATER
14	QUEBEC	CANADA	HANDSET
15	SEOUL	KOREA	APPLIANCES
16	TORONTO	CANADA	HANDSET
17	VANCOUVER	CANADA	GOLF

SOURCE
EXPAND DIM1
EXPAND DIM2
ADD MEASURE1

802

804

1000

FIG. 10

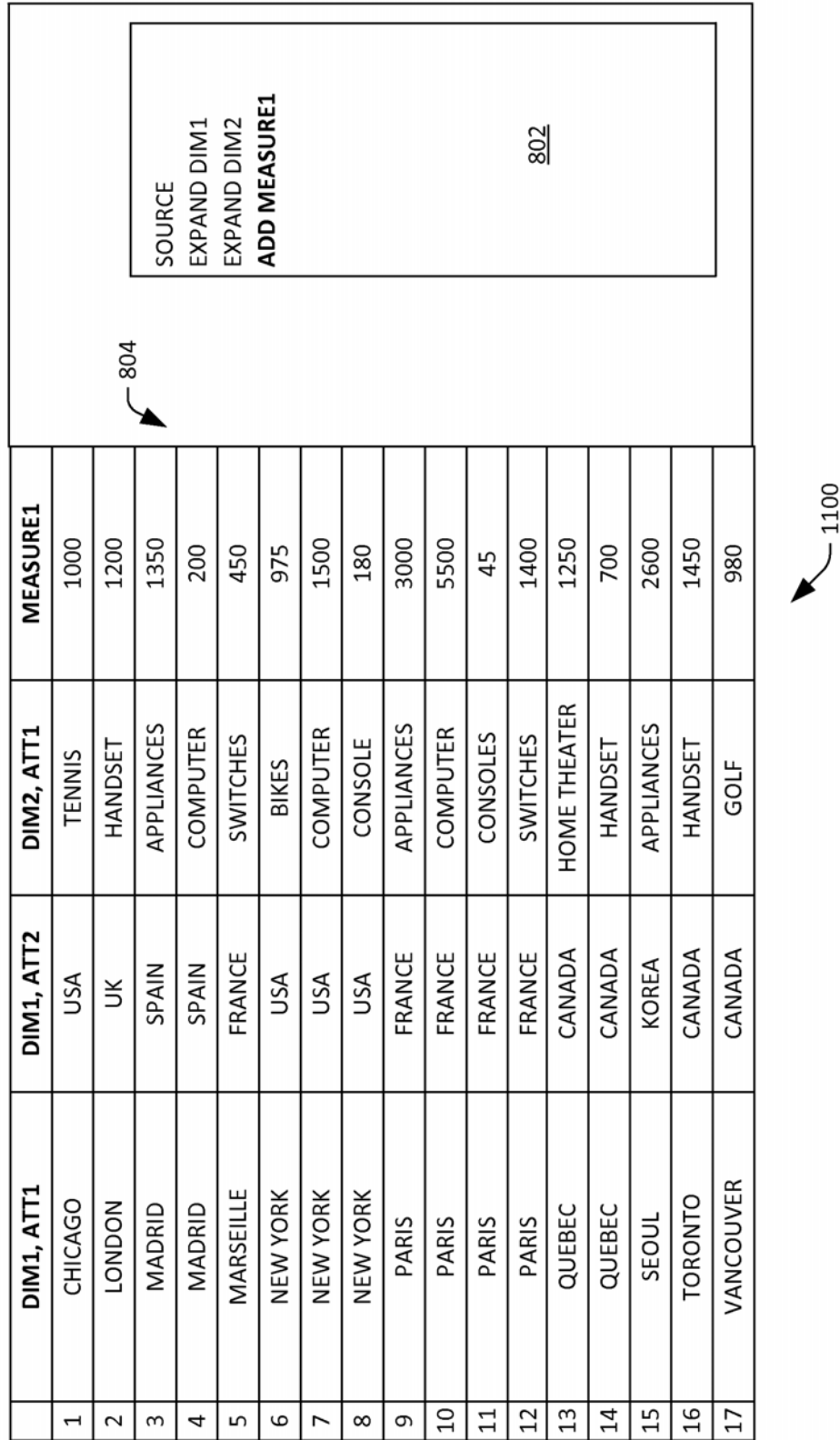


FIG. 11

```
LET
    SOURCE = [LOCATION]
    #"EXPAND CUSTOMER NAME" = EXPANDFUNCTION(CUBE ID, {[NAME = "CUSTOMER"]}[DATA][[NAME = "CUSTOMER NAME"]][DATA]...
    #"EXPAND CUSTOMER NAME" = EXPANDFUNCTION(CUBE ID, {[NAME = "PRODUCT"]}[DATA][[NAME = "CATEGORY"]][DATA]...
    #"ADD ORDERED QTY" = ADDMEASUREFUNCTION(EXPAND FUNCTION(DIMENSION NAME, MEASURE)...DATA)[[NAME = ...
IN
    #"ADD ORDERED QTY"
```

