

VINÍCIUS VALÉRIO DE SIQUEIRA

DESENVOLVIMENTO DE UM DATA WAREHOUSE
PARA A CLÍNICA DO LEITE - Esalq/USP

Projeto de Formatura apresentado à
Disciplina PSI 2594 – Projeto de
Formatura II, da Escola Politécnica da
Universidade de São Paulo

São Paulo
2009

VINÍCIUS VALÉRIO DE SIQUEIRA

DESENVOLVIMENTO DE UM DATA WAREHOUSE
PARA A CLÍNICA DO LEITE - Esalq/USP

Projeto de Formatura apresentado à
Disciplina PSI 2594 – Projeto de
Formatura II, da Escola Politécnica da
Universidade de São Paulo

Área de Concentração:
Engenharia Elétrica com Ênfase em
Sistemas Eletrônicos

Orientador: Prof. Dr. Volnys Borges Bernal

São Paulo
2009

DEDICATÓRIA

Dedico este trabalho a minha
família e minha namorada.

RESUMO

A Clínica do Leite é uma instituição da Esalq/USP que realiza análise de leite para produtores e empresas, e tem como objetivo melhorar a qualidade do leite no estado de São Paulo. Ao longo de sua história, esta instituição coletou uma grande quantidade de dados, que são basicamente informações sobre análise de leite e dados zootécnicos. Consolidar, organizar e cruzar toda esta informação é uma necessidade para se conhecer profundamente o cenário da qualidade do leite no estado e mesmo no país, e poder vislumbrar pontos que possam ser melhorados neste cenário. O objetivo deste trabalho é desenvolver um Data Warehouse (DW) que será parte de uma solução de Business Intelligence (BI) para resolver esses cruzamentos em uma estrutura robusta e eficiente. Com esta ferramenta será possível responder de forma rápida e simples perguntas que no sistema antigo seriam extremamente difíceis e mesmo inviáveis ao usuário final que não possua conhecimento técnico. Com a maior facilidade de acesso aos dados históricos da Clínica do Leite espera-se incentivar seu estudo, e desta forma contribuir para aumentar o conhecimento do cenário leiteiro do estado e até do país. Consequentemente, pesquisadores poderão desenvolver seus estudos com maior eficiência e qualidade, contribuindo para o direcionamento de melhorias neste cenário.

Palavras-chave: Data Warehouse. Business Intelligence. Clínica do Leite. Cenário leiteiro.

ABSTRACT

The Clínica do Leite is an institution of Esalq/USP that makes milk analysis to producers and companies, and aims to improve the quality of milk in the state of São Paulo. Throughout its history, this institution has collected a large amount of data, which are basically information on milk analysis and livestock performance data. Consolidate, organize and cross all of this information is a necessity to deeply know the stage of milk quality in the state and even the country, and can identify parts that may be improved in this scenario. The objective of this work is to develop a Data Warehouse (DW) that will be part of a solution of Business Intelligence (BI) to resolve these crosses in a robust and efficient data structure. With this tool will be possible to respond quickly and easily questions that in the old system would be extremely difficult or even impractical to end user that has no technical knowledge. With easy access to historical data of the Clínica do Leite is expected to encourage it study, and thereby help to raise awareness of the actual scenario of the state and even the country. Consequently, researchers can develop their studies with greater efficiency and quality, contributing to improvements in this scenario.

Keywords: Data Warehouse. Business Intelligence. Clínica do Leite. Dairy Scenario.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de um processo de BI.....	16
Figura 2 – Processo ETL.....	19
Figura 3 – Star schema	21
Figura 4 – Exemplo de modelagem utilizando Star schema	21
Figura 5 – Snowflake schema	22
Figura 6 – Exemplo de relatório estático	24
Figura 7 – Exemplo de relatório Ad hoc	25
Figura 8 – Exemplo de Cubo de dados	26
Figura 9 – Fluxo da Metodologia	28
Figura 10 – Exemplo de estrutura relacional para banco de dados	30
Figura 11 – Exemplo de estrutura multidimensional.....	30
Figura 12 – Modelagem da tabela de Análise de Leite	34
Figura 13 – Modelagem da tabela de Coeficientes Zootécnicos.....	36
Figura 14 – Exemplo do fluxo de dados de uma transformação	37
Figura 15 – Modelagem da tabela de Coeficientes Zootécnicos por tipo de UA e mês	39
Figura 16 – Conteúdo do passo "Table input b_org_ext"	40
Figura 17 – Conteúdo do passo "Sort rows".....	41
Figura 18 – Conteúdo do passo "Row denormaliser"	42
Figura 19 – Conteúdo do passo "Null p End Padrao".....	43
Figura 20 – Conteúdo do passo "Select values".....	44
Figura 21 – Conteúdo do passo "Table output DIM_ACCOUNT".....	45
Figura 22 – Transformação da dimensão mês.....	46
Figura 23 – Transformação da dimensão de indicadores	46
Figura 24 – Transformação da dimensão de tipos de UA	47
Figura 25 – Job da tabela Fato de coeficientes zootécnicos.....	48
Figura 26 – Transformação Coeficientes Zootécnicos	49
Figura 27 – Transformação temperatura.....	50
Figura 28 – Conteúdo do passo "Update FACT_ANALYSIS_RESULT"	51
Figura 29 – Transformação número de coletas.....	52

Figura 30 – Transformação padrão internacional total	53
Figura 31 – Transformação padrão internacional	54
Figura 32 – Transformação número de análises DELVO e CHARM	55
Figura 33 – Transformação número de análises DELVO e CHARM positivas	56
Figura 34 – Transformação mostrar análise	57
Figura 35 – Exemplo de transformação incremental	57
Figura 36 – Exemplo de transformação manual	58
Figura 37 – Relatório de Coeficientes Zootécnicos	62

LISTA DE ABREVIATURAS E SIGLAS

AGL	Ácidos Graxos Livres
ATB	Resíduo de Antibiótico
BI	Business Intelligence
CBT	Contagem Bacteriana Total
CCS	Contagem de Células Somáticas
DW	Data Warehouse
ETL	Extract Transform Load
ID	Identificador
OLAP	Online Analytical Processing
OS	Ordem de Serviço
TI	Tecnologia da Informação
UA	Unidade de Análise

SUMÁRIO

1 INTRODUÇÃO	11
2 OBJETIVOS.....	13
3 JUSTIFICATIVA.....	14
4 REVISÃO DA LITERATURA	15
4.1 CONCEITO DE BUSINESS INTELLIGENCE.....	15
4.2 CONCEITO DE ETL	16
4.2.1 <i>Extração</i>	16
4.2.2 <i>Transformação</i>	17
4.2.3 <i>Carga</i>	18
4.3 CONCEITO DE DATA WAREHOUSE	19
4.4 CONCEITO DE RELATÓRIOS E CUBOS DE DADOS.....	23
4.4.1 <i>Relatórios Estáticos</i>	23
4.4.2 <i>Relatórios Ad hoc</i>	24
4.4.3 <i>Cubos de dados (OLAP)</i>	25
5 MATERIAIS E MÉTODOS	27
5.1 LEVANTAMENTO DAS NECESSIDADES	29
5.1.1 <i>Gestão da Informação</i>	29
5.1.2 <i>Escopo do projeto</i>	31
5.2 ANÁLISE DOS DADOS	34
5.2.1 <i>Análise de Leite</i>	34
5.2.2 <i>Coeficientes Zootécnicos</i>	36
5.3 SELEÇÃO DOS MÉTODOS	36
5.4 CONSTRUÇÃO	37
5.4.1 <i>Fato e dimensões</i>	38
5.4.2 <i>Dimensão de clientes</i>	39
5.4.3 <i>Dimensão mês</i>	45
5.4.4 <i>Dimensão de indicadores</i>	46
5.4.5 <i>Dimensão de tipos de UA</i>	47

5.4.6 Tabela fato de coeficientes zootécnicos.....	47
5.4.6.1 TRANSFORMAÇÃO COEFICIENTES ZOOTÉCNICOS	48
5.4.6.2 TRANSFORMAÇÃO TEMPERATURA	50
5.4.6.3 TRANSFORMAÇÃO NÚMERO DE COLETAS	51
5.4.6.4 TRANSFORMAÇÃO PADRÃO INTERNACIONAL TOTAL	52
5.4.6.5 TRANSFORMAÇÃO PADRÃO INTERNACIONAL	54
5.4.6.6 TRANSFORMAÇÃO NÚMERO DE ANÁLISES DELVO E CHARM	55
5.4.6.7 TRANSFORMAÇÃO NÚMERO DE ANÁLISES DELVO E CHARM POSITIVAS.....	55
5.4.6.8 TRANSFORMAÇÃO MOSTRAR ANÁLISE	56
5.4.7 Carga incremental.....	57
5.4.8 Carga manual.....	58
5.5 TESTES	59
5.6 IMPLEMENTAÇÃO	59
6 RESULTADOS.....	61
7 DISCUSSÃO	63
8 CONCLUSÕES	64
9 REFERÊNCIAS.....	66
10 APÊNDICE.....	68

1 INTRODUÇÃO

A Clínica do Leite é um setor do Departamento de zootecnia da Esalq/USP, que tem como objetivos principais avaliar a qualidade do leite no estado de São Paulo, e mesmo no Brasil, e aumentar a eficiência econômica da cadeia do leite através da identificação de áreas de ineficiência zootécnica.

Uma grande quantidade de dados vem sendo coletada pela Clínica do Leite ao longo de sua história, desde a década de 90, são basicamente informações sobre análise de leite e dados zootécnicos. Consolidar, organizar e cruzar toda esta informação é uma necessidade para se conhecer profundamente o cenário da qualidade do leite no estado e mesmo no país, e poder vislumbrar pontos que possam ser melhorados neste cenário.

Por exemplo, um pesquisador pode estar desenvolvendo um estudo sobre uma determinada doença e estar interessado em saber em quais mesorregiões do estado existe maior incidência dela. O banco de dados atual da Clínica do Leite tem esta informação, porém para chegar até ela será necessário um considerável esforço e também conhecimento técnico. O que torna a resolução do problema mais lenta, até mesmo desestimulando a utilização de toda a informação armazenada durante anos pela Clínica do Leite. No entanto, se existe uma ferramenta de fácil utilização, que responde rapidamente e com pouco esforço este tipo de questão, o acesso a informação fica muito mais simples e, conseqüentemente, o pesquisador pode concentrar seu esforço no que de fato é importante, sua pesquisa.

Atualmente, algumas informações referentes a fazendas e indústrias de laticínios são obtidas por *queries* (consultas diretas no banco de dados, que exigem conhecimento técnico) e tratadas em planilhas eletrônicas ou ferramentas estatísticas. Essas informações nada mais são do que cruzamentos de dados entre diversas tabelas do banco de dados. No caso do pesquisador anteriormente relatado, seria necessário cruzar a tabela de análises de leite, que contém os resultados positivos de existência da doença em amostras, com a tabela de produtores, que contém os dados referentes a localização do animal com a doença, por exemplo.

O foco desse projeto é desenvolver um Data Warehouse (DW) que será parte de uma solução de Business Intelligence (BI) para resolver esses cruzamentos em uma estrutura robusta e eficiente, evitando assim esse processo "manual" de organização de dados.

É incumbência desse projeto a filtragem e tratamento da informação do banco de dados original da Clínica do Leite e outras fontes de dados em um novo banco de dados (DW) disponibilizado de maneira personalizada, com uma nova estruturação, e que facilite sua manipulação tornando-a mais fácil e intuitiva, tanto para os configuradores do sistema como para o usuário final do sistema (mestrados, doutorandos, professores, pesquisadores, etc.). Após a transformação dessa estrutura podemos manipular mais facilmente esses dados para a confecção de novos relatórios, ou mesmo possibilitar aos usuários a confecção de seus próprios relatórios, os chamados ad hoc reports (relatórios customizados).

2 OBJETIVOS

Este trabalho tem como objetivo utilizar técnicas de tecnologia da informação (TI) para facilitar o acesso a informação contida no banco de dados da Clínica do Leite.

Será desenvolvido um data warehouse (DW), ou seja, um banco de dados informacional, projetado de forma a facilitar a obtenção da informação, ao contrario do banco de dados atual, que é desenhado de forma a facilitar a entrada da informação no banco, mas que torna a análise destes dados difícil e pouco eficiente.

Este DW poderá ser utilizado para a geração de relatórios e cubos de dados, o que tornará ainda mais fácil o acesso aos dados pelo usuário final.

3 JUSTIFICATIVA

A Clínica do Leite tem um grande banco de dados com informações sobre análise de leite e dados zootécnicos. Analisar de forma rápida e eficiente toda esta informação é uma necessidade para se conhecer melhor o cenário da qualidade do leite no país, e poder vislumbrar pontos que possam ser melhorados neste cenário.

Da forma como o banco de dados esta hoje, um pesquisador terá um considerável esforço e também necessitará de conhecimento técnico para obter a informação desejada.

Com o desenvolvimento de uma ferramenta de fácil utilização, que responde rapidamente e com pouco esforço as diversas questões que interessam a este pesquisador, haverá mais tempo para o que de fato é importante, a pesquisa zootécnica.

E, assim, com a maior facilidade de acesso a informação do banco de dados da Clínica do Leite, espera-se incentivar seu estudo, e desta forma contribuir para aumentar o conhecimento do cenário leiteiro do país.

Consequentemente, pesquisadores poderão desenvolver seus estudos com maior eficiência e qualidade, contribuindo para o direcionamento de melhorias neste cenário.

4 REVISÃO DA LITERATURA

4.1 Conceito de Business Intelligence

Business Intelligence (BI) é uma arquitetura composta por um conjunto de metodologias de gestão implementadas através de ferramentas de software, cuja função é proporcionar ganhos nos processos decisórios e analíticos, baseando-se na capacidade informacional das ferramentas que integram em um só lugar todas as informações necessárias a tomada de decisão.

O principal desafio de um projeto de BI é reunir e organizar todos os dados relevantes que possibilitem ao usuário final acesso fácil e eficiente a informação, e desta forma, aumentar a eficiência de uma organização.

Basicamente, o sistema deve retirar informações do banco de dados operacional de uma organização, filtrar e limpar a informação relevante, e armazenar novamente esses dados em um banco de dados informacional chamado de data warehouse (DW), que tem uma estrutura adequada a análise destes dados.

O processo que realiza a extração dos dados do banco operacional, a transformação destes dados (filtragem, limpeza, etc) e carga no novo banco informacional é chamado de ETL (Extract Transform Load).

Após a criação do DW, a informação já estará no formato ideal para ser utilizada e poderá ser acessada, dentre outras, por ferramentas de geração de relatórios e por ferramentas conhecidas como Online Analytical Processing (OLAP) ou Processo Analítico em Tempo Real, que geram cubos lógicos de dados para análise multidimensional.

A figura 1 apresenta um diagrama explicativo que descreve um processo de BI.

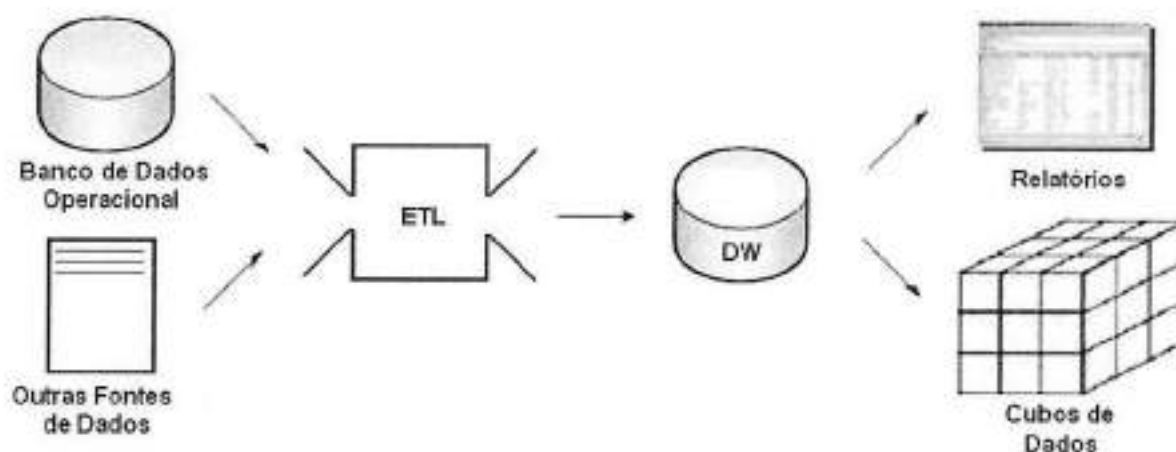


Figura 1 – Fluxo de um processo de BI

4.2 Conceito de ETL

A sigla ETL (Extract Transform Load) significa extração, transformação e carga. É um processo ou metodologia que tem como objetivo extrair dados relevantes de vários sistemas, transformar, limpar e padronizar estes dados e carregá-los em um novo sistema, onde estarão em um formato adequado (KIMBALL et. al.,2004).

4.2.1 Extração

A primeira etapa do processo ETL é a extração de dados dos sistemas operacionais da organização, o que pode envolver sistemas com diferentes formatos de dados. Um dos formatos mais comuns é o de bancos de dados relacionais, onde o relacionamento entre as diversas tabelas ocorre a partir de características em comum, por exemplo, a tabela de análise de leite esta ligada a tabela de produtores através do número de identificação do produtor comum a ambas.

Esta extração deve ser realizada de forma seletiva, apenas a informação relevante deve ser obtida. Em um banco transacional podemos ter tabelas com muitos campos que não são necessariamente importantes para a análise dos dados,

por exemplo, campos com informações operacionais como quem criou o registro ou a data quando isso ocorreu.

Outro exemplo de problema que pode ocorrer são tabelas de bancos operacionais com campos que permitem dados inconsistentes, por exemplo, o campo que contém um determinado resultado deveria ter apenas dados numéricos e por algum motivo um dos registros contém uma letra. Neste caso é necessário decidir se o registro deve ser descartado ou se deve ser transformado de alguma forma na próxima etapa do processo.

A redundância da informação, comum nos sistemas transacionais, também deve ser tratada, consolidando os dados para não haver inconsistências no banco informacional.

Existem ainda problemas que são específicos de cada sistema e precisam ser analisados individualmente. Mas o importante é que os dados que passam por essa etapa devem ser apenas aqueles estritamente necessários às análises que se deseja fazer, não apenas pela eficiência do sistema, mas também para que não se utilize tempo de projeto "limpando" dados desnecessários na próxima etapa do processo, a transformação.

4.2.2 Transformação

A etapa de transformação irá aplicar uma série de regras aos dados extraídos na etapa anterior. Algumas fontes de dados vão requerer pouca ou nenhuma manipulação da informação. Em outros casos, um ou mais tipos de transformações podem ser necessários para se atingir os requisitos apontados pelo usuário final, como por exemplo:

- Traduzir valores codificados. Por exemplo, o banco operacional armazena em determinado campo "P" para vaca preta, e no DW é desejada a

palavra "Prenha". Também pode acontecer o contrário, quando é necessário codificar um campo de alguma forma;

- Campos calculados. Por exemplo, ao invés de vários campos com resultados de análises realizadas para determinado produtor, é necessário apenas um campo com o valor médio das análises e outro com o número de análises;
- Filtragem. Por exemplo, existe uma tabela na fonte de dados com os resultados das análises de um determinado tipo de doença. Se no DW são desejados apenas registros cujas análises apontem a existência da doença, é necessário filtrar especificamente registros com esta característica;
- Agregação. Pode ser necessário consolidar os dados por mês em uma determinada tabela do DW. Por exemplo, para armazenar a média dos resultados das análises no mês de cada animal ao invés do valor das análises separadamente;
- Normalização ou desnormalização. Por exemplo, uma tabela da fonte de dados pode ter em cada coluna algum resultado referente a um tipo de análise diferente e no DW é desejado que estas análises estejam normalizadas, ou seja, em uma única coluna os nomes ou códigos destas análises e em outra coluna os resultados referentes a cada análise.

Estes são apenas alguns exemplos de tipos de tratamentos que são normalmente necessários. Mas cada sistema exigirá transformações diferentes dependendo dos dados de origem e dos requisitos apontados pelo usuário final.

4.2.3 Carga

A última etapa do processo ETL é a carga dos dados devidamente tratados no DW. Este processo pode variar bastante dependendo dos requerimentos da organização. Algumas tabelas do DW podem ser apagadas e ter todos os registros recarregados sempre que for atualizada (carga completa), enquanto outras podem

ter apenas registros novos incluídos periodicamente sem afetar os registros já existentes (carga incremental). Qual processo usar também depende do tempo que se dispõem, pois em tabelas com uma quantidade muito grande de dados, o tempo para fazer uma carga completa pode ser muito grande, o que pode tornar inviável executar o processo diariamente, por exemplo.

A figura 2 apresenta um diagrama explicativo que descreve o processo ETL.

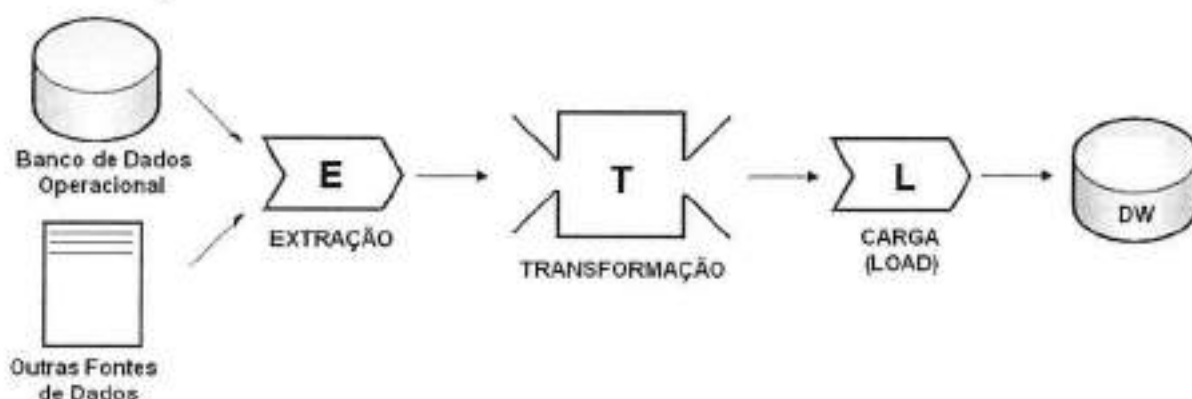


Figura 2 – Processo ETL

4.3 Conceito de Data Warehouse

É um repositório de dados (banco de dados) especificamente desenhado para facilitar a geração de relatórios e a análise da informação. A informação em um DW deve estar em um formato consistente, livre de dados incoerentes e estruturado de forma a cumprir os requerimentos analíticos do usuário final (KIMBALL et. al.,2004).