1200

FIG. 12

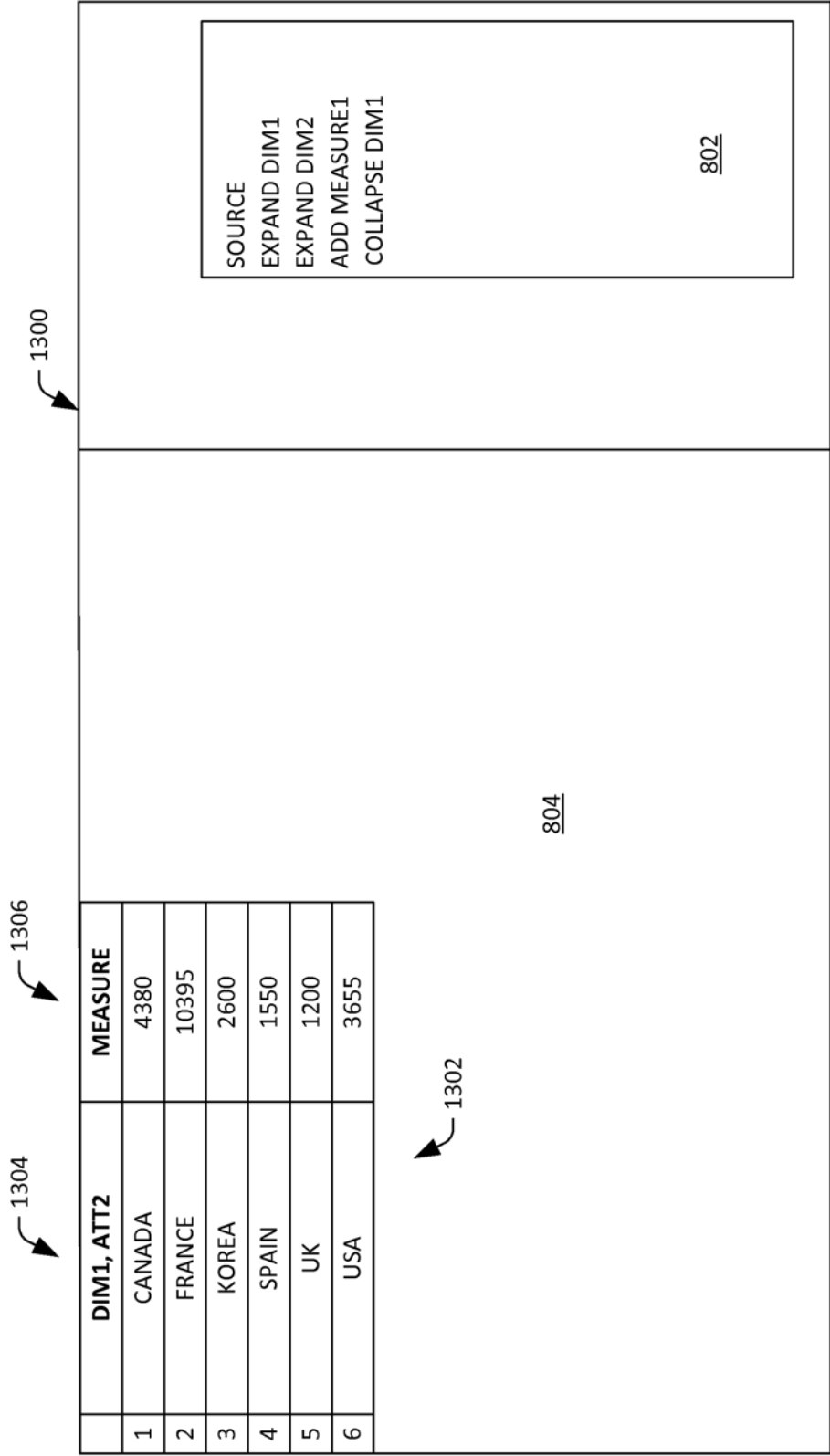


FIG. 13

	DIM1, ATT1	DIM1, ATT2	DIM2, ATT1	MEASURE1
1	CHICAGO	USA	TENNIS	1000
2	LONDON	UK	HANDSET	1200
3	MADRID	SPAIN	APPLIANCES	1350
4	MADRID	SPAIN	COMPUTER	200
5	MARSEILLE	FRANCE	SWITCHES	450
6	<u>1402</u> NEW YORK	LISA	BIKES	975
7	NEW YORK	EQUALS	OMPUTER	1500
8	NEW YORK	DOES NOT EQUAL	CONSOLE	180
9	PARIS	BEGINS WITH	APPLIANCES	3000
10	PARIS	ENDS WITH	OMPUTER	5500
11	PARIS	CONTAINS	CONSOLES	45
12	PARIS	DOES NOT CONTAIN	SWITCHES	1400
13	QUEBEC	FRANCE	HOME THEATER	1250
14	QUEBEC	CANADA	HANDSET	700
15	SEOUL	CANADA	APPLIANCES	2600
16	TORONTO	KOREA	HANDSET	1450
17	VANCOUVER	CANADA	GOLF	980