Ao contrario da maioria dos sistemas de bancos de dados transacionais, que são projetados de forma a tornar o mais eficiente possível a entrada de dados no banco, o DW tem como foco principal a saída destes dados. O DW é projetado de forma a aumentar a eficiência e facilitar o acesso aos dados pelo usuário final.

Outra característica do DW é que, por ser construído visando permitir análises, este banco mantém os dados ao longo do tempo, ou seja, armazena os dados históricos, podendo chegar a grandes quantidades de dados. Com estes dados é

possível ao usuário final fazer pesquisas e gerar relatórios que sejam úteis a tomada de decisão.

O DW tem também como característica uma grande quantidade de dados pré-calculados. Dessa forma é possível aumentar a eficiência na geração de relatórios, por exemplo. Pois não será necessário manipular os dados no momento da geração do relatório. A informação já estará previamente calculada. Isso é possível, pois no momento da construção do banco de dados já se sabia quais possíveis tipos de análises poderiam ser realizados pelo usuário final.

Existem dois modelos principais para a construção do DW, o Star schema e o Snowflake schema. No Star schema (figura 3), o mais utilizado, os dados são modelados em tabelas multidimensionais ligadas a uma única tabela de fatos, o que facilita o cruzamento de informação. As tabelas dimensionais contêm as características de um evento, enquanto a tabela fato armazena os fatos ocorridos e as chaves para as características correspondentes nas tabelas dimensionais (figura 4). Então, cada uma destas características está em uma tabela dimensão, que armazena a descrição desta característica. Por exemplo, na dimensão Produtor, existe além da chave que a liga a tabela fato, o nome do produtor, seu endereço, a sua cidade, etc. Na tabela fato, cada resultado está ligado a suas características, por exemplo, na tabela com resultados de análise de leite, o valor numérico do resultado da análise pode estar ligado a um determinado produtor, uma determinada data e um determinado tipo de análise. A tabela fato conecta-se as demais dimensões por múltiplas junções e as tabelas de dimensões conectam-se com apenas uma junção à tabela fato. Desta forma a consulta ocorre inicialmente nas tabelas de dimensão e depois nas tabelas de fatos, assegurando a precisão dos dados por meio de uma estrutura de chaves onde não é preciso percorrer todas as tabelas, garantindo um acesso mais eficiente e com melhor desempenho.



Figura 3 – Star schema

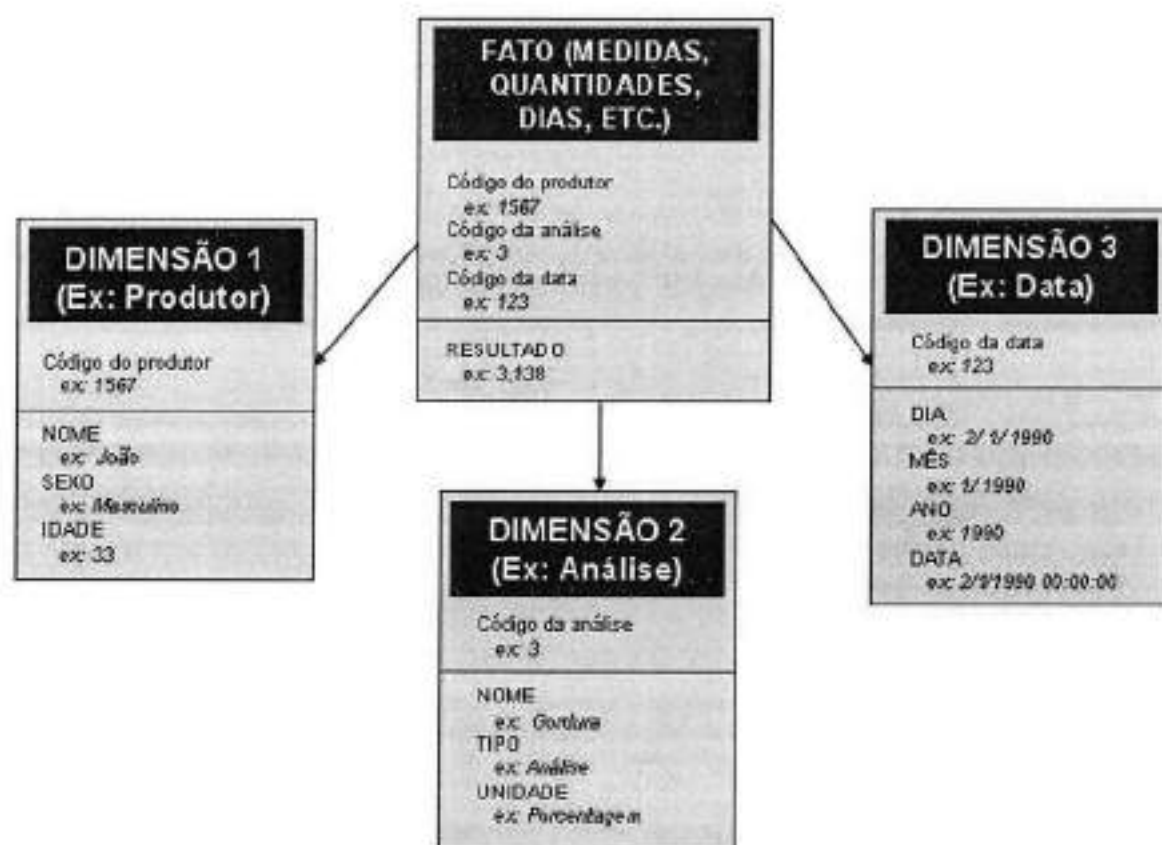


Figura 4 – Exemplo de modelagem utilizando Star schema

O Snowflake schema é uma variação do star schema, onde existe mais de dois níveis de encadeamento de tabelas (figura 5). Utilizando este modelo o tamanho da tabela dimensão principal é reduzido e se evita redundância, porém perde-se desempenho devido ao aumento do número de buscas entre tabelas.

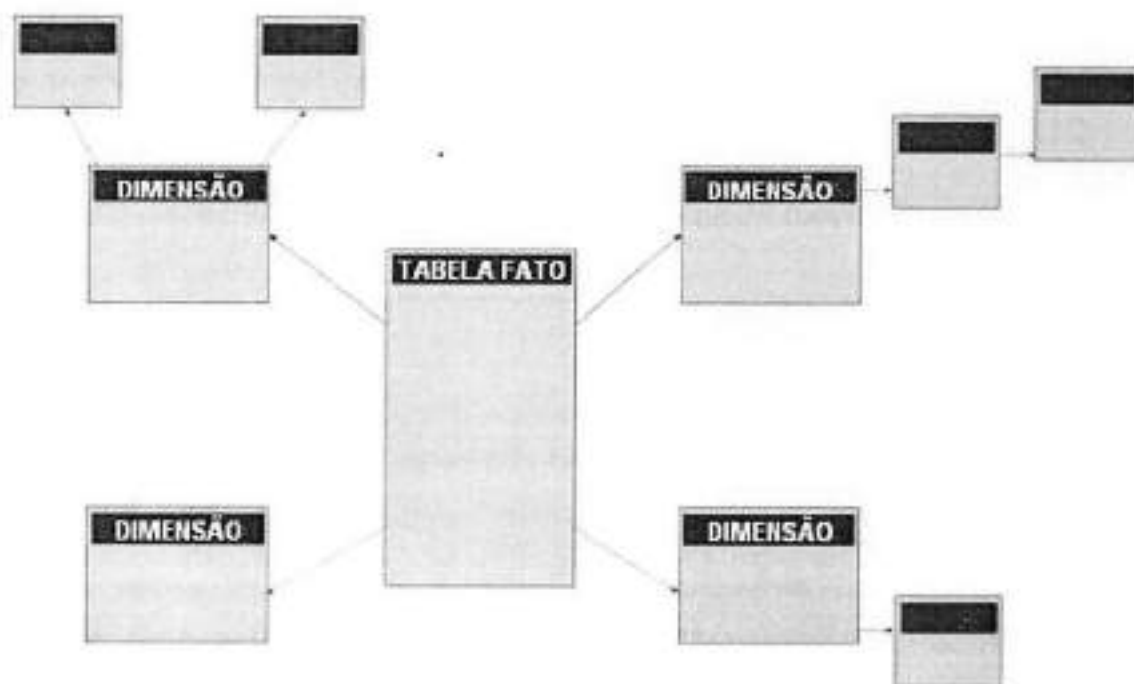


Figura 5 – Snowflake schema

Uma tabela fato, em geral, tem poucas colunas, apenas algumas colunas com resultados de medidas e outras com as chaves que a ligam as dimensões que caracterizam a medida. No entanto, pode ter uma quantidade enorme de linhas (registros), da ordem de milhões. Um dos fatores que levam a isso é que dependendo dos requisitos necessários, a granularidade dos dados precisa ser a menor possível. Por exemplo, em uma determinada tabela os dados podem estar armazenados por dia e em outra consolidados (média, valor máximo, valor mínimo, etc.) por mês, obviamente a tabela com os dados por dia será muito maior. Outro fator que torna as tabelas fato grandes é a necessidade de normalização. Nos bancos operacionais existem muitas tabelas que tem uma ou mais características desnormalizadas. Por exemplo, com colunas com nomes de cores, onde cada registro pode estar associado a uma ou mais cores. Na tabela fato a informação de cor vai se tornar uma dimensão, então será necessário mais de um registro na

tabela fato para cada registro da tabela desnormalizada que estiver associado a mais de uma cor. Dessa forma existirá vários registros idênticos que se diferenciam apenas pela dimensão cor. O que obviamente aumenta o numero de registros na tabela fato em relação a tabela fonte dos dados.

Outro conceito relevante é o de Data Mart, que é idêntico a um Data warehouse, porém voltado apenas a uma determinada área da organização, ao contrario do DW, que engloba assuntos de toda a organização.

4.4 Conceito de Relatórios e Cubos de dados

Após a construção do DW, a informação já esta disponível para ser utilizada. No entanto, ainda seria necessário realizar *queries* (buscas) diretamente no banco de dados para se obter a informação desejada, o que continua exigindo conhecimento técnico.

Para se resolver este problema são criados relatórios e cubos de dados com objetivo de facilitar o acesso a informação pelo usuário final e também utilizar da melhor forma possível o potencial informacional do DW.

Estas ferramentas podem ser divididas em relatórios estáticos, relatórios Ad hoc e cubos de dados (OLAP).

4.4.1 Relatórios Estáticos

São relatórios pré-prontos, com algumas regras já definidas. Dessa forma, não é possível alterar o formato dos relatórios, apenas os filtros de dados. Um exemplo deste tipo de relatório esta apresentado na figura 6, onde os campos (Mês, Média Aritmética, Desvio, etc.) são estáticos e só é possível alterar os filtros do relatório variando, por exemplo, a empresa ou o período que se deseja considerar.



ANÁLISE DESCRITIVA MENSAL

R100

Análise: GOR-Gordura- (% m/m)
 Período: 01/08/2007 a 20/08/2008
 Empresa: _____
 Tipo de UA: Tanque

Data de Emissão: 22/10/2009

Mês	Média Aritm.	Desvio Padrão	Média Geom.	Nº Observações	Máximo	Mínimo	1º Quartil	Mediana	2º Quartil
Ago/07	3.60	0.48	3.57	1165	9.47	1.72	3.82	3.62	3.37
Set/07	3.57	0.45	3.54	1236	10.81	1.43	3.82	3.59	3.35
Out/07	3.55	0.44	3.52	1393	8.54	1.26	3.77	3.58	3.31
Nov/07	3.54	0.38	3.52	1189	5.36	1.57	3.78	3.57	3.31
Dez/07	3.46	0.34	3.44	1196	4.83	2.04	3.67	3.48	3.27
Jan/08	3.54	0.35	3.52	1733	4.88	2.03	3.75	3.54	3.35
Total	3.49	0.49	3.48	7692	17.00	1.40	3.80	3.50	3.38

Figura 6 – Exemplo de relatório estático

4.4.2 Relatórios Ad hoc

São relatórios customizáveis, onde não apenas é possível mudar os filtros de dados, mas também os campos que serão apresentados ou a forma como as medidas são agrupadas (contagem, soma, média, etc.), por exemplo. Tomando como referência o exemplo apresentado na figura 7, e considerando este relatório um Ad hoc, seria possível, por exemplo, substituir o produto por produtor (para gerar um relatório de número de análises realizadas por cada produtor por ano, por exemplo), ou o parâmetro Ano substituído por trimestre, ou ainda o resultado ao invés de apresentar o número de análises de determinado produto por ano, apresentar a média dessas análises no ano por produtor, por exemplo. Ou seja, com um relatório Ad hoc o usuário final pode fazer qualquer tipo de análise que desejar, contanto que o determinado tipo de análise tenha sido previsto no momento da modelagem do DW.

Produto	Ano				
	2002	2003	2004	2005	2006
CBT	2200	4300	5400	5800	200
ESD	3200	4700	5600	6400	100
CSV	1100	2300	4200	3500	700
ATB	300	3400	4300	4500	900

Figura 7 – Exemplo de relatório Ad hoc

4.4.3 Cubos de dados (OLAP)

O Cubo de dado (OLAP, sigla de Online analytical processing) é uma forma rápida de análise em modelos multidimensionais. É um tipo de relatório Ad hoc, onde existe liberdade para se realizar qualquer tipo de análise envolvendo cruzamento de dados a partir de um banco analítico, porém de forma mais dinâmica e interativa, com funcionalidades como agrupamentos em tempo real, por exemplo. Um exemplo de Cubo de dados está apresentado na figura 8, onde é possível fazer os mesmos tipos de análises realizadas nos Ad hocs, mas ainda é possível, por exemplo, selecionar um dos resultados consolidados (por Produtor x Produto x Período) e observar em detalhes os dados que levaram aquela consolidação (drill-through). Também é possível, por exemplo, selecionar o parâmetro Ano e realizar um drill-down em tempo real, fazendo com que o detalhamento do dado seja expandido e o dado seja consolidado por semestre, por exemplo.

			Ano				
Tipo de Cliente	Cliente	Produto	2002	2003	2004	2005	2006
Premium	Abigail V. Vargas - Faz. M.Sra. das Valias - Coopervass	CBT	2200	4300	5400	5600	200
		ESD	3200	4700	5600	6400	100
		CSV	1100	2300	4200	3500	700
		ATB	300	3400	4300	4500	900
Abigail V. Vargas - Faz. M.Sra. das Valias - Coopervass			400	4300	40400	20500	900
Cód. da OS	Cód. Cliente	Nome do cliente	No. de amostra	Data de Coleta	No. de colheita	Data do re	
OS-031207	CL-000123	Abigail V. Vargas - Fa	0	28/2/2009 00:00	10	2/3	6000 300
OS-031206	CL-000123	Abigail V. Vargas - Fa	0	28/2/2009 00:00	10	2/3	6000 300
OS-031205	CL-000511	Abigail V. Vargas - Fa	139	25/2/2009 00:00	3	2/3	
OS-031204	CL-000173	Abigail V. Vargas - Fa	2	2/3/2008 00:00	1	2/3	2100 800
OS-031203	CL-000173	Abigail V. Vargas - Fa	1	2/3/2008 00:00	1	2/3	1000 500
OS-031202	CL-000819	Abigail V. Vargas - Fa	60	23/2/2009 00:00	2	2/3	
OS-031201	CL-000048	Abigail V. Vargas - Fa	23	28/2/2009 00:00	1	2/3	5100 300
OS-031200	CL-001867	Abigail V. Vargas - Fa	45	24/2/2009 00:00	1	2/3	
OS-031199	CL-000344	Abigail V. Vargas - Fa	110	27/2/2009 00:00	3	2/3	
OS-031198	CL-001999	Abigail V. Vargas - Fa	29	20/2/2009 00:00	1	2/3	
OS-031197	CL-000779	Abigail V. Vargas - Fa	397	27/2/2009 00:00	8	2/3	
OS-031196	CL-000452	Abigail V. Vargas - Fa	20	20/2/2009 00:00	1	2/3	
OS-031195	CL-000079	Abigail V. Vargas - Fa	122	24/2/2009 00:00	3	2/3	
OS-031194	CL-000079	Abigail V. Vargas - Fa	120	24/2/2009 00:00	3	2/3	

Figura 8 – Exemplo de Cubo de dados

5 MATERIAIS E MÉTODOS

Neste trabalho foi utilizada a seguinte metodologia:

1. Levantamento – Nesta fase foram definidas as necessidades da Clínica do Leite e os resultados que deveriam ser obtidos. As informações coletadas neste levantamento serviram como base para o fechamento do escopo do projeto e elaboração do cronograma detalhado de implementação.

2. Análise de Dados – Com base no escopo fechado na fase anterior, uma análise detalhada das bases de dados, de onde os dados foram extraídos, foi realizada. Este procedimento auxiliou na seleção dos métodos que foram utilizados na extração e/ou exploração dos dados e na modelagem e execução do processo. Como resultado desta fase foi criada a listagem dos registros a serem processados. Este documento contém, também, a estrutura destes registros. A partir da lista de registros a serem processados, uma identificação de dados incorretos e/ou redundantes foi realizada. Esta atividade gerou a listagem dos registros redundantes/incorretos que foi utilizada para realizar a limpeza dos dados, possibilitando uma utilização eficiente das informações.

3. Seleção dos Métodos – Utilizando as informações coletadas nas etapas anteriores foi possível identificar e escolher o método mais adequado para realizar a extração/exploração dos dados e para exibir as informações processadas pelo BI. Além disso, foi, também, nesta fase, que as ferramentas, equipamentos e instalações foram providenciados.

4. Construção – Foi neste momento que a solução de BI definida no escopo do projeto foi desenvolvida. Ao final desta etapa o aplicativo estava pronto para ser utilizado.

5. Testes – Para garantir que os resultados obtidos estejam de acordo com os esperados é de extrema importância que o usuário final, guiado por scripts de teste, realize uma série de testes/validações para avaliar a confiabilidade dos

dados. Durante a avaliação dos resultados, os problemas encontrados são corrigidos e re-testados e, ao final desta etapa, a ferramenta de BI é aprovada pelos usuários finais.

6. **Implementação** – Após a aprovação do aplicativo pelos usuários finais, os últimos detalhes do projeto devem ser definidos: periodicidade de execução do processo, responsáveis pela execução do processo, treinamento de usuários chaves, etc.

Abaixo, segue um fluxo simplificado da metodologia do projeto:



Figura 9 – Fluxo da Metodologia

5.1 Levantamento das necessidades

5.1.1 Gestão da Informação

Com base nos conceitos de tecnologia da informação (TI) relatados anteriormente, este projeto visa melhorar a gestão da informação na Clínica do Leite, para possibilitar aos seus administradores e pesquisadores conhecer melhor o cenário leiteiro do estado e mesmo do país, e dessa forma, auxiliá-los a direcionar possíveis melhorias neste cenário.

Conforme apresentado anteriormente, a forma como este projeto visa melhorar a gestão da informação na Clínica do Leite é através da implantação de uma solução de Business Intelligence que tem como ponto de partida a criação de um banco de dados informacional, o Data Warehouse (DW, foco principal deste trabalho), que é especificamente projetado para favorecer acesso a informação de forma fácil e rápida. E a partir deste novo banco de dados ferramentas poderão ser utilizadas para possibilitar a geração de relatórios e cubos de dados que utilizem ao máximo as vantagens de um banco de dados informacional. Com estas ferramentas será possível responder de forma rápida e simples perguntas que no sistema antigo seriam extremamente difíceis e mesmo inviáveis ao usuário final que não possua conhecimento técnico. A figura 10 apresenta uma pequena parte do banco operacional da Clínica do Leite. Se um pesquisador está interessado em realizar um estudo associando resultados de algum tipo de análise com as cidades onde os animais analisados estão localizados, teria que fazer um busca (*query*) no banco de dados passando por uma cadeia de tabelas interligadas (banco de dados relacional) de forma a conseguir relacionar os resultados das análises aos produtores e então as cidades onde estão localizados. Depois ainda poderia ser necessário tratar os dados, pois seu formato pode não estar adequado para se fazer cálculos, por exemplo.



Figura 10 – Exemplo de estrutura relacional para banco de dados

Com o modelo informacional do DW (Figura 11), a tabela de resultados de análises esta diretamente ligada a todas as características de interesse e os dados já estão tratados e formatados de forma a tornar simples a realização de cálculos e agrupamentos.



Figura 11 – Exemplo de estrutura multidimensional

Além disso, não será mais necessário realizar as buscas diretamente no banco de dados, pois pode ser utilizada a ferramenta OLAP para geração de cubos

e relatórios customizados de forma muito mais intuitiva e sem a necessidade de conhecimento técnico. Dessa forma, se reduz o esforço do pesquisador para se chegar aos resultados desejados e o tempo economizado pode ser utilizado de melhor forma na análise zootécnica dos resultados encontrados.

Com a maior facilidade de acesso aos dados históricos da Clínica do Leite espera-se incentivar seu estudo, e desta forma contribuir para aumentar o conhecimento do cenário leiteiro do estado e até do país. Consequentemente, pesquisadores poderão desenvolver seus estudos com maior eficiência e qualidade, contribuindo para o direcionamento de melhorias neste cenário.

5.1.2 Escopo do projeto

A seguir, é apresentado o escopo do projeto, dividido por cada uma de suas frentes:

1. Laboratório – O principal mote desse segmento do projeto é proporcionar aos usuários um cruzamento de dados completo relacionando as informações das fazendas e indústrias, por exemplo:

Dados cadastrais das fazendas e indústrias (p.e. nome, meso-região, micro-região, estado);

Unidade analisada (p.e. código, nome do produtor, raça de um animal);

Análise de Leite (p.e. resultados de análise de CCS, CBT, gordura, etc);

Assim, cruzando esses dados, o usuário poderá responder a perguntas de forma mais fácil e rápida através de simples filtros. Por exemplo:

Quais as análises um determinado grupo de fazendas realizou no último ano?

Como foi a demanda de uma determinada análise em um determinado período?

Quantas análises de CCS foram feitas por uma determinada fazenda no ano passado?

Qual a média do resultado de CBT para fazenda? E para indústrias?

Resultado de AGL x temperatura de chegada da amostra no laboratório?

Qual é quantidade de amostras enviadas para indústrias em um determinado período?

Podem ainda ser desenvolvidos gráficos com o levantamento desses dados cruzados, para um diagnóstico analítico mais intuitivo sobre o comportamento de fazendas e indústrias de laticínios.