SOURCE
EXPAND DIM1
EXPAND DIM2
ADD MEASURE1
COLLAPSE DIM1

702

1400

FIG. 14

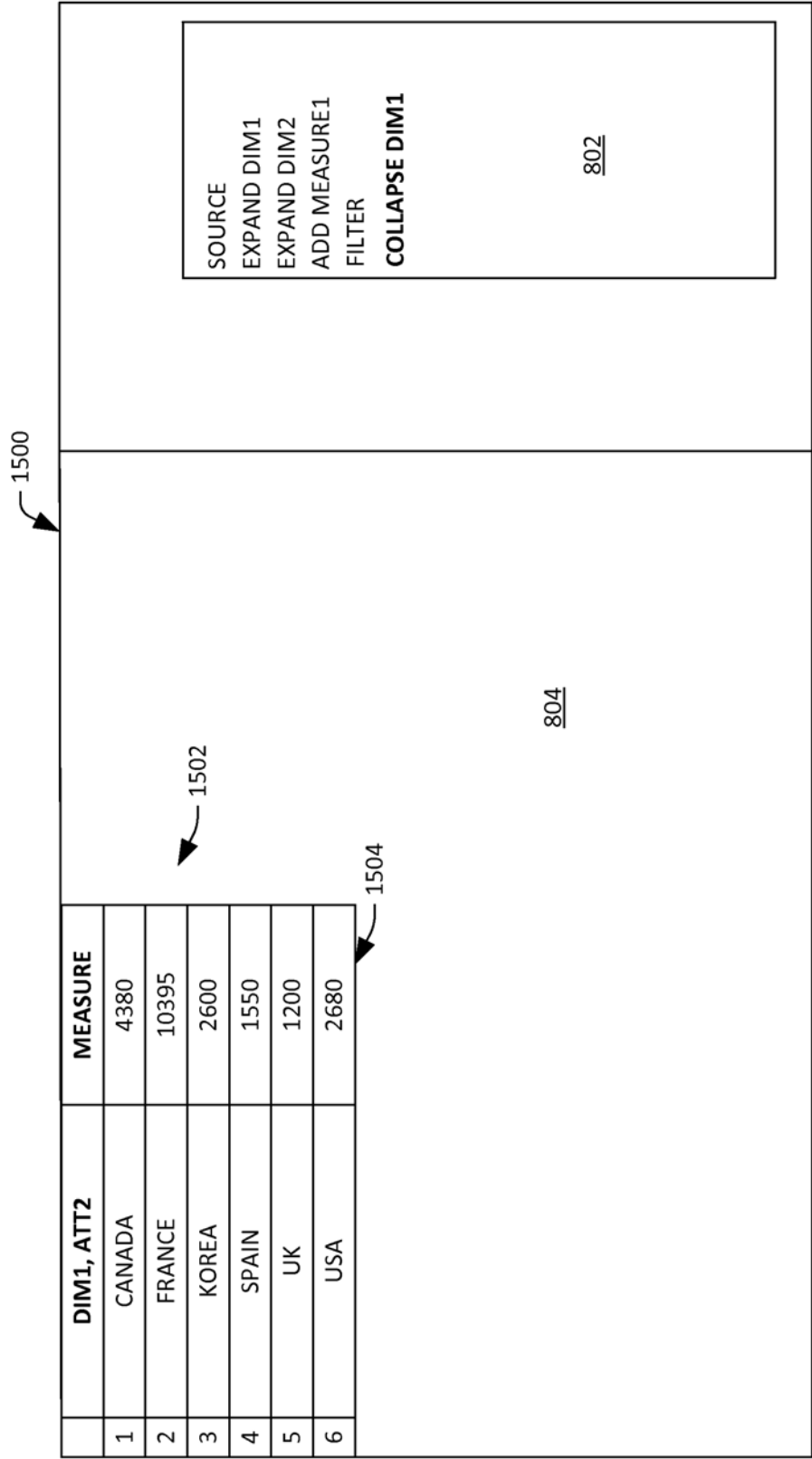


FIG. 15

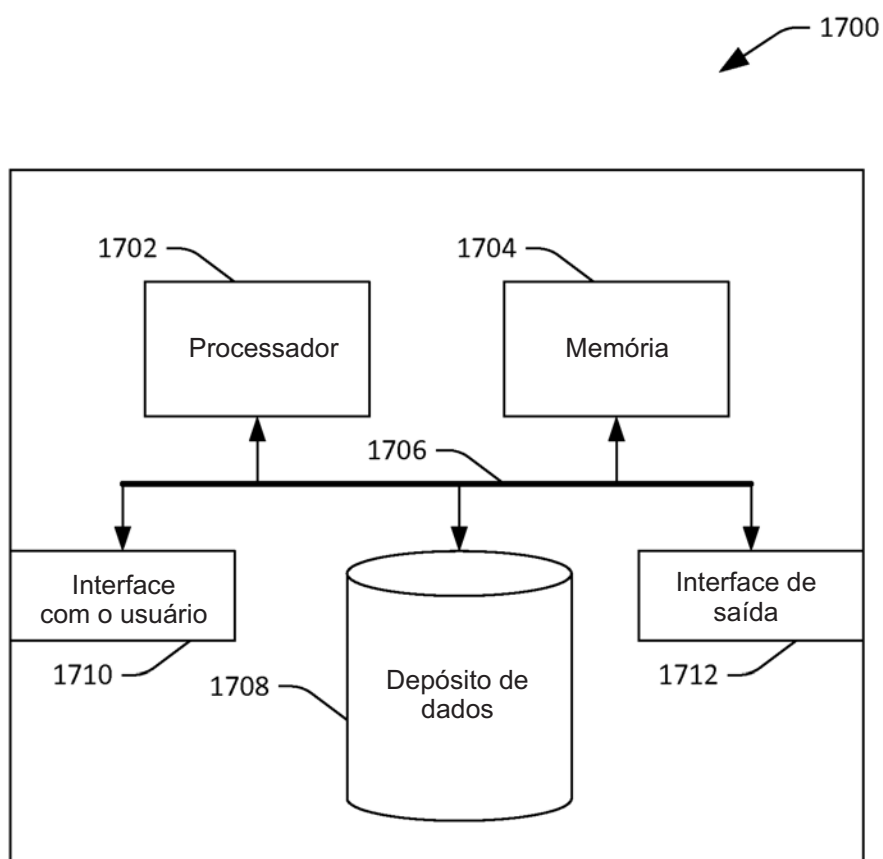
```

LET
    SOURCE = [LOCATION]
    #"EXPAND CUSTOMER NAME" = EXPANDFUNCTION(CUBE ID, {[NAME = "CUSTOMER"]}[DATA][{[NAME = "CUSTOMER NAME"]}[DATA]...
    #"EXPAND CUSTOMER NAME" = EXPANDFUNCTION(CUBE ID, {[NAME = "PRODUCT"]}[DATA][{[NAME = "CATEGORY"]}[DATA]...
    #"ADD ORDERED QTY" = ADDMEASUREFUNCTION(EXPAND FUNCTION(DIMENSION NAME, MEASURE) ..DATA){[NAME = ..
    FILTERED ROWS = SELECTROWFUNCTION("#"ADD ORDERED QTY", EACH [CUSTOMER NAME.CITY] <> "NEW YORK"),
    COLLAPSEDATTRIBUTES = COLLAPSEFUNCTION (FILTEREDROWS, {"CUSTOMER NAME.CITY", "CATEGORY NAME.CUSTOMER NAME...

IN
    COLLAPSEDATTRIBUTES

```

FIG. 16

**FIG. 17**