Caso seja necessário, os usuários podem ainda criar seus próprios relatórios de maneira bem simples, apenas selecionando quais os modelos de negócios que serão utilizados, quais os dados a serem disponibilizados no relatório, como se dará a divisão em blocos (grupos) do relatório e quais os filtros aplicados a esses dados.

2. Pesquisa – O objetivo deste segmento é realizar, de acordo com a necessidade, levantamentos de dados zootécnicos de gado leiteiro.

Para garantir que os mais diversos temas de pesquisas zootécnicas sobre gado leiteiro possam ser suportadas por este segmento, é necessário que todos os dados zootécnicos e resultados de análise de leite dos animais estejam mapeados no banco de dados (DW). Desta forma, o pesquisador ou o responsável por realizar os levantamentos de dados poderá realizar a busca que for mais conveniente para cada pesquisa.

Além disso, como a demanda nesta frente varia para cada pesquisa, os relatórios customizados constituirão uma ferramenta extremamente importante dentro deste segmento.

3. BI Fazenda – O principal objetivo desse segmento do projeto é dar aos usuários um cruzamento de dados completo, relacionando:

Dados cadastrais das fazendas e indústrias (p.e. nome, grupo, cidade);

Coeficientes zootécnicos (média de produção de leite, taxa de prenhez, média do peso de novilhas, taxa de animais com a sanidade casco diagnosticada, média da condição corporal, entre outros);

Coeficientes de análise de leite (G/P, %novas, Uréia, entre outros);

Assim, cruzando esses dados, assim como seus atributos (média, valor máximo, valor mínimo, dispersão, número de observações, quartil, entre outros), o usuário terá todas as informações das fazendas através de simples filtros. Por exemplo:

Qual a produção das fazendas de determinada região;

Qual a relação entre a variação da taxa de prenhez do rebanho ao longo do tempo e a região na qual se encontram as fazendas;

Qual a média de produção de leite em função da raça dos animais;

Qual a variação da taxa de animais com a sanidade casco diagnosticada em função da taxa de prenhez do rebanho.

Ao final do desenvolvimento desta frente, será possível efetuar comparações entre fazendas, entre uma fazenda e as demais situadas na mesma região, entre uma fazenda e as demais que utilizam os serviços da Clínica do Leite e assim por diante. Ou seja, será possível realizar qualquer tipo de comparação envolvendo uma fazenda e qualquer outro conjunto de fazendas.

Assim sendo, será possível definir as metas para cada fazenda, sendo que esta meta deve ser superior à média das demais fazendas para que em longo prazo

aumente significativamente esta média, melhorando assim a qualidade do leite produzido pelo estado, por exemplo.

5.2 Análise dos Dados

Nesta etapa, foi realizada a modelagem do DW. A estrutura de tabelas fato e dimensões foi definida considerando-se o levantamento de requisitos realizado na etapa anterior e também as perguntas as quais se pretende responder através de cada conjunto de tabelas (fato e suas dimensões).

Abaixo estão apresentados exemplos da forma como a modelagem foi realizada:

5.2.1 Análise de Leite



Figura 12 – Modelagem da tabela de Análise de Leite

Esta tabela Fato (figura 12) é baseada na maior tabela do banco operacional, onde estão armazenados os resultados das diversas análises de leite realizadas pela Clínica do Leite. Esta tabela Fato se liga as dimensões Produtor, Unidades de Análise, Indicador (CCS, CBT, etc) e Ordem de Serviço, por meio de uma chave (ID) do tipo inteiro e que é gerado sequencialmente nas tabelas dimensão quando são criadas. Após a criação das tabelas dimensão, a tabela Fato é criada com uma coluna de ID's para cada dimensão e uma coluna contendo o resultado da análise. Então, cada linha da tabela Fato ligará a um valor de análise o Produtor que a requisitou, a Unidade de Análise (UA, pode ser uma vaca de um produtor ou um caminhão de uma cooperativa, por exemplo) que gerou a amostra, a análise que foi realizada e a ordem de serviço (OS, requisição de serviço realizada pelo produtores e empresas) a qual a análise em questão pertence. Abaixo um exemplo da forma como uma questão poderá ser respondida:

Qual a média de gordura nas amostras analisadas no ano de 2008 na Meso-Região de Presidente Prudente? A partir da dimensão OS temos a informação da data em que a análise foi realizada, a partir da dimensão Produtor temos a informação da Meso-Região e através da dimensão Indicador filtramos por análises somente de gordura, após todas essas filtrações só precisamos tirar a média dos resultados de análises restantes.

5.2.2 Coeficientes Zootécnicos



Figura 13 – Modelagem da tabela de Coeficientes Zootécnicos

A tabela Fato de Coeficientes Zootécnicos (figura 13) é baseada em uma tabela do banco operacional, onde estão armazenados os resultados das diversas análises de leite realizadas pela Clínica do Leite consolidados por mês. E a tabela Fato de Análise de Leite pode responder perguntas envolvendo todas as características de suas dimensões.

5.3 Seleção dos métodos

Para realizar o processo de ETL (Extract Transform Load), que irá extrair os dados do banco operacional, limpar e organizar esses dados e então carregá-los no DW, foi escolhida a ferramenta Pentaho Data Integration, também conhecido como Kettle, que é open source.

Esta ferramenta é de fácil utilização, com um ambiente de trabalho intuitivo e gráfico, que permite a criação de transformações e *jobs* complexos sem a necessidade de programar por linhas de código.

Tem uma biblioteca de transformações rica, com aproximadamente 100 objetos (passos de transformações), o que facilita bastante o desenvolvimento. Estes objetos são “ícones” com funções como extrair dados de uma tabela de banco de dados ou agrupar informação segundo alguma regra, por exemplo.

5.4 Construção

Após as etapas de levantamento dos requisitos, análise dos dados e seleção dos métodos foi iniciada a etapa de construção do DW.

Basicamente, foi desenvolvido um conjunto de transformações que retiram dados do banco operacional da Clínica do Leite, os transformam e os carregam no novo banco (DW). Estas transformações foram criadas através da ferramenta Pentaho Data Integration (ou Kettle), e representam de forma bastante intuitiva o fluxo dos dados desde o banco operacional até o DW. A figura 14 apresenta um exemplo deste fluxo em uma transformação:

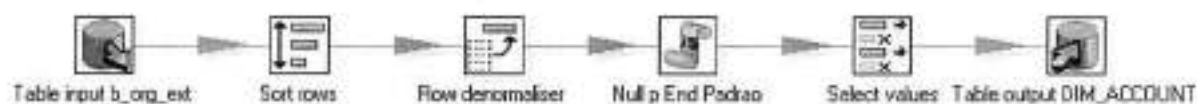


Figura 14 – Exemplo do fluxo de dados de uma transformação

Esta transformação, por exemplo, é responsável por criar a tabela do DW que contem os dados cadastrais dos produtores e empresas ligados a Clínica do Leite. Seu fluxo de dados, de forma bastante resumida, primeiramente retira os dados da tabela “b_org_ext” do banco operacional através de uma *query* definida no passo da transformação chamado “Table input b_org_ext”, onde já ocorrem algumas transformações como filtragens apenas dos dados relevantes, por exemplo. Após este passo inicial é realizada uma serie de transformações nos dados em cada um dos passos “Sort rows”, “Row denormaliser”, “Null p End Padrão” e “Select values”, necessárias para se obter o formato desejado dos dados no DW. E finalmente, os

dados com o formato adequado são carregados no DW através do passo "Table output DIM_ACCOUNT", onde "DIM_ACCOUNT" é o nome da tabela destino no DW.

Além desta transformação outras foram criadas, cada uma objetivando a criação de uma das tabelas do DW. Após a execução de todas as transformações o DW está pronto para ser utilizado.

A seguir será apresentado, de forma bastante resumida, todo o processo de construção de um dos conjuntos Fato/Dimensões. Mais detalhes são disponibilizados no Apêndice.

5.4.1 Fato e dimensões

A figura 15 apresenta a estrutura Fato e Dimensões da tabela de Coeficientes Zootécnicos consolidados por produtor ou empresa (dimensão clientes), por indicador (contagem bactéria, porcentagem de gordura, etc), por tipo de UA (unidade de análise, por exemplo, um animal ou o caminhão de uma cooperativa) e por mês. Isso quer dizer que cada registro armazenado nesta tabela terá como resultado a média no mês dos resultados de determinado indicador, referentes a um determinado produtor ou empresa e uma determinada UA. Ou seja, esta tabela Fato está ligada a quatro dimensões.

Os campos desta tabela podem ser divididos em chaves (ou ID, que liga cada registro da tabela Fato ao respectivo registro na tabela dimensão) e medidas ou resultados (neste caso, médias, contagens e outros coeficientes zootécnicos).

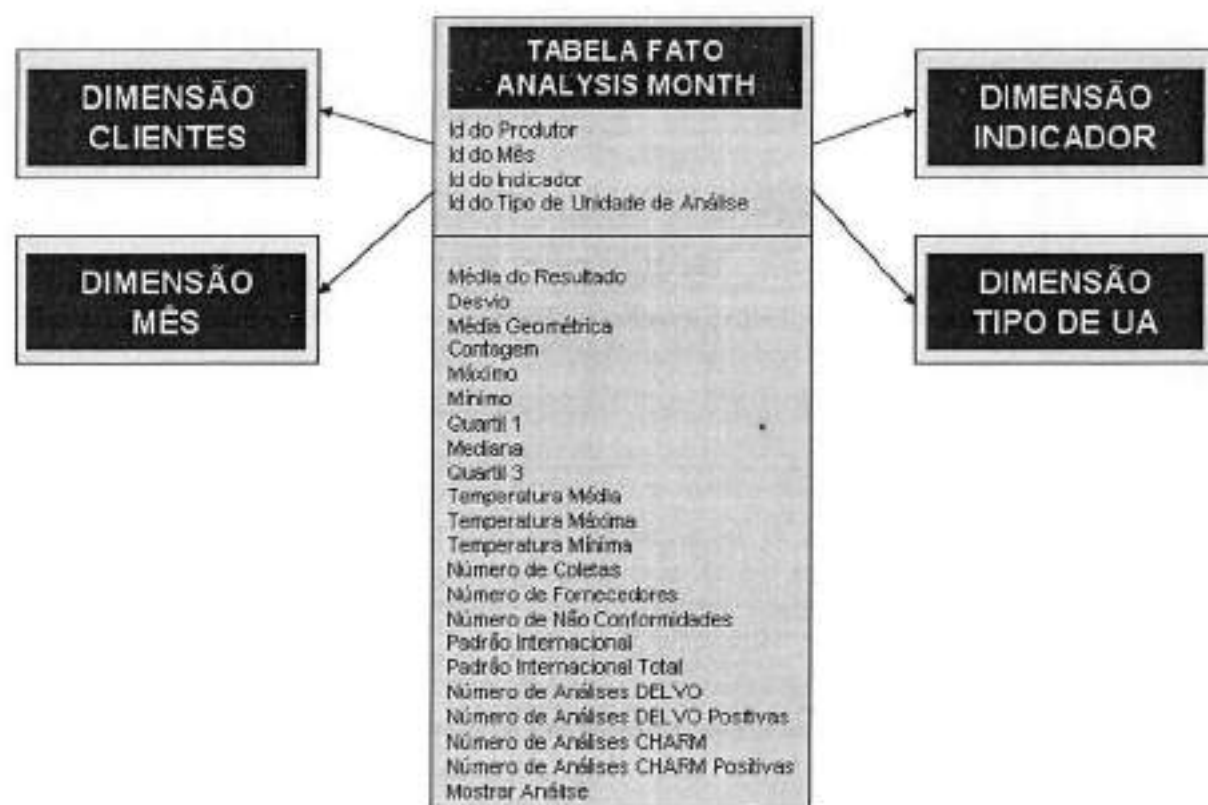


Figura 15 – Modelagem da tabela de Coeficientes Zootécnicos por tipo de UA e mês

5.4.2 Dimensão de clientes

A dimensão de clientes ou produtores é a tabela do DW onde esta armazenada a informação a respeito de produtores e empresas que utilizam os serviços da Clínica do Leite. Esta dimensão foi construída através da transformação apresentada na figura 14.

O primeiro passo (figura 16) desta transformação, denominado "Table input b_org_ext", é onde a informação relevante do banco operacional é obtida através de uma *query*. Esta *query* reúne informações de cinco tabelas do banco operacional, ou seja, neste ponto já se começa a moldar a tabela dimensão que será criada, pois o que antes necessitava de uma busca em todas estas tabelas, agora será necessário apenas uma busca, pois já estará tudo previamente armazenado em uma única tabela do DW.

Table Input

Step name: Table input b_org_ext

Connection: Clínica do Leite - QA

SQL

Get SQL select statement...

```

SELECT
  OE.ROWID
  OE.NAME
  OE.GROUP_ACCOUNT
  OE.SEGMENTATION
  OE.ACCNT_TYPE
  OE.SIGSIF_TYPE
  U.NAME
  P.NAME
  OA.TYPE
  OA.ADDR
  OA.COMPLEMENT
  OA.CITY
  OA.STATE
  GI.MESO_NAME
  GI.MESO_CODE
  GI.MICRO_NAME
  GI.MICRO_CODE
  OE.CODE
  OE.STATUS
  GI.LATITUDE
  GI.LONGITUDE
FROM b_org_ext OE
INNER JOIN b_user U ON OE.RESPONSIBLE_ID = U.ROWID AND U.DELETED_BY = ''
INNER JOIN b_pri_lst P ON OE.PRICE_LIST_ID = P.ROWID AND P.DELETED_BY = ''
INNER JOIN b_org_addr OA ON OA.PAR_ROW_ID = OE.ROWID AND OA.DELETED_BY = ''
INNER JOIN b_gov_ident GI ON GI.ROWID = OA.REGION_ID AND GI.DELETED_BY = ''
WHERE OE.TYPE = 'Cliente' AND OE.DELETED_BY = ''

```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Executes for each row? ☐

Limit size 0

OK Cancela Preview

Figura 16 – Conteúdo do passo "Table input b_org_ext"

O segundo passo (figura 17) desta transformação, denominado "Sort rows", é responsável por ordenar os registros obtidos, através do primeiro passo, usando o campo "ROWID". Este processo é um pré-requisito para o próximo passo que irá realizar a desnormalização de parte dos campos, referentes ao endereço dos produtores.

Sort rows

Nome do Step: Sort rows

Sort directory: %%java.io.tmpdir%% Navega...

TMP-file prefix: out

Sort size (rows in memory):

Free memory threshold (in %): 25

Compress TMP Files? ☐

Only pass unique rows? ☐

Fields :

	Fieldname	Ascending	Case sensitive compare?
1	ROWID	S	N

OK Cancela Obtem campos

Figura 17 – Conteúdo do passo "Sort rows"

O terceiro passo (figura 18) desta transformação, denominado "Row denormaliser", é responsável por desnormalizar os campos referentes a endereço, ou seja, se antes haviam seis registros com endereços diferentes, passará a existir apenas um registro com seis vezes mais campos destinados a cada um dos endereços do produtor. Isto é necessário, pois a Clínica do Leite cadastra seis tipos de endereços para cada cliente, além do endereço padrão, existem também endereços para cobrança, envio de análise, recebimento de material, envio de nota fiscal e ponto de coleta de amostras.

The screenshot shows the 'Denormaliser' application window. At the top, the title bar says 'Denormaliser'. Below it, there are two input fields: 'Step name' with the value 'Row denormaliser' and 'The key field' with the value 'TYPE'. Below these, a label reads 'The fields that make up the grouping:'. Under this label is a table with two columns: an index and 'Group field'. The first row has index '1' and 'ROWID'. To the right of this table is a 'Get Fields' button. Below the grouping table, a label reads 'Target fields:'. Under this label is a table with five columns: an index, 'Target fieldname', 'Value fieldname', 'Key value', and 'Type'. The table contains 13 rows of data. To the right of this table is a 'Get lookup fields' button. At the bottom of the window are 'OK' and 'Cancela' buttons.

	Group field
1	ROWID

	Target fieldname	Value fieldname	Key value	Type
1	ADDR_PADRAO	ADDR	Padrão	String
2	COMPLEMENT_PADRAO	COMPLEMENT	Padrão	String
3	CITY_PADRAO	CITY	Padrão	String
4	STATE_PADRAO	STATE	Padrão	String
5	MESO_PADRAO	MESO_NAME	Padrão	String
6	MESO_CODE_PADRAO	MESO_CODE	Padrão	String
7	MICRO_PADRAO	MICRO_NAME	Padrão	String
8	MICRO_CODE_PADRAO	MICRO_CODE	Padrão	String
9	LATITUDE_PADRAO	LATITUDE	Padrão	Number
10	LONGITUDE_PADRAO	LONGITUDE	Padrão	Number
11	ADDR_COBRANCA	ADDR	Cobrança	String
12	COMPLEMENT_COBRANCA	COMPLEMENT	Cobrança	String
13	CITY_COBRANCA	CITY	Cobrança	String

Figura 18 – Conteúdo do passo "Row denormaliser"

O quarto passo (figura 19) desta transformação, denominado "Null p End Padrão", é responsável por atribuir, via java script, os valores do endereço padrão para os demais campos de endereços que estiverem vazios.

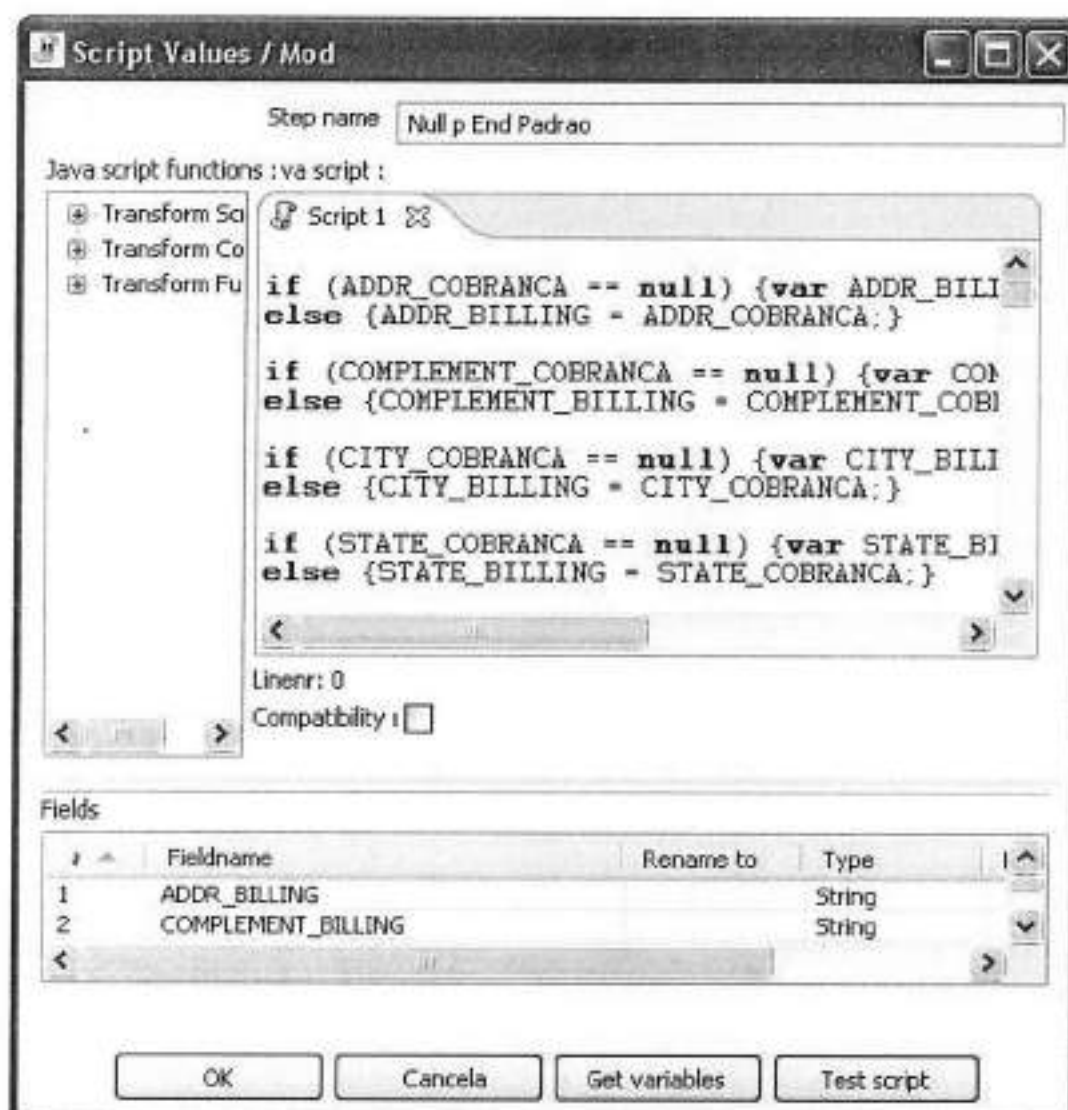


Figura 19 – Conteúdo do passo "Null p End Padrao"

O quinto passo (figura 20) desta transformação, denominado "Select values", é responsável por adequar os nomes dos campos do fluxo de dados da transformação aos nomes dos campos da tabela destino.

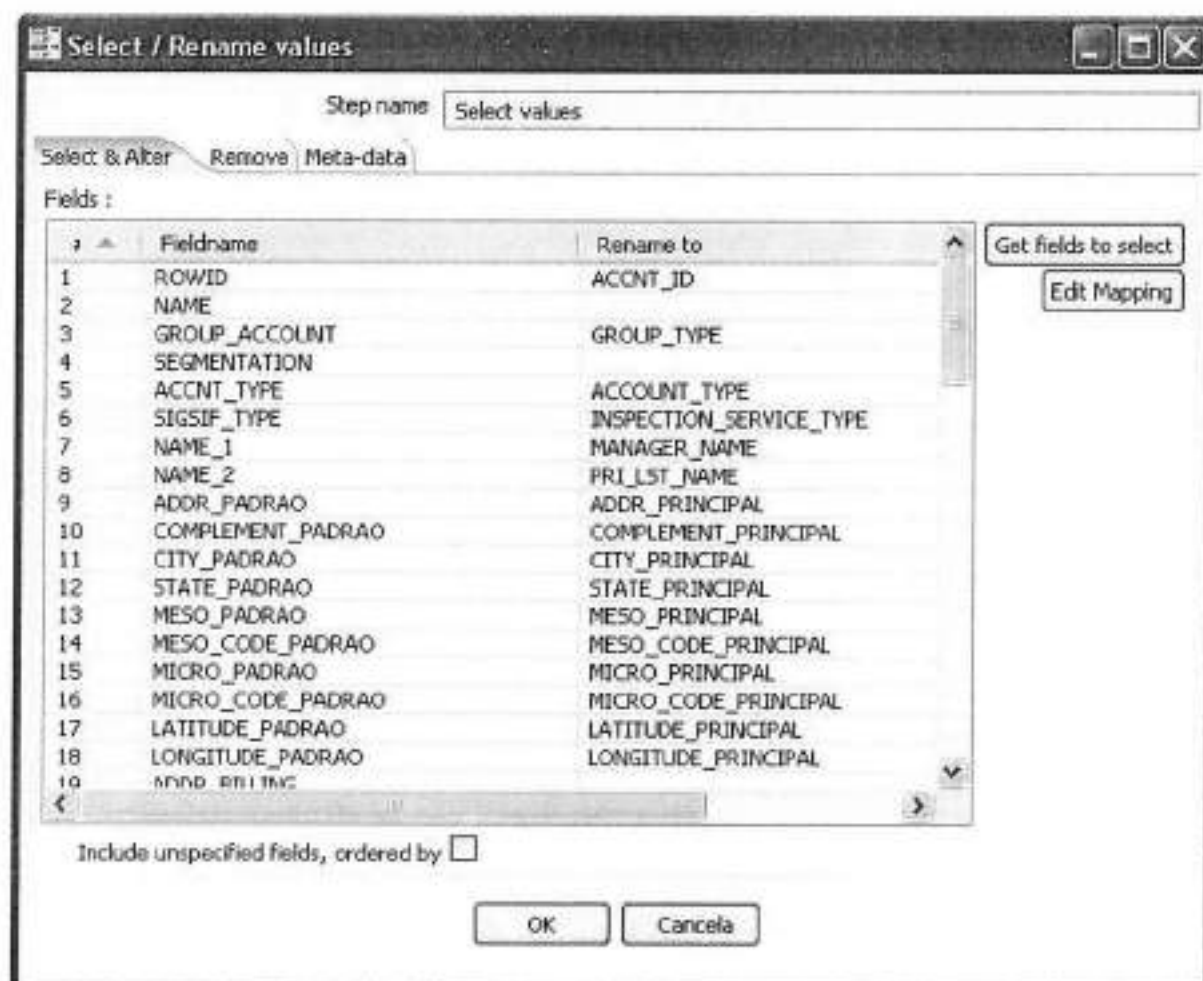


Figura 20 – Conteúdo do passo "Select values"

O sexto e ultimo passo (figura 21) desta transformação, denominado "Table output DIM_ACCOUNT", é responsável por inserir os registros no DW, criando assim a tabela dimensão de clientes "DIM_ACCOUNT".

Table output

Nome do Step: Table output DIM_ACCOUNT

Connection: DW [Edit...] [New...]

Target schema: []

Target table: DIM_ACCOUNT [Navega...]

Commit size: 100

Truncate table: ☒

Ignore insert errors: ☐

Use batch update for inserts: ☒

Partition data over tables: ☐

Partitioning field: []

Partition data per month: ☒

Partition data per day: ☐

Is the name of the table: ☐

Field that contains name of: []

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key: []

[OK] [Cancela] [SQL]

Figura 21 – Conteúdo do passo "Table output DIM_ACCOUNT"

5.4.3 Dimensão mês

Esta dimensão também poderia ser chamada de dimensão tempo ou datas, pois nada mais é do que uma tabela onde cada registro representa um dia. No entanto, como a tabela fato "ANALYSIS_MONTH" armazena apenas registros consolidados por mês, é conveniente para este caso o nome dimensão mês.

Esta dimensão foi construída através da transformação apresentada na figura 22.

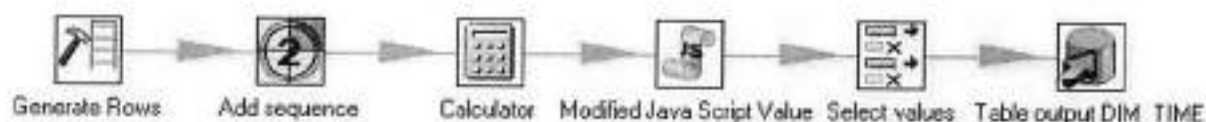


Figura 22 – Transformação da dimensão mês

Os quatro primeiros passos desta transformação, denominados "Generate Rows", "Add sequence", "Calculator" e "Modified Java Script Value", geram cinco campos: o campo "ID" inteiro e incremental, que representa cada dia deste o primeiro dia da década de 90, na qual a Clínica do Leite iniciou suas atividades, o campo "DATE" com as datas (tipo date) deste período, e ainda os campos "YEAR", "MONTH" e "DAY".

Os dois últimos passos "Select values" e "Table output DIM_TIME" realizam as mesmas funções já relatadas no tópico sobre a dimensão de clientes.

5.4.4 Dimensão de indicadores

A dimensão de indicadores é onde fica armazenada a informação sobre todos os indicadores analisados pela Clínica do Leite, como porcentagem de gordura, proteína ou lactose presente em amostras de leite, por exemplo.

Esta dimensão foi construída através da transformação apresentada na figura 23.

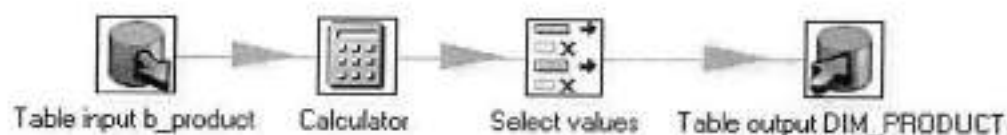


Figura 23 – Transformação da dimensão de indicadores

Basicamente, os dados relevantes são obtidos do banco operacional através do passo "Table input b_product". A partir deste passo, são obtidos os campos:

"PRODUCT_ID", com o código do indicador utilizado pela Clínica do Leite (não é o ID utilizado pela tabela fato que é gerado automaticamente e de forma incremental no próprio banco de dados), o campo "NAME", com o nome do indicador, o campo "TYPE", com o tipo do indicador. O campo "DESCRIPTION" é obtido através da junção de outros campos no passo "Calculator".

5.4.5 Dimensão de tipos de UA

A dimensão de tipos de UA é onde fica armazenada a informação sobre todos os tipos de unidade de análise com as quais a Clínica do Leite trabalha. Uma unidade de análise pode ser uma vaca para o caso do cliente ser um produtor ou pode ser um caminhão de uma cooperativa ou empresa leiteira, por exemplo.

Esta dimensão foi construída através da transformação apresentada na figura 24.

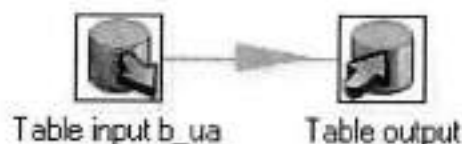


Figura 24 – Transformação da dimensão de tipos de UA

Esta transformação apenas agrupa por tipo de UA os registros da tabela que tem as informações sobre UA's para obter todos os tipos existentes.

5.4.6 Tabela fato de coeficientes zootécnicos

O processo de construção desta tabela fato foi dividido em várias partes, cada uma com sua transformação, e todas estas transformações foram organizadas

através de um *job*, que pode ser acionado via “batch”, por exemplo. Este *job* esta apresentado na figura 25.



Figura 25 – Job da tabela Fato de coeficientes zootécnicos

O primeiro passo, denominado “START”, apenas dá início ao *job* e oferece algumas opções de agendamento e repetição.

Os demais passos serão apresentados, resumidamente, nos tópicos a seguir.

5.4.6.1 Transformação coeficientes zootécnicos

Esta transformação (figura 26) retira diretamente de uma tabela do banco operacional a maior parte dos campos de medidas necessários a tabela fato. São eles: Média, Desvio, Média Geométrica, Contagem, Máximo, Mínimo, Quartil 1, Mediana, Quartil 3, Número de Fornecedores, Número de Não Conformidades e Mostrar Análise. Além, das chaves (ou ID) que ligam a fato a suas dimensões: ID do Produtor, ID do Mês, ID do Indicador e ID do Tipo de Unidade de Análise.

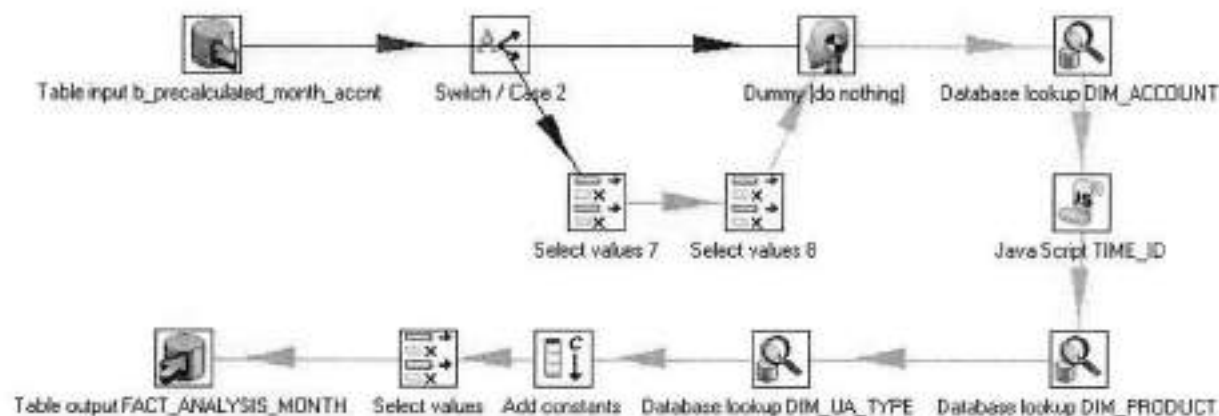


Figura 26 – Transformação Coeficientes Zootécnicos

No passo “Table input b_precalculated_month_accnt” são obtidos todos os campos relevantes a partir de uma tabela do banco operacional.

Nos passos “Switch / Case”, “Select values 7 e 8” e “Dummy” é realizada uma divisão no fluxo de dados para alguns indicadores e a inversão de valores dos campos “Quartil 1” e “Quartil 3”. Isto é necessário para facilitar a geração de relatórios padronizando estes campos para todos os indicadores, pois alguns indicadores têm valores melhores quanto menores os resultados e outros têm valores melhores quanto maiores os resultados.

Os passos “Database lookup DIM_ACCOUNT, DIM_PRODUCT e DIM_UA_TYPE” são responsáveis pela troca dos respectivos códigos para clientes, indicadores e tipo de UA pelas chaves utilizadas nas dimensões.

O passo “Java Script TIME_ID” gera a chave para a dimensão mês. Neste caso não é necessário fazer “Lookup” na dimensão (ou seja, buscar na dimensão o valor das chaves), pois esta chave pode ser calculada a partir da data de referencia no registro, o que é mais eficiente.

O passo “Add constants” cria o campo “Mostrar Análise” e atribui o valor “N” para todos os registros, este campo tem a informação se o resultado pode ou não ser apresentado no relatório (é necessário que o registro seja verificado por um funcionário da Clínica do Leite antes de liberado). Em uma transformação posterior

este campo será atualizado, pois até este ponto nenhum resultado poderá ser apresentado.

5.4.6.2 Transformação temperatura

Esta transformação (figura 27) é responsável por três campos da tabela fato, que são: Temperatura Média, Máxima e Mínima. Estes campos são respectivamente a temperatura média, máxima e mínima dos registros de determinado produtor, tipo de UA e indicador consolidados por mês.

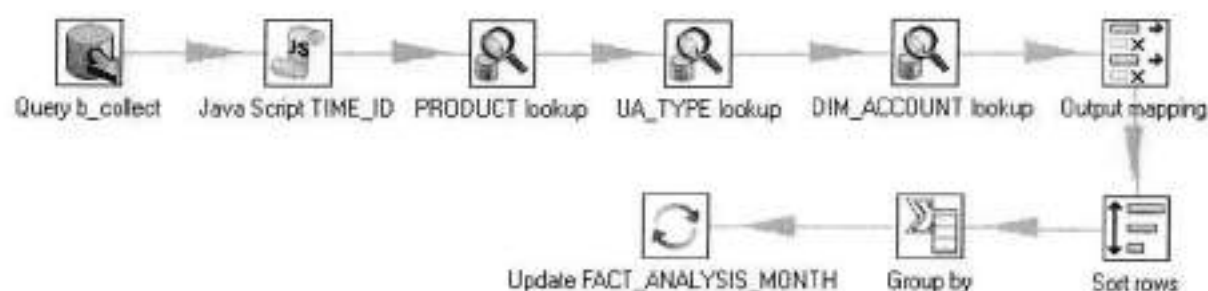


Figura 27 – Transformação temperatura

Os passos iniciais desta transformação realizam a obtenção dos dados relevantes a partir do banco operacional e a criação das chaves relacionais.

No passo “Group by” ocorre a consolidação dos registros com relação as dimensões e o calculo dos temperaturas média, máxima e mínima.

No ultimo passo “Update FACT_ANALYSIS_MONTH” (figura 28) ocorre a atualização dos três campos referentes à temperatura. Este passo é diferente do “Table output” apresentado anteriormente, pois nenhum registro novo é inserido na tabela fato do DW, apenas são atualizados três campos de todos os registros. Estes campos já existiam, porém com valor nulo. As quatro chaves de cada registro são comparadas antes da atualização.

Update

Step name: Update FACT_ANALYSIS_MONTH

Connection: DW [Edit...] [New...]

Target schema: []

Target table: FACT_ANALYSIS_MONTH [Browse...]

Commit size: 100

Ignore lookup failure? ☒ Flag field (key found) []

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1
1	ACCOUNT_ID	=	ACCOUNT_ID
2	UA_TYPE_ID	=	UA_TYPE_ID
3	PRODUCT_ID	=	PRODUCT_ID
4	TIME_ID	=	TIME_ID

[Get fields]

Update fields:

#	Table field	Stream field
1	AVG_TEMP	AVG_TEMP
2	MAX_TEMP	MAX_TEMP
3	MIN_TEMP	MIN_TEMP

[Get update fields]

[OK] [Cancel] [SQL]

Figura 28 – Conteúdo do passo "Update FACT_ANALYSIS_RESULT"

5.4.6.3 Transformação número de coletas

Essa transformação (figura 29) visa contar o número de datas de coleta de amostras referente a cada cliente e mês, logo, para esta transformação apenas duas dimensões são relevantes, a dimensão de clientes e a dimensão mês.

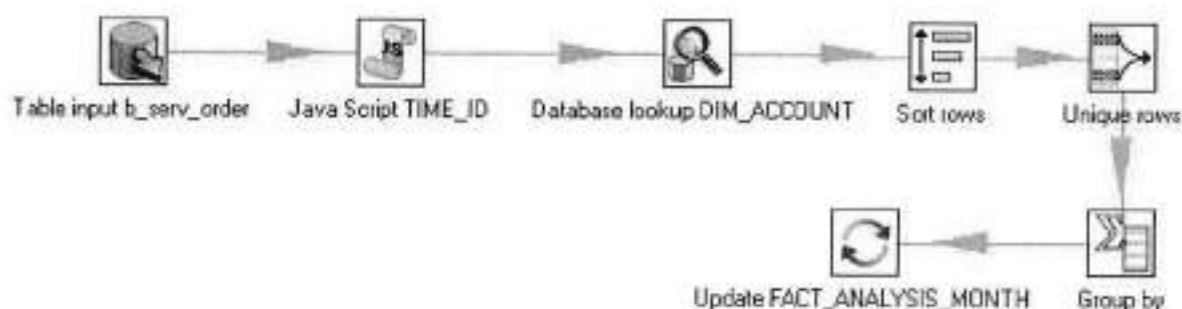


Figura 29 – Transformação número de coletas

Os passos iniciais desta transformação realizam a obtenção dos dados relevantes a partir do banco operacional e a criação das chaves relacionais.

O passo "Unique rows" garante que nenhum registro duplicado chegará ao passo seguinte, ou seja, se existir um produtor que em determinado mês teve duas datas de coletas iguais, apenas uma será considerada, a outra provavelmente é fruto de algum erro no banco de dados operacional.

No passo "Group by" ocorre a consolidação dos registros com relação as dimensões de clientes e mês, e o calculo do número de datas de coleta.

Novamente foi utilizado o passo "Update", pois é necessário apenas atualizar o campo número de coletas. Mas, diferentemente do passo anterior, neste passo apenas serão comparadas as chaves das dimensões de clientes e mês antes da atualização.

5.4.6.4 Transformação padrão internacional total

O padrão internacional ao qual este campo se refere é uma regra que dita que uma determinada amostra de leite deve ter um valor máximo aceitável dos indicadores CCS e CBT. Nesta transformação (figura 30) será calculado o campo "Padrão Internacional Total" que é o número de UAs consolidadas sobre produtor e mês que realizaram análises de CCS e CBT. Ou seja, é o número máximo de UAs

de um determinado produtor e em determinado mês que podem estar dentro do padrão internacional.

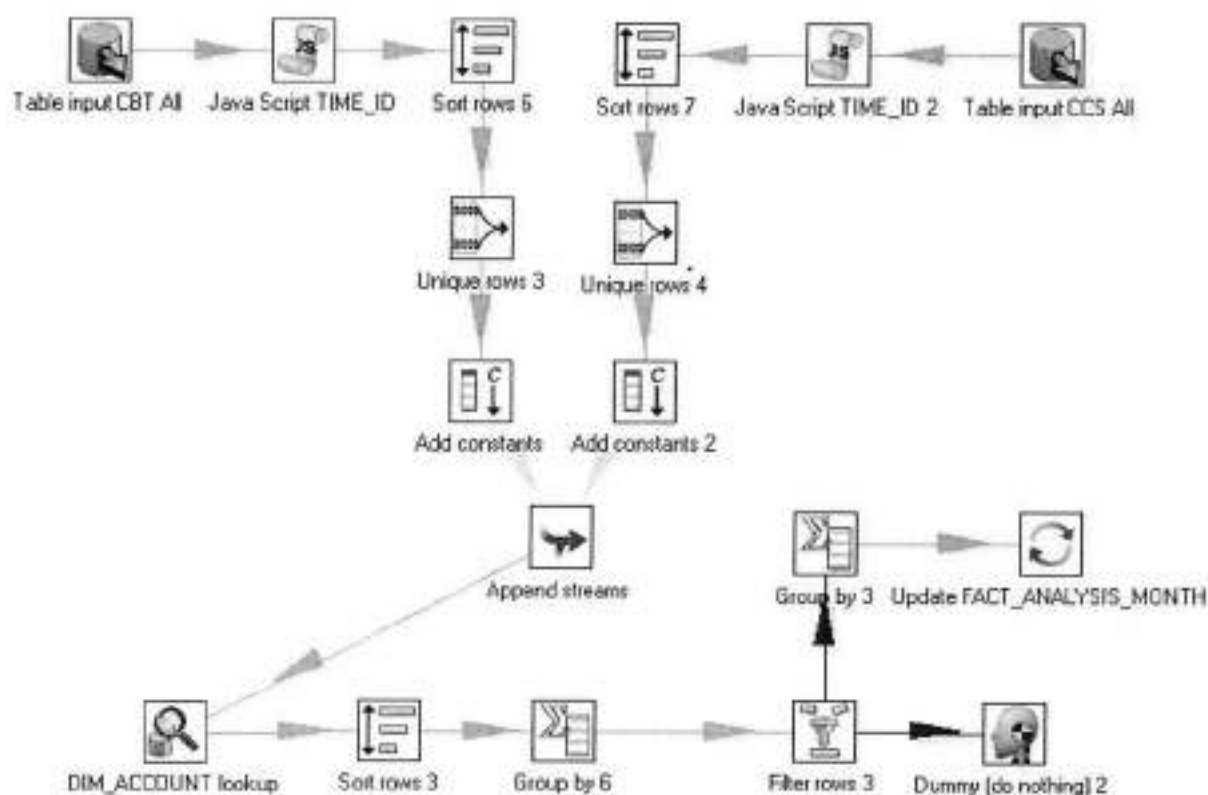


Figura 30 – Transformação padrão internacional total

Basicamente, através dos passos "Unique rows 3 e 4" um novo campo é adicionado ao fluxo de dados com o valor "1" para todos os registros, então no passo "Group by 6" é realizada a consolidação dos registros por produtor, mês e UA, somando-se o campo recém criado, ou seja, as UAs que realizaram análise tanto de CCS quanto de CBT terão o resultado "2" neste campo após este passo. Então no passo "Filter rows 3" são excluídos registro com resultado "1", e finalmente no passo "Group by 3" ocorre a consolidação por produtor e mês contando-se o número de UAs que podem estar dentro do padrão internacional.

5.4.6.5 Transformação padrão internacional

Após o cálculo do número total de UAs com possibilidade de adequação ao padrão internacional, nesta transformação (figura 31) é calculado o número efetivo de UAs que se adequam a este padrão. Assim, com estes dois campos é possível se calcular a porcentagem de UAs no padrão internacional, por exemplo.

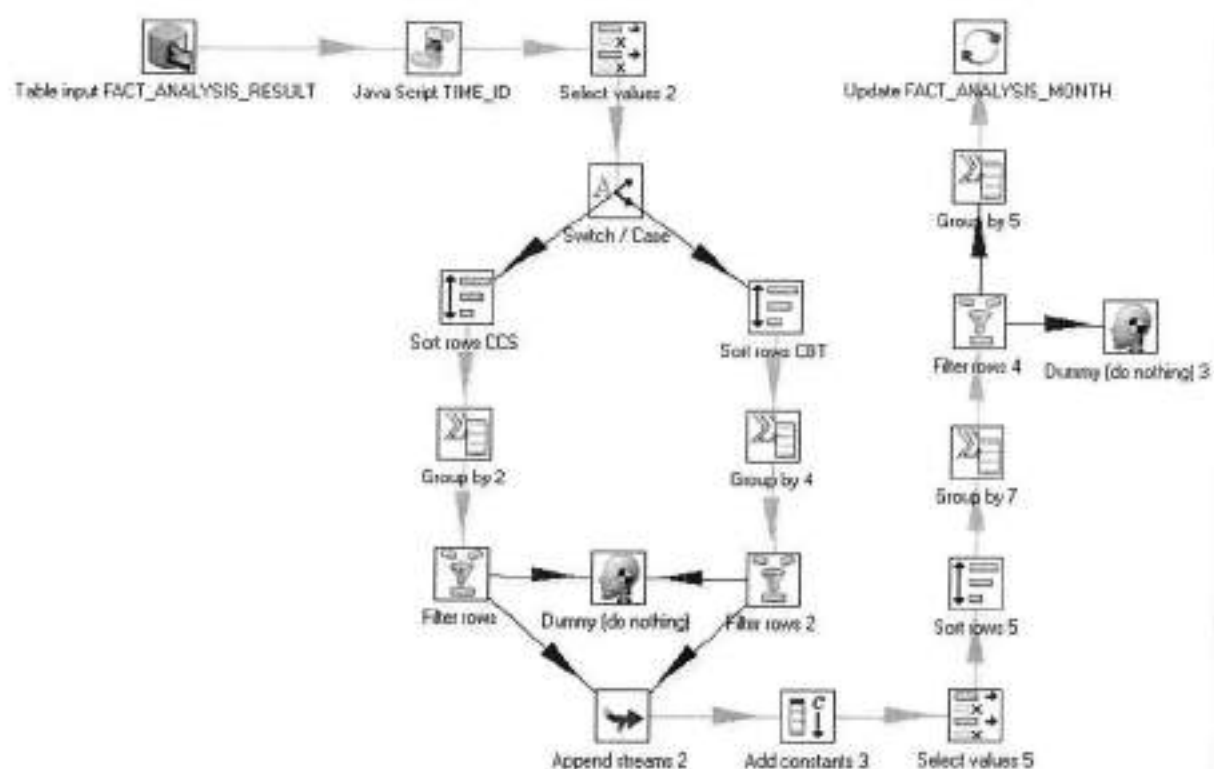


Figura 31 – Transformação padrão internacional

A diferença desta transformação para a anterior é que nos passos "Filter rows" e "Filter rows 2" as UAs cujos resultados de CCS e CBT estão acima do permitido são descartadas. Logo no passo "Group by 7" só são somadas as UAs que estão dentro do padrão internacional.

Outra diferença é que no passo "Table input FACT_ANALYSIS_RESULT" as informações são obtidas a partir de outra tabela fato do DW. Isto é necessário, pois como existem filtros de resultados de análises, é necessário que os resultados estejam limpos e padronizados.

5.4.6.6 Transformação número de análises DELVO e CHARM

Nesta transformação (figura 32) é calculado o número de UAs que realizaram análises de ATB – DELVO e ATB – CHARM consolidados por todas as dimensões da tabela fato.

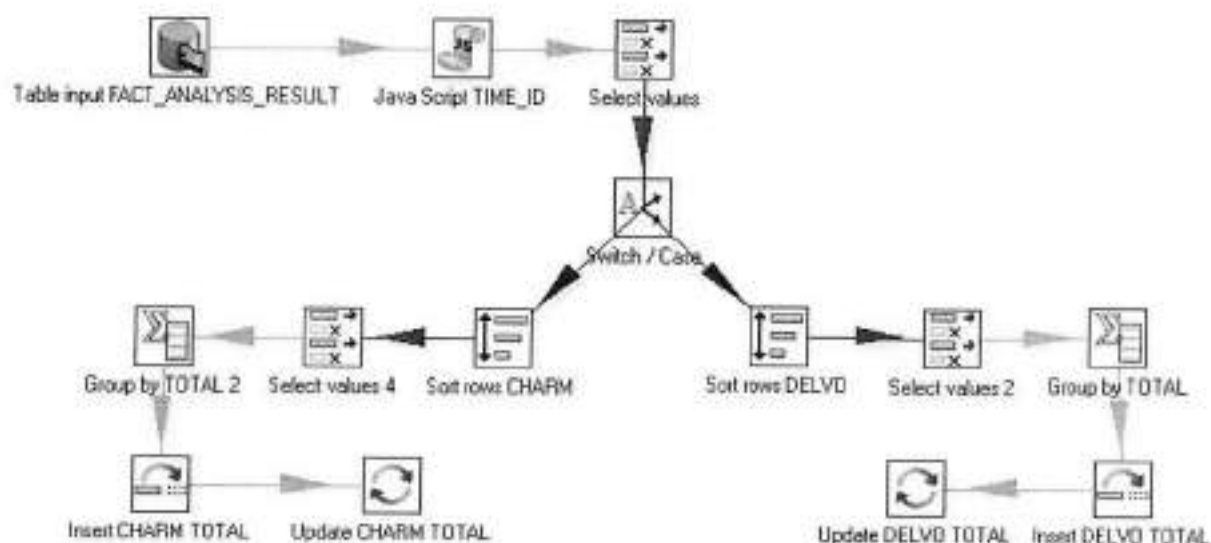


Figura 32 – Transformação número de análises DELVO e CHARM

5.4.6.7 Transformação número de análises DELVO e CHARM positivas

Nesta transformação (figura 33) é calculado o número de UAs de ATB – DELVO e ATB – CHARM com resultado positivo.

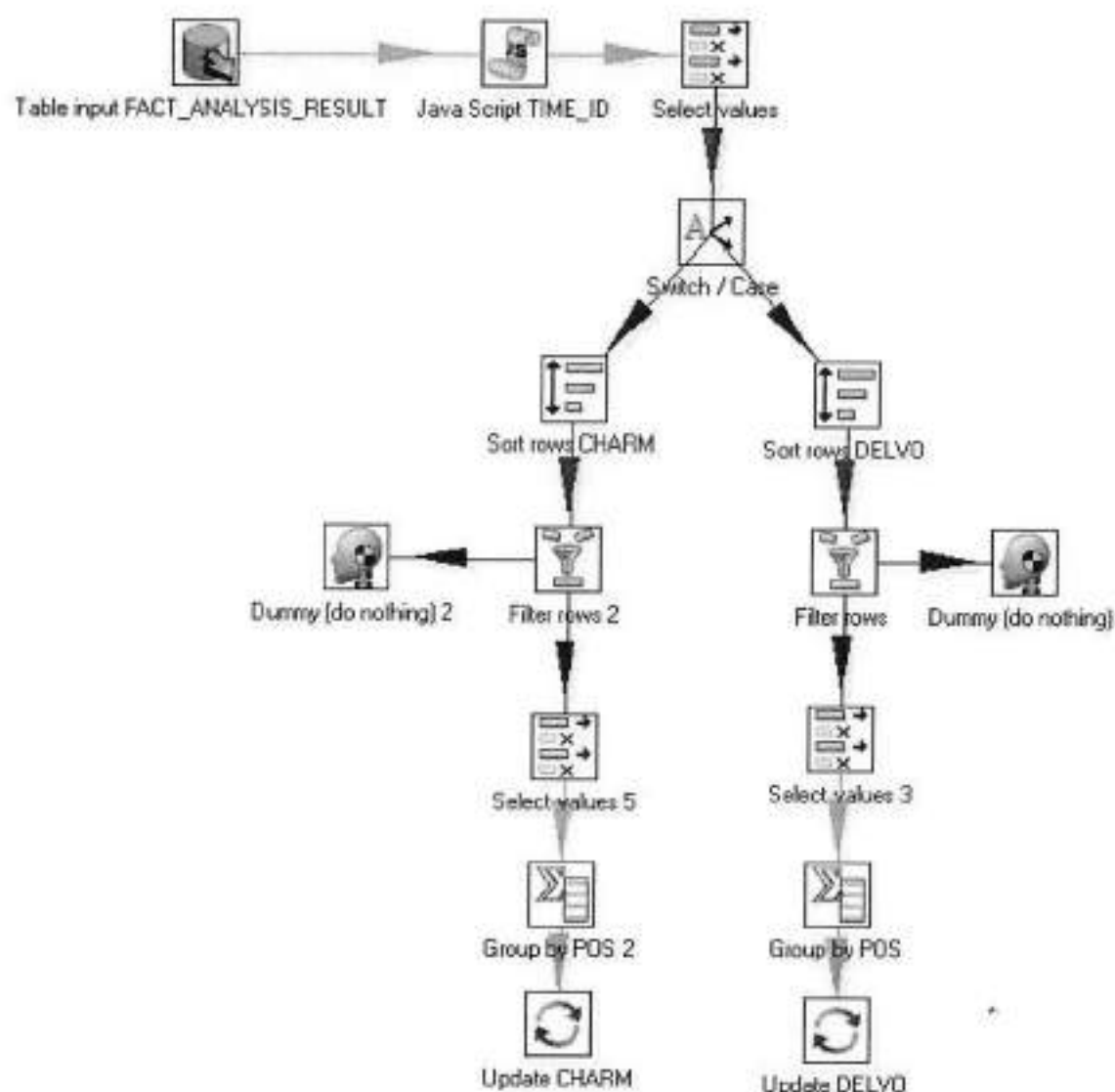


Figura 33 – Transformação número de análises DELVO e CHARM positivas

A única diferença desta transformação para a anterior é que nos passos "Filter rows" e "Filter rows 2" os registros com resultado de ATB negativo são descartados.

5.4.6.8 Transformação mostrar análise

Esta transformação (figura 34) é responsável por atualizar o campo "Mostrar Análise" que foi definido como "N" para todos os registros na primeira transformação

apresentada desta tabela fato. Este campo define se o registro pode ou não ser disponibilizado nos relatórios.

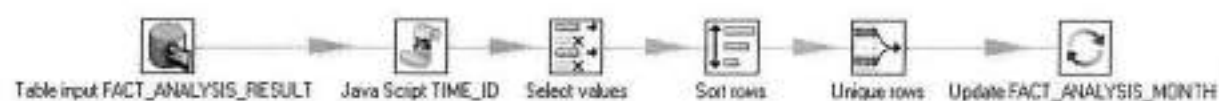


Figura 34 – Transformação mostrar análise

5.4.7 Carga incremental

As transformações apresentadas anteriormente são responsáveis pela carga completa de suas respectivas tabelas no DW. No entanto, estas transformações só são executadas uma vez, quando o banco de dados é criado, pois são extremamente demorada devido à enorme quantidade de registros. Após esta primeira carga, será realizada a carga incremental, que atualiza periodicamente o DW com os registros criados após a última carga. Assim, mantém-se o banco de dados atualizado sem a necessidade de longos períodos executando as transformações do ETL. Um exemplo de transformação deste tipo está apresentado na figura 35.



Figura 35 – Exemplo de transformação incremental

Esta é a transformação incremental que carrega a dimensão de clientes. Ela é idêntica a transformação completa (figura 14) com exceção do último passo "Insert / Update DIM_ACCOUNT" que, ao contrário do passo "Table output", compara as chaves do fluxo de dados com as dos registros da tabela dimensão para verificar se o registro já existe. Se o registro já existir na tabela, este é apenas atualizado, se não existir, é criado.

Outra diferença é que na *query* do passo "Table input b_org_ext" utiliza-se o campo onde fica armazenada a ultima vez que o registro foi alterado para extrair apenas os registros que foram alterados do dia anterior até o momento da extração, assim, apenas os registros novos são considerados, o que faz a transformação ser muito mais rápida.

5.4.8 Carga manual

Algumas vezes os registros referentes a uma OS especifica precisam ser atualizados imediatamente, sem a possibilidade de esperar até que a carga incremental ocorra automaticamente no fim do dia, por exemplo. Para estes casos é utilizada a carga manual. Um exemplo de transformação deste tipo é apresentado na figura 36.

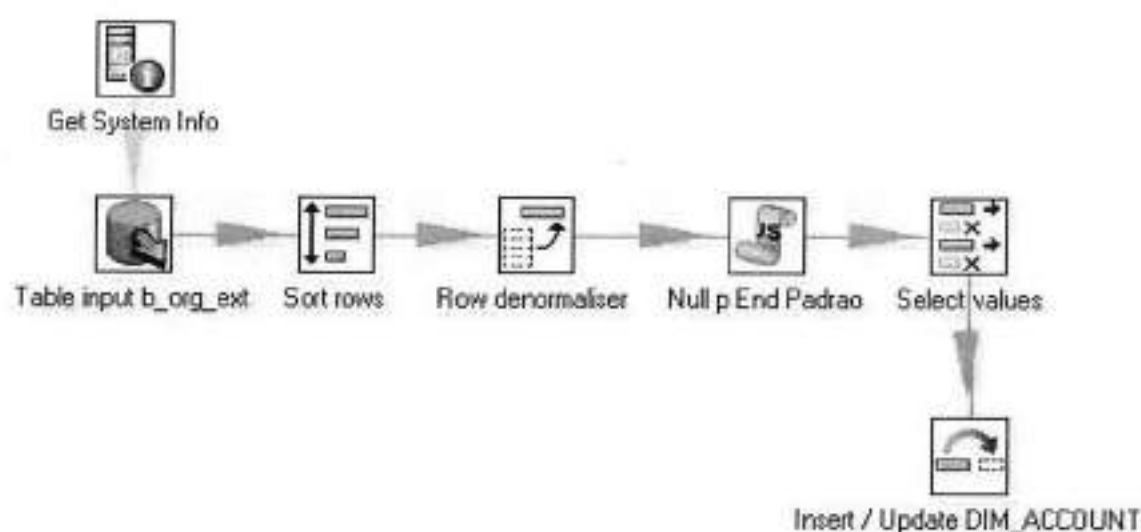


Figura 36 – Exemplo de transformação manual

Esta é a transformação manual que carrega a dimensão de clientes. Ela é idêntica a transformação incremental (figura 35) com exceção do primeiro passo "Get System Info" que recebe a informação do código da OS que deve ser atualizada.

5.5 Testes

Após a finalização da etapa de construção o DW está pronto para ser utilizado, no entanto, ainda é necessário a realização de testes para garantir que os resultados obtidos estejam de acordo com os esperados pelo usuário final.

Estes testes foram realizados por representantes do conjunto de usuários finais, que realizaram uma série de validações para avaliar a confiabilidade dos dados.

Para facilitar este processo foram criados scripts de teste com sugestões de testes que permitem avaliar de forma satisfatória os resultados obtidos através do sistema de BI. Estes scripts são conjuntos de passos a serem seguidos para se avaliar o funcionamento de cada parte do sistema de BI. Por exemplo, 1 - entrar na tela "Análise de Leite", 2 - criar uma nova análise, 3 - inserir um determinado indicador zootécnico na análise, 4 - verificar se a análise foi corretamente criada, etc.

Durante a avaliação dos resultados, os problemas encontrados foram corrigidos e re-testados até que o sistema apresentasse os resultados desejados.

5.6 Implementação

Após a validação do projeto pelo usuário final, os últimos detalhes do projeto devem ser definidos:

Periodicidade de execução do processo – Quando o projeto entrar em produção será realizada uma carga completa do DW, utilizando toda a informação do banco de dados operacional da Clínica do Leite. No entanto, esta carga leva muito tempo e só deve ser realizada uma vez. A partir desta carga inicial serão realizadas, com um determinado período, cargas incrementais para se realizar a atualização do DW. Por exemplo, a carga incremental pode ser executada todos os

dias a meia noite, com o intuito de atualizar o DW com as informações novas do dia anterior.

Responsáveis pela execução do processo – Muitos dos processos necessários ao sistema de BI ocorreram automaticamente de forma agendada, porém pode ser necessário a definição de alguns responsáveis por alguns processos manuais.

Treinamento de usuários chaves – Alguns usuários serão escolhidos para receber treinamento sobre o sistema e ferramentas utilizados. O intuito é que estes usuários principais transmitam posteriormente o treinamento para outros usuários interessados na utilização da ferramenta de BI.

Usuários da ferramenta de BI – Os primeiros usuários da ferramenta de BI são definidos, recebendo usuário e senha para acesso a ferramenta.

6 RESULTADOS

A etapa de construção resultou em 16 tabelas dimensão e 9 tabelas fato, totalizando 41 transformações e 4 *jobs* necessários para a construção e carga completa de todas as tabelas. Também foram criados o mesmo número de transformações e *jobs* para realizar a carga incremental. Além disso, foram criadas 16 transformações e 2 *jobs* para realizar a carga manual de algumas tabelas específicas.

Após a carga completa das tabelas fato e dimensões através das transformações para este tipo de carga, as transformações incrementais foram agendadas para serem executadas no fim de cada dia, o que garante que o DW será diariamente atualizado. Com isso, o DW já está pronto para ser utilizado.

Atualmente o projeto de Business Intelligence para a Clínica do Leite está em fase final de homologação em uma de suas frentes, onde relatórios baseados no DW foram desenvolvidos. No entanto, em outra frente, onde cubos de dados foram criados para realizar o cruzamento de informação a partir do DW, o projeto já está em produção e foi bem avaliado por funcionários da Clínica do Leite.

Na figura 37 é apresentado um exemplo dos relatórios desenvolvidos a partir do DW. O formato deste relatório foi definido no início do projeto e a construção da tabela fato de coeficientes zootécnicos foi realizada de forma a viabilizar a geração deste relatório de forma eficiente.



PAINEL DE CONTROLE E BENCHMARKING DA QUALIDADE DO LEITE

R600

Empresa: Su/Sudoeste de Minas
Localização (Meso-Região):
Mês Referência: Ago/08

Data de Emissão: 25/11/2009

Indicadores	Ago/08	Jul/08	Jun/08	Mai/08	Abr/08	Mar/08	Fev/08	Jan/08	Dez/07	Nov/07	Out/07	Sep/07	Média 3 Meses CLI MR IND	Média 12 Meses CLI MR IND				
Dados Gerais																		
# Fomecedores	408	420	0	0	0	0	0	0	0	0	0	0	222	177	448	148	122	420
# Coletas	7	8	7	11	8	6	6	6	6	6	6	6	1	1	7	1	1	7
Temperatura																		
mínima	4	3	0	3	1	4	2	3	5	2	1	3	5	6	2	6	6	2
máxima	6	8	5	6	10	7	6	4	9	5	8	5	7	7	7	6	6	7
média	5	4	3	5	5	5	4	4	6	4	4	4	6	6	4	7	7	4
Qualidade do Leite																		
CCS e CBT (padrão internacional)																		
# Fomecedores	115	89	95	84	81	42	55	50	50	76	82	48	40	40	99	36	25	88
% Fomecedores	26	21	22	20	16	12	13	12	14	21	21	18	23	21	23	22	21	18
Coleta de Amostra																		
% Possível Falha	0	0	0	0	0	0	0	0	0	0	0	0						
ATB - Delvo																		
# Positivos	1	5	10	3	8	4	12	9	13	11	8	4	4	4	6	3	4	7
% Positivos	0,5	1,4			1,9	1,2	3,3	2,6	4,3	3,8			1,9	1,7	1,0	4,9	3,4	2,4

Figura 37 – Relatório de Coeficientes Zootécnicos

7 DISCUSSÃO

Este projeto implementou de forma bem sucedida técnicas de tecnologia da informação (TI) para melhorar a forma como a informação, de um grande banco de dados com informações zootécnicas, é analisada.

Apesar do projeto não ter gerado melhorias nas técnicas de TI, apenas utilizou o que já existia no cenário atual, foi inovador no sentido de ter introduzido estas técnicas no setor de zootecnia. E pode ter aberto um precedente importante no sentido de incentivar a utilização destas técnicas, não apenas neste setor, mas em qualquer outro que necessite de formas eficientes de análise de grandes quantidades de dados.

8 CONCLUSÕES

O objetivo deste trabalho foi desenvolver um Data Warehouse (DW) que é parte de uma solução de Business Intelligence (BI) para melhorar a forma como a Clínica do Leite, uma instituição da Esalq/USP, analisa a grande quantidade de informações, basicamente sobre análise de leite e dados zootécnicos, acumulada ao longo de sua história. Analisar melhor toda esta informação era uma necessidade para se conhecer profundamente o cenário da qualidade do leite no estado, e poder vislumbrar pontos que podem ser melhorados neste cenário.

Business Intelligence (BI) é uma arquitetura composta por um conjunto de metodologias de gestão implementadas através de ferramentas de software, cuja função é proporcionar ganhos nos processos decisórios e analíticos, baseando-se na capacidade informacional das ferramentas que integram em um só lugar, um data warehouse (DW), todas as informações necessárias à tomada de decisão.

O processo que realiza a extração dos dados do banco operacional, a transformação destes dados (filtragem, limpeza, etc) e carga no novo banco informacional (DW) é chamado de ETL (Extract Transform Load).

O DW é um repositório de dados (banco de dados) especificamente desenhado para facilitar a geração de relatórios e a análise da informação. A informação em um DW deve estar em um formato consistente, livre de dados incoerentes e estruturado de forma a cumprir os requerimentos analíticos do usuário final.

Após a criação do DW, a informação já estará no formato ideal para ser utilizada e poderá ser acessada, dentre outras, por ferramentas de geração de relatórios e por ferramentas conhecidas como Online Analytical Processing (OLAP) ou Processo Analítico em Tempo Real, que geram cubos lógicos de dados para análise multidimensional.

Com esta ferramenta é possível responder de forma rápida e simples perguntas que no sistema antigo eram extremamente difíceis e mesmo inviáveis ao usuário final que não possua conhecimento técnico. Com a maior facilidade de acesso aos dados históricos da Clínica do Leite espera-se incentivar seu estudo, e desta forma contribuir para aumentar o conhecimento do cenário leiteiro do estado e até do país. Consequentemente, pesquisadores poderão desenvolver seus estudos com maior eficiência e qualidade, contribuindo para o direcionamento de melhorias neste cenário.

Atualmente uma frente do projeto de Business Intelligence para a Clínica do Leite já está em produção, onde cubos de dados foram criados para realizar o cruzamento de informação a partir do DW. Este frente foi bem avaliado por funcionários da Clínica do Leite.

9 REFERÊNCIAS

KIMBALL, R.; CASERTA, J. *The Data Warehouse ETL Toolkit*. Wiley. Indianapolis, Wiley Publishing, 2004, 526p.

KIMBALL, R.; ROSS, M. *Data Warehouse Toolkit*. Rio de Janeiro, Makron Books, 2000, 388p.

TAKIZAWA, M.; UENOYAMA, M.; YAMASAKI, F. Soluções de Business Intelligence baseadas em Software Livre. 2005. 124p. Projeto de Formatura – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

ECKERSON, W. *The Evolution of ETL*. Journal of Data Warehousing, p. 5, Fall 2003.

ON, P. *The Importance of Enterprise Information Management for Business Intelligence*. Business Intelligence Journal, p. 6, First Quarter 2006.

WILLIAMS, S.; WILLIAMS, N. *The Business Value of Business Intelligence*. p. 10, Fall 2003.

GRAHAM, P. *Linking Information Architecture to Enterprise Strategy*. Business Intelligence Journal, p. 5, Fourth Quarter 2006.

ROSENBERG, A. *Improving Query Performance in Data Warehouses*. Business Intelligence Journal, p. 6, First Quarter 2006.

DW!. Página com informações e notícias sobre business intelligence. Disponível em: <<http://www.datawarehouse.inf.br/>>. Acesso em: 15 nov. 2009.

DATA_WAREHOUSE. Página sobre Data Warehouse. Disponível em: <http://en.wikipedia.org/wiki/Data_warehouse>. Acesso em: 4 set. 2009.

STAR_SCHEMA. Página sobre Star Schema. Disponível em:

< http://en.wikipedia.org/wiki/Star_schema >. Acesso em: 4 set. 2009.

SNOWFLAKE_SCHEMA. Página sobre Snowflake Schema. Disponível em:

< http://en.wikipedia.org/wiki/Snowflake_schema >. Acesso em: 4 set. 2009.

10 APÊNDICE

Exemplo de transformações com mais detalhes:

Dimensão de clientes

Dimensão que contem os dados referentes aos clientes da Clínica do Leite.

Origem dos dados

Tabela principal: b_org_ext.

Join's com: b_user, b_pri_lst, b_org_addr e b_gov_ident.

Filtros: Tanto na tabela principal quanto nas join's só são utilizados registros não excluídos (DELETED_BY = "") e apenas tipo de conta igual a 'Cliente' através do campo TYPE da tabela principal.

Destino dos dados

Tabela: DIM_ACCOUNT

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
ACCNT_ID	ID do Cliente utilizado pelo negócio.
NAME	Nome do Cliente.
CODE	Código do Cliente utilizado pelo negócio.
ACCOUNT_TYPE	Tipo da conta, Fazenda ou Empresa.
SEGMENTATION	Segmentação. Utilizado para classificar os clientes.
GROUP_TYPE	Grupo do cliente.
MANAGER_NAME	Gerente responsável.
PRI_LST_NAME	Lista de preços.
STATUS	Status.
ADDR_PRINCIPAL*	Endereço padrão.
COMPLEMENT_PRINCIPAL*	Complemento do endereço padrão.
CITY_PRINCIPAL*	Cidade.
STATE_PRINCIPAL*	Estado.
MESO_PRINCIPAL*	Mesorregião.
MESO_CODE_PRINCIPAL*	Código da mesorregião.
MICRO_PRINCIPAL*	Microrregião.
MICRO_CODE_PRINCIPAL*	Código da microrregião.
LATITUDE_PRINCIPAL*	Latitude.
LONGITUDE_PRINCIPAL*	Longitude.

* Na tabela existem também estes campos para endereço de cobrança, envio de análise, recebimento de material, envio de nota fiscal e ponto de coleta.

Transformação



Passos da transformação

Table input b_org_ext: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_org_ext*, inclusive os referentes a endereço que estão normalizados, ou seja, existe apenas um campo para endereço, cidade, estado, etc, em cada registro, e mais um campo para se definir o tipo de endereço (padrão, cobrança, envio de material, etc.).

Sort rows: Neste passo será realizada a ordenação do campo *ACCNT_ID*, pois é necessário para o próximo passo.

Row denormaliser: Neste passo será realizado o processo de desnormalização dos campos de endereço. Até este ponto temos para cada cliente (*ACCNT_ID*) seis registros, cada um com seus campos de endereço relacionados a objetivos diferentes (endereço padrão, cobrança, envio de material, etc.). Após este ponto iremos ter apenas um registro, porém com um maior numero de campos, ou seja, teremos no mesmo registro um campo para o endereço padrão, um para o de cobrança, um para o de envio de material, etc.

Null p End Padrão: Neste passo iremos preencher os campos de endereço que estiverem vazios com os de endereço padrão. Assim, se o endereço de envio de material de um determinado Cliente não esta preenchido vamos assumir que o material deve ser enviado para seu endereço padrão.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_ACCOUNT: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de centros de custo

Dimensão que contem os dados dos diversos centros de custo utilizados pelo negócio.

Origem dos dados

Tabela principal: *b_cost_center*.

Join's com: nenhuma.

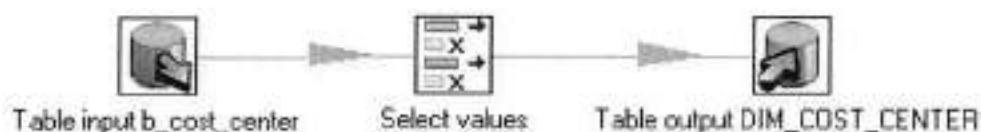
Filtros: São considerados apenas registros não excluídos (*DELETED_BY* = "").

Destino dos dados

Tabela: DIM_COST_CENTER

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
COST_CENTER_ID	ID do centro de custo utilizado pelo negócio.
NAME	Nome do centro de custo.
CODE	Código do centro de custo.

Transformação



Passos da transformação

Table input b_cost_center: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_cost_center*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_COST_CENTER: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de lançamentos financeiros

Dimensão que contem os dados de lançamentos financeiros.

Origem dos dados

Tabela principal: b_finance.

Join's com: nenhuma.

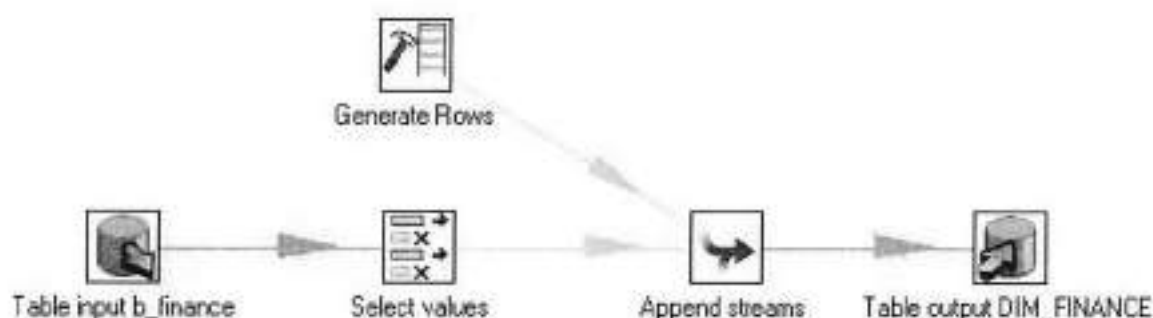
Filtros: São considerados apenas registros não excluídos (DELETED_BY = "").

Destino dos dados

Tabela: DIM_FINANCE

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
FINANCE_ID	ID do lançamento utilizado pelo negócio.
INVOICE_ID	ID do boleto que contem o lançamento.
CODE	Código do lançamento.

Transformação



Passos da transformação

Table input b_finance: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_finance*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Generate Rows: Neste passo é gerado um registro com o *FINANCE_ID* em branco. Este artifício é necessário, pois a tabela Fato que utiliza esta dimensão necessita armazenar os registros que não tem lançamento financeiro associado. E, como explicado anteriormente, registros sem alguma das chaves associadas as dimensões não são inseridos na tabela Fato. Então, quando a tabela Fato for carregada, os registros sem lançamento associado receberão o ID deste registro.

Append streams: Este passo apenas junta os registros vindos dos passos anteriores. É possível definir qual bloco de registros irá entrar primeiro na "stream" (fluxo) final de dados.

Table output DIM_FINANCE: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de pagadores

Dimensão que contém os dados referentes aos clientes que efetivamente efetuam pagamentos.

Origem dos dados

Tabela principal: *b_org_ext*.

Join's com: *b_org_ext*, *b_pri_lst* e *b_pay_cond*. É necessário realizar uma "join" com a própria *b_org_ext*, porque um dos campos da tabela destino é referente a fundação a qual o cliente está ligado e esta fundação também é armazenada como cliente na mesma tabela.

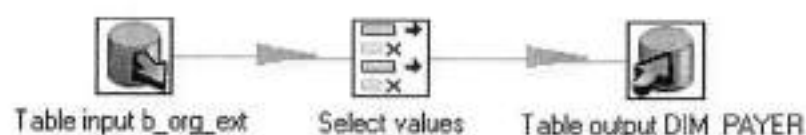
Filtros: Tanto na tabela principal quanto nas join's só são utilizados registros não excluídos (*DELETED_BY* = '') e apenas tipo de conta igual a 'Cliente' através do campo *TYPE* da tabela principal.

Destino dos dados

Tabela: DIM_PAYER

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
PAYER_ID	ID do Cliente (Pagador) utilizado pelo negócio.
NAME	Nome do Pagador.
FOUNDATION	Fundação ligada ao cliente.
PRICE_LIST	Lista de preços.
PAYMENT_TERM	Período de pagamento (15 Dias, 28 Dias, etc.)

Transformação



Passos da transformação

Table input b_org_ext: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_org_ext*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_PAYER: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de posições

Dimensão que contém os dados referentes às posições para utilização do software de CRM da Clínica do Leite. Estas posições definem, por exemplo, quais telas um determinado usuário poderá acessar.

Origem dos dados

Tabela principal: b_postn.

Join's com: b_bu.

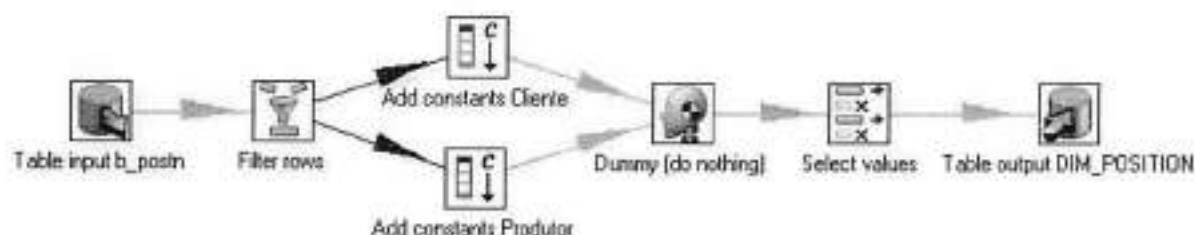
Filtros: Tanto na tabela principal quanto na 'join' com a b_bu só são utilizados registros não excluídos (DELETED_BY = "").

Destino dos dados

Tabela: DIM_POSITION

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
POSITION_ID	ID da Posição utilizado pelo negócio.
POSITION_NAME	Nome da Posição.
ORGANIZATION	Fundação ligada ao cliente.
TYPE	Tipo da Posição.

Transformação



Passos da transformação

Table input b_postn: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_postn*.

Filter rows: Neste passo ocorre a separação entre os registros no *PAR_ID* nulo ou não. Se o registro tem *PAR_ID* nulo, a posição é do tipo Cliente, do contrario, é do tipo Produtor.

Add Constants Cliente e Add Constants Produtor: Estes passos adicionam um o campo *TYPE* aos registros com valor Cliente ou Produtor dependendo do direcionamento realizado no passo anterior.

Dummy: Este passo simplesmente junta os registros vindos dos passos anteriores. Para isso é necessário que os dois fluxos de dados tenham as mesmas colunas, com os mesmos nomes e na mesma ordem.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_POSITION: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de produtos

Dimensão que contem os dados referentes aos produtos oferecidos pela Clinica do Leite. Principalmente análises laboratoriais e material de coleta, entre outros.

Origem dos dados

Tabela principal: b_product.

Join's com: b_product e b_report_patterns (LEFT OUTER JOIN). É necessário realizar uma "join" com a própria b_product, porque um dos campos da tabela destino é referente ao tipo (TYPE) do produto e este tipo é armazenado na mesma tabela origem (b_product) também como um produto.

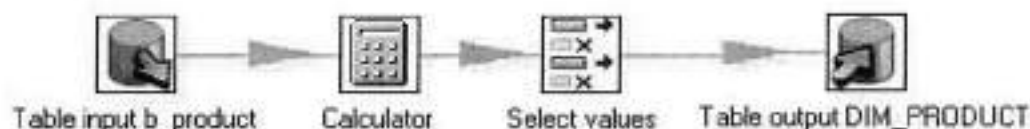
Filtros: Tanto na tabela principal quanto nas join's só são utilizados registros não excluídos (DELETED_BY = ""). Na b_report_patterns só são considerados registros com TYPE igual a 'PRODUCTS_UM', que são registros que armazenam a unidade do produto, e na tabela principal só são considerados registros com TYPE igual a 'Produto'.

Destino dos dados

Tabela: DIM_PRODUCT

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
PRODUCT_ID	ID do Produto utilizado pelo negócio.
NAME	Nome do Produto.
DESCRIPTION	Descrição do Produto. Consiste em uma concatenação do nome do produto, sua descrição na tabela de origem e sua unidade.
TYPE	Tipo do Produto. Em geral "Análise Laboratorial" ou "Material de Coleta".

Transformação



Passos da transformação

Table input b_product: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela b_product.

Calculator: Neste passo é onde se concatena o nome do produto, sua descrição na tabela de origem e sua unidade para formar o novo campo DESCRIPTION da tabela destino.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_PRODUCT: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Ordens de Solicitação de Material

Dimensão que contém os dados referentes a ordens de solicitação de material.

Origem dos dados

Tabela principal: b_order.

Join's com: nenhuma.

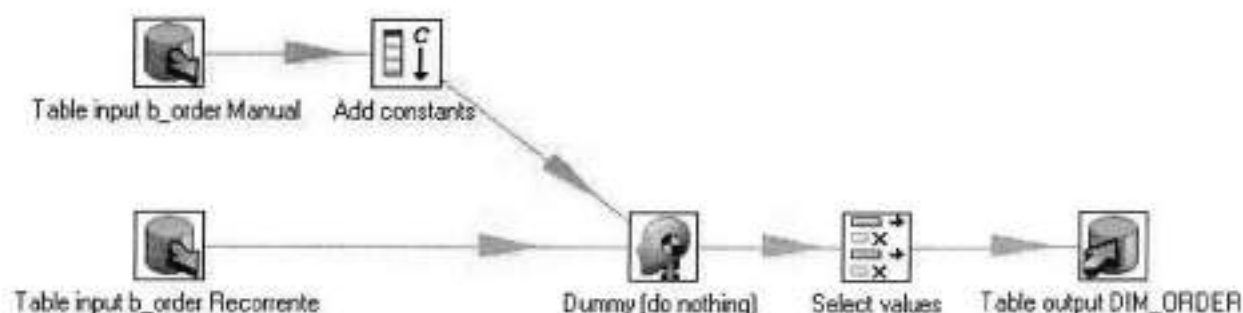
Filtros: Foi necessário usar dois passos para a entrada dos dados, um para filtrar apenas registros com tipo igual a "Manual" e status igual a "Enviado", e outro para filtrar apenas registros com tipo igual a "Recorrente". Ambos os passos tem também o filtro que desconsidera registros excluídos.

Destino dos dados

Tabela: DIM_ORDER

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
ORDER_ID	ID da Order de Solicitação de Material utilizado pelo negócio.
CODE	Código da Solicitação de Material utilizado pelo negócio.
DELIVERY_MODE	Informa como o material requerido será enviado. Ex: Correios – Sedex.
TYPE	Define se a solicitação de material é manual ou recorrente.
RECURRENCE_PERIOD	Define o período de recorrência quando for o caso. Para solicitações manuais é <i>NULL</i> .

Transformação



Passos da transformação

Table input b_order Manual: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_order*, porém apenas registros do tipo "Manual".

Table input b_order Recorrente: Neste passo é onde se encontra a query que puxa todos os campos relevantes da tabela *b_order* , porém apenas registros do tipo "Recorrente".

Add constants: Neste passo é incluído ao fluxo de dados a coluna *RECURRENCE_PERIOD* com valor *NULL* para todos os registros, já que estes registros são manuais, e portanto, sem nenhuma recorrência.

Dummy: Este passo simplesmente junta os registros vindos dos passos anteriores.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_ORDER: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Boletos

Dimensão que contem os dados referentes aos boletos emitidos pela Clinica do Leite.

Origem dos dados

Tabela principal: *b_invoice*.

Join's com: nenhuma.

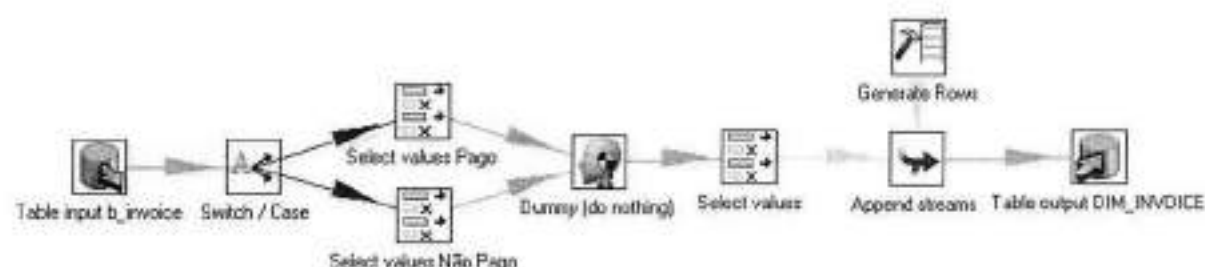
Filtros: Só são utilizados registros não excluídos (*DELETED_BY* = "").

Destino dos dados

Tabela: *DIM_INVOICE*

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
INVOICE_ID	ID do Boleto utilizado pelo negócio.
CODE	Código do Boleto utilizado pelo negócio.
VALUE	
STATUS	Informa o status do boleto. Ex: Pago ou Atrasado.
CHARGED_DATE	
MAIL_DATE	
DUE_DATE	Data de vencimento do boleto.
PAYMENT_DATE	Data em que o pagamento foi realizado.
VALUE_PAID	Valor efetivamente pago.
FISCAL_CODE	Código fiscal do boleto.

Transformação



Passos da transformação

Table input b_invoice: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_invoice*.

Switch / Case: Neste passo existe uma divisão no fluxo de dados, registros com status "Pago" ou "Cancelado" são direcionados para o passo "Select values Pago" e os demais para o passo "Select values Não Pago". Isto é necessário, porque alguns campos são gerados na query do primeiro passo de forma duplicada, são gerados os mesmos campos considerando o registro com status "Pago" ou "Cancelado" e considerando outros status quaisquer, então dependendo para onde o registro é direcionado, campos diferentes serão utilizados no passo posterior.

Select values Pago: Neste passo aos registros com status "Pago" e "Cancelado" vão ser atribuídos os campos calculados pela query do primeiro passo considerando um registro pago.

Select values Não Pago: Neste passo aos registros com status diferente de "Pago" e "Cancelado" vão ser atribuídos os campos calculados pela query do primeiro passo considerando um registro não pago.

Dummy: Este passo simplesmente reúne os registros vindos dos passos anteriores.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Generate Rows: Neste passo é gerado um registro com o INVOICE_ID em branco. Este artifício é necessário, pois a tabela Fato que utiliza esta dimensão necessita armazenar os registros que não tem boleto associado. E, como explicado anteriormente, registros sem alguma das chaves associadas as dimensões não são inseridos na tabela Fato. Então, quando a tabela Fato for carregada, os registros sem boleto associado receberão o ID deste registro.

Append streams: Este passo apenas junta os registros vindos dos passos anteriores. É possível definir qual bloco de registros irá entrar primeiro na "stream" (fluxo) final de dados.

Table output DIM_INVOICE: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Ordens de Serviço

Dimensão que contém os dados referentes a ordens de serviço recebidas pela Clínica do Leite.

Origem dos dados

Tabela principal: b_serv_order.

Join's com: b_org_ext.

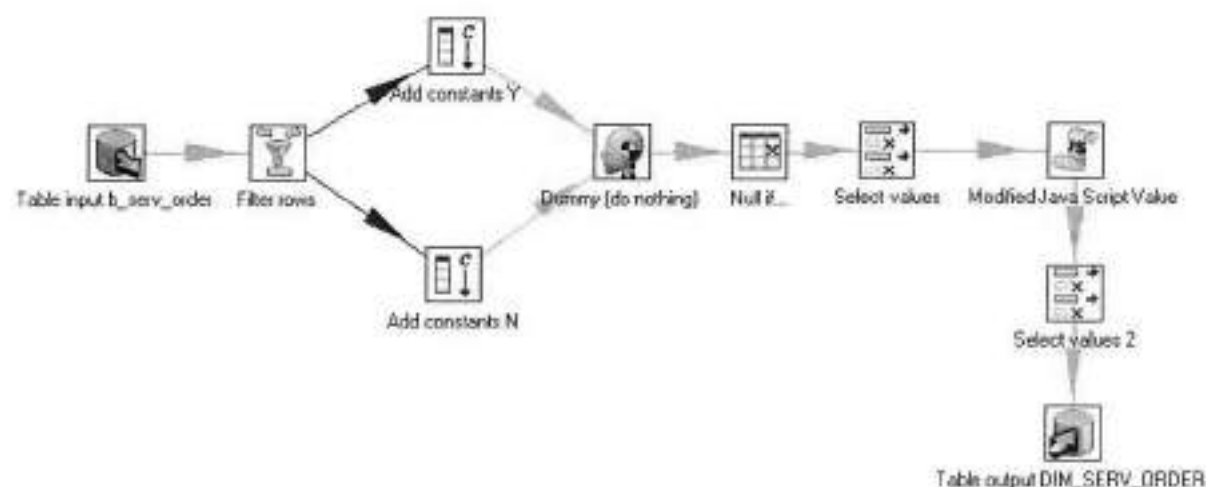
Filtros: Tanto na tabela principal quanto na join só são utilizados registros não excluídos (DELETED_BY = "") e só são considerados registros com número de amostras maior que zero para o caso de ordens de serviço do tipo 'Analysis'.

Destino dos dados

Tabela: DIM_SERV_ORDER

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
SERV_ORDER_ID	ID da OS utilizado pelo negócio.
CODE	Código da OS utilizado pelo negócio.
TEMPERATURE	Temperatura das amostras da OS.
ANALYSIS_LINE	Linha de análise. Ex: CBT - Bentley
LAB_EXPRESS_FLG	
ANALYSIS_DATE	Data da análise.
SAMPLE_NUMBER	Número de amostras.
COLLECTION_DATE	Data em que as amostras foram coletadas.
RECEIVE_DATE	Data em que as amostras chegaram a Clínica do Leite.
TYPE	Tipo da OS. Ex: Analysis, Generic, etc.
CREATED	Data de criação da OS.
CLINICALOG	
PROCESSING_DATE	
REVISION_DATE	Data em que a OS foi validada.
COMMENTS	Comentários.

Transformação



Passos da transformação

Table input b_serv_order: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_serv_order*.

Filter rows, Add constants Y e Add constants N: Estes passos em conjunto definem se o registro tem o campo CLINICALOG com valor "Y" ou "N". Este campo indica se a OS foi criada diretamente ou através de um ponto de coleta.

Dummy: Este passo simplesmente reúne os registros vindos dos passos anteriores.

Null IF...: Este passo substitui valores incoerentes de temperatura por *NULL*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino e o campo *TEMPERATURE* foi definido com inteiro.

Modified Java Script Value: Este passo contém um script que tem como função substituir valores de temperatura menores que 0 ou maiores que 30 por *NULL*.

Select values 2: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_SERV_ORDER: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Unidades de Análise

Dimensão que contém os dados referentes as Unidades de Análise (animais, tanques de cooperativas, caminhões com leite de vários produtores, etc.) cadastradas pela Clínica do Leite.

Origem dos dados

Tabela principal: b_ua.

Join's com: b_gov_prod (LEFT OUTER JOIN).

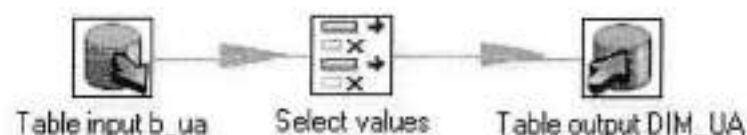
Filtros: Tanto na tabela principal quanto na join só são utilizados registros não excluídos (DELETED_BY = "") e só são considerados registros da b_ua que tenham os campos CODE e ROWID diferentes de NULL e vazio.

Destino dos dados

Tabela: DIM_UA

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
UA_ID	ID da UA utilizado pelo negócio.
CODE	Código da UA utilizado pelo negócio.
NAME	Nome da UA.
LINE	Rota utilizada para coleta de amostras desta UA. (Verificar)
TYPE	Tipo da UA. Ex: Animal, Silo, Tanque, etc.
NRP	Código de uma UA cadastrado no Ministério Agricultura. (Verificar)

Transformação



Passos da transformação

Table input b_ua: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_ua*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_UA: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Tabela Fato de Resultados de Análises

Tabela Fato que contem os resultados das análises realizadas pela Clínica do Leite. Esta tabela Fato tem como dimensões a DIM_PRODUCT, DIM_UA, DIM_SERV_ORDER e DIM_ACCOUNT. Ou seja, cada registro desta tabela armazena um resultado de análise associado as características produto (CCS, CBT, etc.), Unidade de Análise, Ordem de Serviço e Cliente/Produtor.

Origem dos dados

Tabela principal: b_analysis.

Join's com: b_serv_order_item, b_serv_order e b_product.

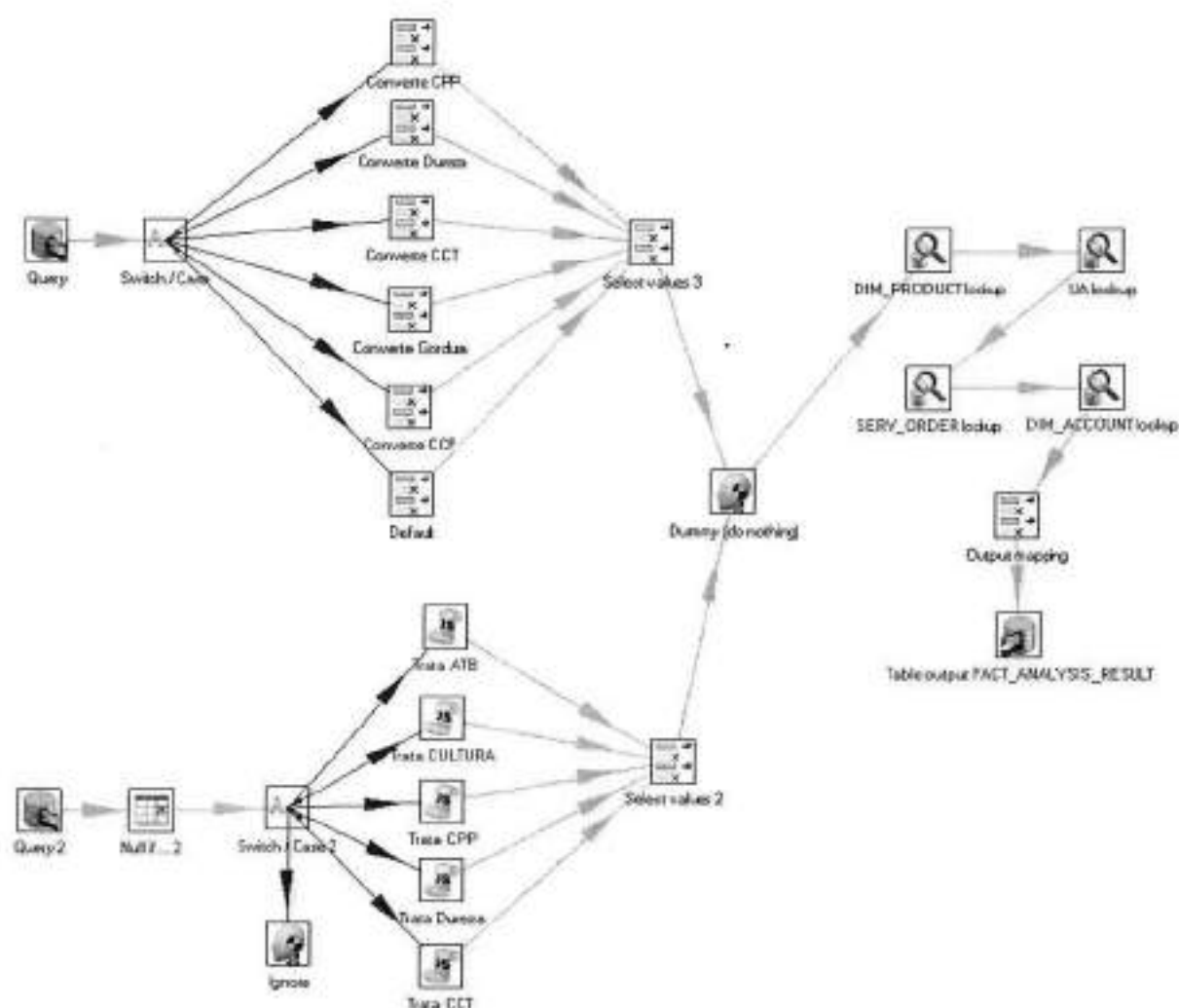
Filtros: Tanto na tabela principal quanto nas join's só são utilizados registros não excluídos (DELETED_BY = "") e só são considerados registros da b_serv_order_item com UA_ID não nulo e não vazio.

Destino dos dados

Tabela: FACT_ANALYSIS_RESULT

Nome do campo	Descrição
ROWID	Código inteiro e incremental utilizado apenas para controle da tabela.
LAST_UPD	Data em que o registro foi inserido ou atualizado pela ultima vez.
SERV_ORDER_ID	Código inteiro e incremental que relaciona o registro a sua OS na dimensão DIM_SERV_ORDER.
UA_ID	Código inteiro e incremental que relaciona o registro a sua UA na dimensão D_UA.
PRODUCT_ID	Código inteiro e incremental que relaciona o registro ao seu Produto na dimensão DIM_PRODUCT.
ACCNT_ID	Código inteiro e incremental que relaciona o registro ao Cliente/Produtor na dimensão DIM_ACCOUNT.
SHOW_ANALYSIS	Flag que permite ou não acesso a informação do registro por alguns relatórios.
_SERV_ORDER_ITEM_CODE	Código do item de OS utilizado pelo negócio.
RESULT	Resultado da análise.
UA_NAME	Nome da UA.
UA_LINE	Rota utilizada para coleta de amostras desta UA.
NRP	Código de uma UA cadastrado no Ministério Agricultura.

Transformação



Passos da transformação

Query: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_analysis*. Este passo só considera resultados do tipo numérico.

Query 2: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_analysis*. Este passo só considera resultados do tipo não numérico. Ex: POS ou NEG.

Switch / Case: Este passo separa os registros de acordo com o nome dos produtos analisados (CPP, Dureza, CCT, GOR, CCF e demais), pois o resultado destas análises serão tratados de forma distintas.

Converte CPP, Converte Dureza, Converte CCT, Converte Gordura, Converte CCF e Default: Cada um desses passos trata de forma diferente os resultados dos

respectivos produtos. Por exemplo, os resultados de Gordura são definidos como do tipo Number com 6 dígitos e duas casas decimais de precisão.

Null if... 2: Este passo substitui valores incoerentes ("Incontável" ou "Ausente", por exemplo) de resultados de análise por *NULL*.

Switch / Case 2: Este passo separa os registros de acordo com o nome dos produtos analisados (ATB – Delvo, ATB – CHARM, CULTURA, CPP, Dureza, CCT e demais), pois o resultado destas análises serão tratados de forma distintas. Os demais produtos são ignorados.

Trata ATB, Trata CULTURA, Trata CPP, Trata Dureza e Trata CCT: Cada um desses passos trata de forma diferente os resultados dos respectivos produtos. Por exemplo, para ATB caso o resultado seja "POS" é alterado para "1" e caso seja "NEG" é alterado para "-1".

Dummy: Este passo simplesmente reúne os registros vindos dos passos anteriores.

DIM_PRODUCT lookup, UA lookup, SERV_ORDER lookup e DIM_ACCOUNT lookup: Estes passos realizam a troca dos códigos utilizados pelo negocio respectivamente para Produtos, Unidades de Análise, Ordens de Serviços e Clientes/Produtores, pelos códigos inteiros e incrementais definidos nas dimensões. Ou seja, estes passos criam a ligação entre a fato e as dimensões.

Output mapping: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output FACT_ANALYSIS_RESULT: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Lista de Preços

Dimensão que contem os dados referentes aos valores cobrados pela Clínica para cada produto ou material.

Origem dos dados

Tabela principal: b_pri_lst_item.

Join's com: b_pri_lst.

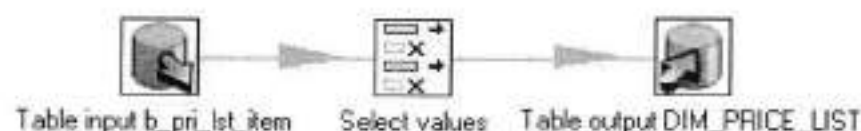
Filtros: Tanto na tabela principal quanto na join só são utilizados registros não excluídos (DELETED_BY = "").

Destino dos dados

Tabela: DIM_PRICE_LIST

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
PRICE_LIST_ITEM_ID	ID do item da lista de preços utilizado pelo negócio.
CODE	Código do item da lista de preços utilizado pelo negócio.
NAME	Nome do item da lista de preços.
PRICE	Valor do item.
TYPE	Tipo do item. Ex: Análise ou Material de Coleta.
QUANTITY_FROM	Quantidade mínima disponível do item.
QUANTITY_UNTIL	Quantidade máxima disponível do item.
PARENT_CODE	LS-000001
PARENT_NAME	Lista padrão
PARENT_DESCRIPTION	Lista menor preço

Transformação



Passos da transformação

Table input b_pri_lst_item: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_pri_lst_item*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM PRICE_LIST: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Usuários

Dimensão que contém os dados referentes aos usuários do sistema utilizado pela Clínica do Leite.

Origem dos dados

Tabela principal: b_user.

Join's com: nenhuma.

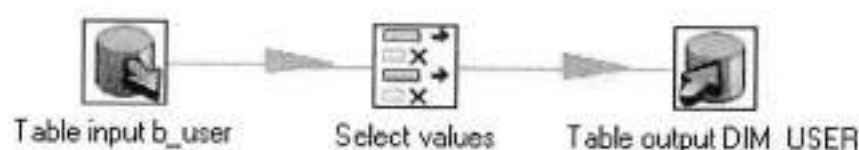
Filtros: Só são utilizados registros não excluídos (DELETED_BY = "").

Destino dos dados

Tabela: DIM_USER

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
USER_ID	ID do usuário utilizado pelo negócio.
USER_NAME	Nome do usuário.
TYPE	Tipo do usuário. Ex: Cliente ou Funcionário.

Transformação



Passos da transformação

Table input b_user: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_user*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_USER: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Dimensão de Telas

Dimensão que contém os dados referentes as telas do sistema utilizado pela Clínica do Leite.

Origem dos dados

Tabela principal: b_view.

Join's com: t_screen.

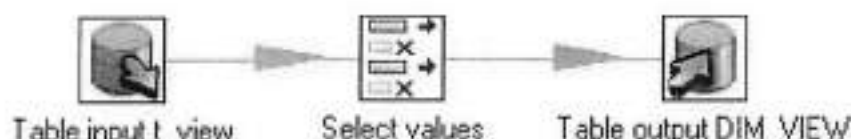
Filtros: Tanto na tabela principal quanto na join só são utilizados registros não excluídos (DELETED_BY = "").

Destino dos dados

Tabela: DIM_VIEW

Nome do campo	Descrição
ID	Surrogate Key, gera a identidade relacional.
VIEW_ID	ID da tela utilizado pelo negócio.
NAME	Nome da tela.
SCREEN_NAME	Nome do módulo ao qual a tela esta associada.

Transformação



Passos da transformação

Table input b_view: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_view*.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output DIM_VIEW: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Tabela Fato de Resultados de Análises consolidados por mês e por unidades de análise

Tabela Fato que contem os resultados das análises realizadas pela Clínica do Leite consolidados por mês. Esta tabela Fato tem como dimensões a DIM_PRODUCT, DIM_UA, DIM_TIME e DIM_ACCOUNT. Ou seja, cada registro desta tabela armazena o resultado da média de análises associada as características produto (CCS, CBT, etc.), Unidade de Análise, Cliente/Produtor e Mês.

O processo para criação da tabela foi dividido em duas transformações, pois a tentativa de realizar todo o processo em uma única transformação resultou em baixa performance.

Origem dos dados

Tabela principal: *b_analysis*.

Join's com: *b_serv_order_item*, *b_serv_order* e *b_product*.

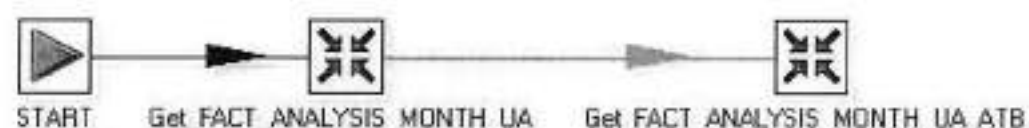
Filtros: Tanto na tabela principal quanto nas join's só são utilizados registros não excluídos (*DELETED_BY* = ""), só são considerados registros da *b_serv_order_item* com *UA_ID* não nulo e não vazio e *UA_TYPE* igual a 'Tanque', e só são considerados registros da *b_serv_order* com o campo *TEMPERATURE* numérico.

Destino dos dados

Tabela: FACT_ANALYSIS_MONTH_UA

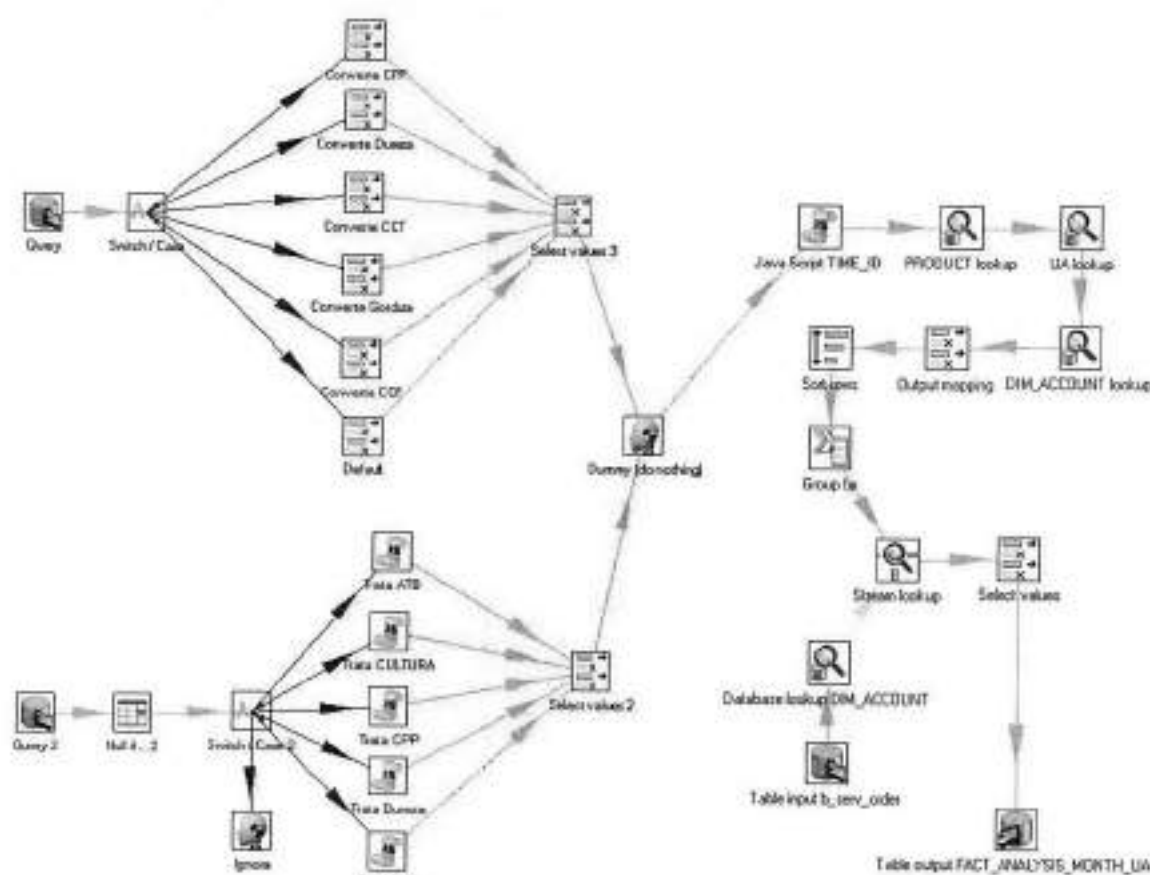
Nome do campo	Descrição
ACCOUNT_ID	Código inteiro e incremental que relaciona o registro ao Cliente/Produtor na dimensão DIM_ACCOUNT.
TIME_ID	Código inteiro e incremental que relaciona o registro a uma data na dimensão DIM_TIME. No caso desta tabela, como os dados são consolidados por mês, os IDs foram definidos como sempre referentes ao primeiro dia de cada mês.
PRODUCT_ID	Código inteiro e incremental que relaciona o registro ao seu Produto na dimensão DIM_PRODUCT.
UA_ID	Código inteiro e incremental que relaciona o registro a sua UA na dimensão D_UA.
AVERAGE	Média dos resultados consolidados por mês.
AVG_TEMP	Média de temperatura consolidadas por mês.
MAX_TEMP	Temperatura máxima.
MIN_TEMP	Temperatura mínima.
COLLECT_NUM	Numero de datas de recebimentos distintas consolidadas por mês.
SHOW_ANALYSIS	Flag que permite ou não acesso a informação do registro por alguns relatórios.

Job



Este é o Job que realiza o processo de carga da tabela Fato Analysis Month UA. Os passos `Get_FACT_ANALYSIS_MONTH_UA` e `Get_FACT_ANALYSIS_MONTH_UA_ATB` simplesmente executam as respectivas transformações na ordem apresentada na figura.

Transformação 1 – Get_FACT_ANALYSIS_MONTH_UA



Passos da transformação

Query: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_analysis*. Este passo só considera resultados do tipo numérico.

Query 2: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_analysis*. Este passo só considera resultados do tipo não numérico. Ex: POS ou NEG.

Switch / Case: Este passo separa os registros de acordo com o nome dos produtos analisados (CPP, Dureza, CCT, GOR, CCF e demais), pois o resultado destas análises serão tratados de forma distintas.

Converte CPP, Converte Dureza, Converte CCT, Converte Gordura, Converte CCF e Default: Cada um desses passos trata de forma diferente os resultados dos respectivos produtos. Por exemplo, os resultados de Gordura são definidos como do tipo Number com 6 dígitos e duas casas decimais de precisão.

Null if... 2: Este passo substitui valores incoerentes ("Incontável" ou "Ausente", por exemplo) de resultados de análise por NULL.

Switch / Case 2: Este passo separa os registros de acordo com o nome dos produtos analisados (ATB – Delvo, ATB – CHARM, CULTURA, CPP, Dureza, CCT e demais), pois o resultado destas análises serão tratados de forma distintas. Os demais produtos são ignorados.

Trata ATB, Trata CULTURA, Trata CPP, Trata Dureza e Trata CCT: Cada um desses passos trata de forma diferente os resultados dos respectivos produtos. Por exemplo, para ATB caso o resultado seja "POS" é alterado para "1" e caso seja "NEG" é alterado para "-1".

Dummy: Este passo simplesmente reúne os registros vindos dos passos anteriores.

Java Script TIME_ID: Este passo calcula através do campo CREATED da b_analysis o TIME_ID referente ao primeiro dia do mês desta data. Neste caso não é necessário fazer o lookup na dimensão TIME_ID, pois este ID pode ser calculado, o que é mais rápido.

PRODUCT lookup, UA lookup, e DIM_ACCOUNT lookup: Estes passos realizam a troca dos códigos utilizados pelo negocio respectivamente para Produtos, Unidades de Análise e Clientes/Produtores, pelos códigos inteiros e incrementais definidos nas dimensões. Ou seja, estes passos criam a ligação entre a fato e as dimensões.

Output mapping: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Sort rows: Este passo ordena os registros por ACCOUNT_ID, UA_ID, PRODUCT_ID, TIME_ID e RESULT. É um pré-requisito para o próximo passo.

Group by: Este passo agrupa os registros pelos campos ACCOUNT_ID, UA_ID, PRODUCT_ID e TIME_ID. E realiza as seguintes operações com os demais campos: Média do campo RESULT para criação do campo AVERAGE, média do campo TEMPERATURE para criação do campo AVG_TEMP, valor máximo do campo TEMPERATURE para criação do campo MAX_TEMP, valor mínimo do campo TEMPERATURE para criação do campo MIN_TEMP, valor máximo do campo SHOW_ANALYSIS para o mesmo campo com o objetivo de inserir o valor "Y" se existir algum registro com este valor. Para o campo LOOKUP_REFERENCE também utiliza-se a função valor máximo, mas apenas para o campo continuar no fluxo de dados, pois este campo é sempre igual para um mesmo mês.

Table input b_serv_order: Este passo é utilizado para calcular por query o número de datas de recebimento por mês e por Cliente/Produtor a partir da tabela b_serv_order_item considerando apenas UAs do tipo 'Tanque'. E também gera o campo LOOKUP_REFERENCE que será utilizado em um passo posterior para levar esta informação ao fluxo principal de dados.

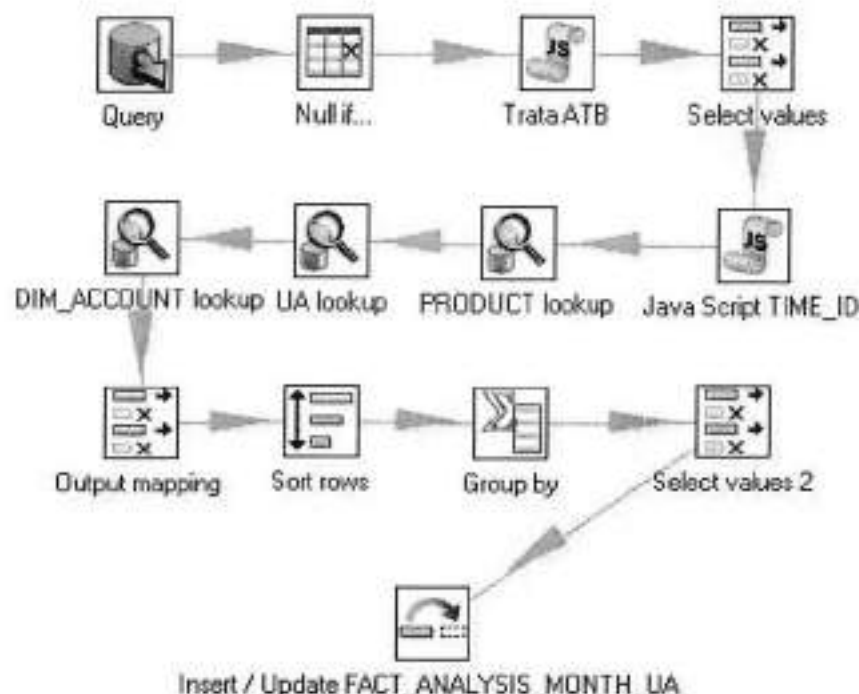
Database lookup DIM_ACCOUNT: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão DIM_ACCOUNT.

Stream lookup: Este passo adiciona ao fluxo principal de dados o campo COLLECT_NUM calculado em um fluxo secundário. Utilizando para realizar esta tarefa os campos ACCOUNT_ID e LOOKUP_REFERENCE existentes nos dois fluxos.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output FACT_ANALYSIS_MONTH_UA: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Transformação 2 – Get_FACT_ANALYSIS_MONTH_UA_ATB



Esta transformação tem o objetivo de atualizar o campo AVERAGE para os produtos "ATB – Delvo" e "ATB – CHARM", que são calculados de forma errada na transformação 1, pois realiza a média e o correto é o valor máximo.

Passos da transformação

Query: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_analysis*. Este passo só considera resultados do tipo não numérico, com valor de temperatura numérico, UAs do tipo 'Tanque' e apenas os produtos "ATB – Delvo" e "ATB – CHARM".

Null if...: Este passo substitui valores incoerentes ("Incontável" ou "Ausente", por exemplo) de resultados de análise por *NULL*.

Trata ATB: Este passo converte os resultados iguais a "POS" para "1" e "NEG" para "-1".

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Java Script TIME_ID: Este passo calcula através do campo CREATED da b_analysis o TIME_ID referente ao primeiro dia do mês desta data. Neste caso não é necessário fazer o lookup na dimensão TIME_ID, pois este ID pode ser calculado, o que é mais rápido.

PRODUCT lookup, UA lookup, e DIM_ACCOUNT lookup: Estes passos realizam a troca dos códigos utilizados pelo negocio respectivamente para Produtos, Unidades de Análise e Clientes/Produtores, pelos códigos inteiros e incrementais definidos nas dimensões. Ou seja, estes passos criam a ligação entre a fato e as dimensões.

Output mapping: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Sort rows: Este passo ordena os registros por ACCOUNT_ID, UA_ID, PRODUCT_ID, TIME_ID e RESULT. É um pré-requisito para o próximo passo.

Group by: Este passo agrupa os registros pelos campos ACCOUNT_ID, UA_ID, PRODUCT_ID e TIME_ID. E realiza a consolidação do campo RESULT obtendo o valor máximo no mês considerado e criando o campo AVERAGE com esta informação. Esta é a informação que será corrigida na tabela destino.

Select values 2: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Insert / Update FACT_ANALYSIS_MONTH_UA: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados. Ao contrario do "Table Output", este passo não insere simplesmente os registros na tabela destino. Antes de carregar cada registro, este passo verifica se o registro já existia previamente e neste caso o atualiza, caso contrario insere o novo registro.

Tabela Fato de Acessos

Tabela Fato que contem o numero de acessos as diversas telas do sistema utilizado pela Clínica do Leite por Cliente/Produtor, Usuário e posição consolidados por dia. Ou seja, cada registro desta fato contem o numero de acessos a determinada tela realizados em determinado dia por determinado usuário, com determinada posição e associado a determinado Cliente/Produtor.

Origem dos dados

Tabela principal: b_log_audit_trail.

Join's com: b_postn e b_bu.

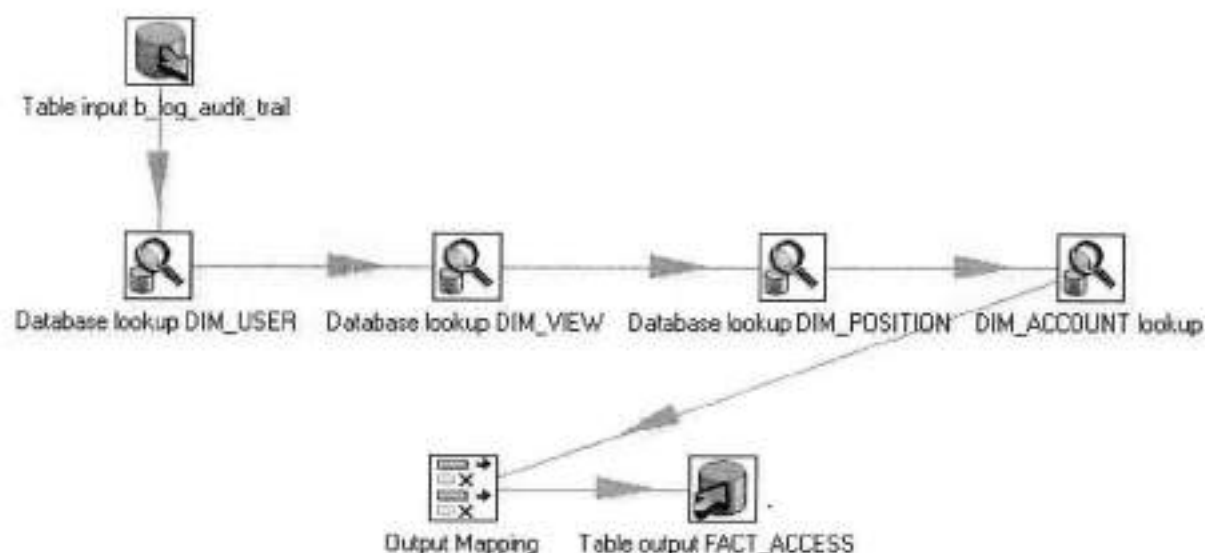
Filtros: Na tabela principal só são utilizados registros não excluídos (DELETED_BY = ""). O agrupamento por usuário, tela, posição, Cliente/Produtor e dia já ocorre nesta query, e conseqüentemente, a contagem do numero de acessos também.

Destino dos dados

Tabela: FACT_ACCESS

Nome do campo	Descrição
USER_ID	Código inteiro e incremental que relaciona o registro ao seu Usuário na dimensão DIM_USER.
VIEW_ID	Código inteiro e incremental que relaciona o registro a sua tela na dimensão DIM_VIEW.
POSTN_ID	Código inteiro e incremental que relaciona o registro a sua posição na dimensão DIM_POSITION.
TIME_ID	Código inteiro e incremental que relaciona o registro ao dia na dimensão DIM_TIME.
ACCNT_ID	Código inteiro e incremental que relaciona o registro ao seu Cliente/Produtor na dimensão DIM_ACCOUNT.
NUM_ACCESS	Número de acessos.

Transformação



Passos da transformação

Table input b_log_audit_trail: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_log_audit_trail*. Este passo já realiza o

agrupamento por usuário, tela, posição, Cliente/Produtor e a contagem do número de acessos.

Database lookup DIM_USER, Database lookup DIM_VIEW, Database lookup DIM_POSITION e DIM_ACCOUNT lookup: Estes passos realizam a troca dos códigos utilizados pelo negócio respectivamente para usuários, telas, posições e Clientes/Produtores, pelos códigos inteiros e incrementais definidos nas dimensões. Ou seja, estes passos criam a ligação entre o fato e as dimensões.

Output mapping: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output FACT_ACCESS: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Tabela Fato de Resultados de Análises consolidados por mês e por tipo de unidade de análise

Tabela Fato que contém os resultados das análises realizadas pela Clínica do Leite consolidados por mês. Esta tabela Fato tem como dimensões a DIM_PRODUCT, DIM_UA_TYPE, DIM_TIME e DIM_ACCOUNT. Ou seja, cada registro desta tabela armazena o resultado da média de análises associada às características produto (CCS, CBT, etc.), tipo de Unidade de Análise, Cliente/Produtor e Mês.

O processo para criação da tabela foi dividido em várias transformações, que estão organizadas por um job.

Origem dos dados

Tabela principal: b_precalculated_month_accnt.

Join's com: b_product.

Filtros: São utilizados registros não excluídos (DELETED_BY = "") da b_product e registros com número de amostras maior que zero na tabela principal.

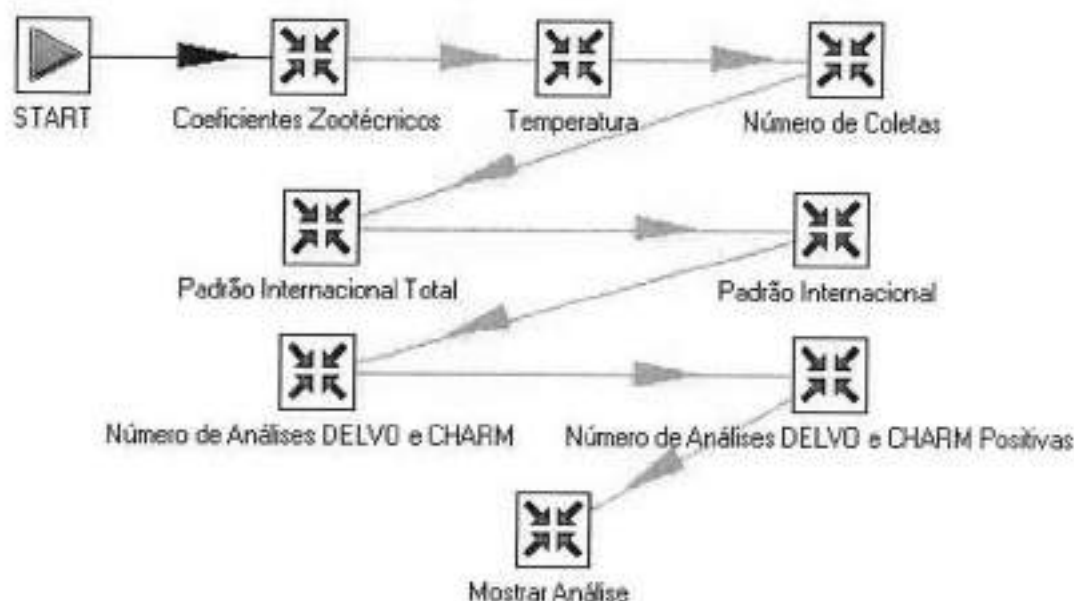
Destino dos dados

Tabela: FACT_ANALYSIS_MONTH

Nome do campo	Descrição
ACCOUNT_ID	Código inteiro e incremental que relaciona o registro ao Cliente/Produtor na dimensão DIM_ACCOUNT.
TIME_ID	Código inteiro e incremental que relaciona o registro a uma data na dimensão DIM_TIME. No caso desta tabela, como os dados são consolidados por mês, os IDs foram definidos como sempre referentes ao primeiro dia de cada mês.

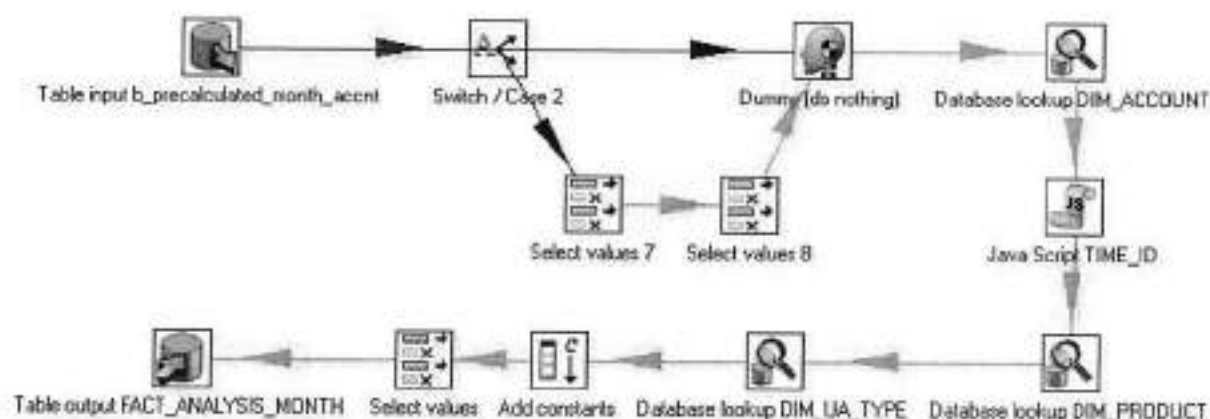
PRODUCT_ID	Código inteiro e incremental que relaciona o registro ao seu Produto na dimensão DIM PRODUCT.
UA_TYPE_ID	Código inteiro e incremental que relaciona o registro ao seu tipo de UA na dimensão DIM UA TYPE.
AVERAGE	Média dos resultados.
STDDEV	Desvio padrão dos resultados.
GEO_AVERAGE	Média geométrica dos resultados.
NUM_SAMPLES	Número de amostras.
MAX_VALUE	Valor máximo encontrado.
MIN_VALUE	Valor mínimo encontrado.
QUARTIL_1	Quartil dos resultados entre o valor mínimo e a mediana.
MEDIANA	Mediana dos resultados.
QUARTIL_3	Quartil dos resultados entre a mediana e o valor máximo.
AVG_TEMP	Temperatura média.
MAX_TEMP	Temperatura máxima.
MIN_TEMP	Temperatura mínima.
COLLECT_NUM	Número de coletas.
SUPPLIERS_NUM	Número de fornecedores.
NO_CONFORM	Número de resultados não conformes.
INT_PATTERN_NUM	Número de unidades de análise dentro do padrão internacional. (consideram-se apenas os resultados de CCS e CBT)
INT_PATTERN_TOT_NUM	Número de unidades de análise que tem resultado de CCS e CBT no mesmo mês.
POS_DELVO_NUM	Número de unidades de análise que tem resultado de ATB_DELVO Positivo.
ATB_DELVO_NUM	Número de unidades de análise que tem resultado de ATB_DELVO.
POS_CHARM_NUM	Número de unidades de análise que tem resultado de ATB_CHARM Positivo.
ATB_CHARM_NUM	Número de unidades de análise que tem resultado de ATB_CHARM.
COLLECT_FLAW_NUM	Número de possíveis falhas de coleta.
COLLECT_FLAW_TOT	Número de resultados de GOR (Gordura).
SHOW_ANALYSIS	Campo que define se o resultado pode ou não ser disponibilizado.

Job



Este é o Job que realiza o processo de carga da tabela Fato Analysis Month. Os passos Coeficientes Zootécnicos, Temperatura, Número de Coletas, Padrão Internacional Total, Padrão Internacional, Número de Análises DELVO e CHARM, Número de Análises DELVO e CHARM Positivas, e Mostrar Análise simplesmente executam as respectivas transformações na ordem apresentada na figura.

Transformação 1 – Coeficientes Zootécnicos



Passos da transformação

Table input b_precalculated_month_accnt: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_precalculated_month_accnt*.

Switch / Case 2: Este passo envia os registros de CCS, CCF, CCP, CCT e CBT para o passo *Select values 7* e os demais para o passo *Dummy*.

Select values 7 e Select values 8: Estes passos em conjunto trocam o nome da coluna QUARTIL_1 para QUARTIL_3 e o nome da coluna QUARTIL_3 para QUARTIL_1.

Dummy: Este passo simplesmente reúne os registros vindos dos passos anteriores.

Database lookup DIM_ACCOUNT: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão DIM_ACCOUNT.

Java Script TIME_ID: Este passo calcula através do campo REFERENCE_DATE da *precalculated_month_acct* o TIME_ID referente ao primeiro dia do mês desta data. Neste caso não é necessário fazer o lookup na dimensão TIME_ID, pois este ID pode ser calculado, o que é mais rápido.

Database lookup DIM_PRODUCT: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão DIM_PRODUCT.

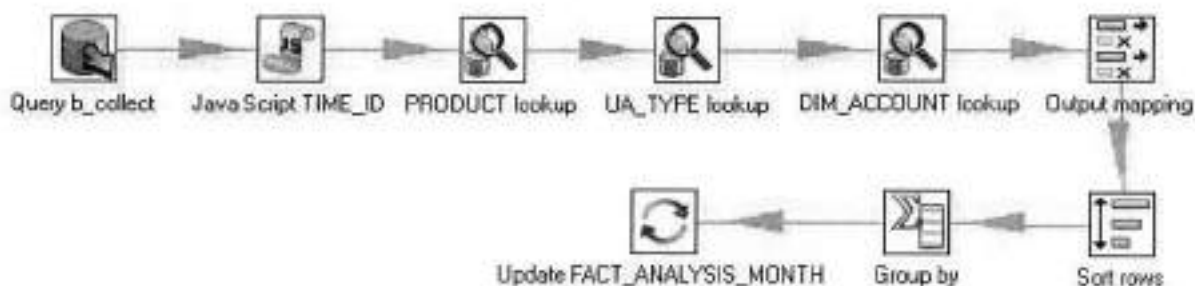
Database lookup DIM_UA_TYPE: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão DIM_UA_TYPE.

Add constants: Este passo insere o valor "N" na coluna SHOW_ANALYSIS de todos os registros. Na transformação *Mostrar Análise* este valor será atualizado.

Select values: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Table output FACT_ANALYSIS_MONTH: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados.

Transformação 2 – Temperatura



Esta transformação tem o objetivo de calcular os campos AVG_TEMP, MAX_TEMP e MIN_TEMP, que não existiam na tabela de origem.

Passos da transformação

Query b_collect: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_collect*.

Java Script TIME_ID: Este passo calcula através do campo *LAST_UPD* da *b_serv_order* o *TIME_ID* referente ao primeiro dia do mês desta data.

PRODUCT lookup: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão *DIM_PRODUCT*.

UA_TYPE lookup: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão *DIM_UA_TYPE*.

DIM_ACCOUNT lookup: Este passo realiza a troca dos códigos utilizados pelo negocio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão *DIM_ACCOUNT*.

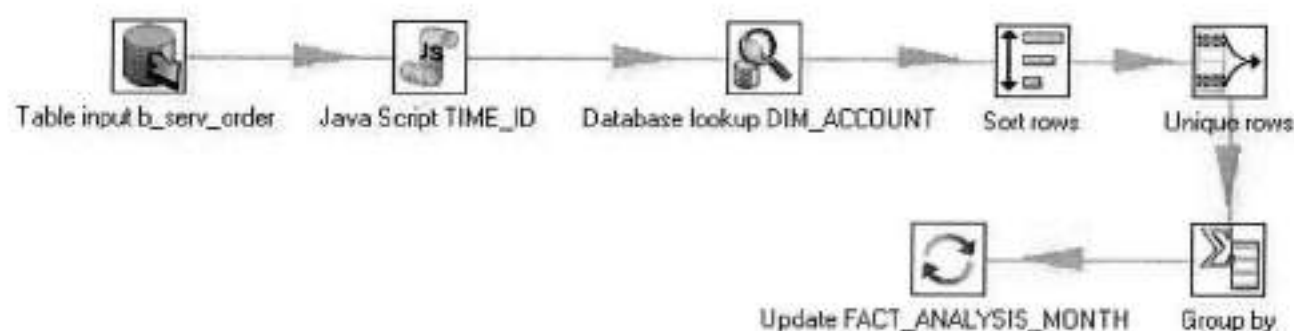
Output mapping: Neste passo só ocorre a mudança de nome de alguns campos para que fiquem iguais aos nomes dos respectivos campos na tabela destino.

Sort rows: Este passo ordena os registros por *ACCOUNT_ID*, *UA_TYPE_ID*, *PRODUCT_ID* e *TIME_ID*. É um pré-requisito para o próximo passo.

Group by: Este passo agrupa os registros pelos campos *ACCOUNT_ID*, *UA_TYPE_ID*, *PRODUCT_ID* e *TIME_ID*. E realiza a consolidação do campo *TEMPERATURE* criando três novos campos: *AVG_TEMP*, com a média das temperaturas, *MAX_TEMP*, com a temperatura máxima e *MIN_TEMP*, com a temperatura mínima.

Update FACT_ANALYSIS_MONTH: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados. Ao contrario do "Table Output", este passo não insere registros na tabela destino. Este passo verifica se o registro já existia e neste caso o atualiza, caso o registro não exista nada irá acontecer. Neste caso os campos que serão atualizado são: *AVG_TEMP*, *MAX_TEMP* e o *MIN_TEMP*.

Transformação 3 – Número de Coletas



Passos da transformação

Table input b_serv_order: Neste passo é onde se encontra a query que retira todos os campos relevantes da tabela *b_serv_order*.

Java Script TIME_ID: Este passo calcula através do campo *COLLECTION_DATE* da *b_serv_order* o *TIME_ID* referente ao primeiro dia do mês desta data.

DIM_ACCOUNT lookup: Este passo realiza a troca dos códigos utilizados pelo negócio para Clientes/Produtores pelos códigos inteiros e incrementais definidos na dimensão *DIM_ACCOUNT*.

Sort rows: Este passo ordena os registros por *ACCOUNT_ID*, *TIME_ID* e *COLLECTION_DATE*. É um pré-requisito para o próximo passo.

Unique rows: Exclui registros duplicados.

Group by: Este passo agrupa os registros pelos campos *ACCOUNT_ID* e *TIME_ID*. E realiza a consolidação do campo *COLLECTION_DATE* criando um novo campo, *COLLECT_NUM*, com o número de datas de coleta de amostras consolidados por produtor e mês.

Update FACT_ANALYSIS_MONTH: É o passo onde é definida a conexão ao banco de dados destino (DW) e indicamos a tabela para onde os dados serão direcionados. Nesta transformação o único campo atualizado é *COLLECT_NUM*.