Instituto Superior de Engenharia do Porto

Desenho e implementação de um data warehouse para a empresa AdClick

João Filipe Lima Albuquerque

Dissertação para a obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Arquitecturas, Sistemas e Redes.

Orientadora: Professora Doutora Maria de Fátima Rodrigues

Orientador na empresa: Engenheiro Filipe José Pimenta da Silva

Júri:

Presidente:

Professor Doutor José António Reis Tavares, Instituto Superior de Engenharia do Porto

Vogais:

Professor Doutor Jorge Fernandes Rodrigues Bernardino, Instituto Superior de Engenharia de Coimbra

Professora Doutora Maria de Fátima Coutinho Rodrigues, Instituto Superior de Engenharia do Porto

Porto, Setembro 2013

Aos meus pais.

Resumo

Esta dissertação incide sobre a problemática da construção de um *data warehouse* para a empresa AdClick que opera na área de marketing digital. O marketing digital é um tipo de marketing que utiliza os meios de comunicação digital, com a mesma finalidade do método tradicional que se traduz na divulgação de bens, negócios e serviços e a angariação de novos clientes.

Existem diversas estratégias de marketing digital tendo em vista atingir tais objetivos, destacando-se o tráfego orgânico e tráfego pago. Onde o tráfego orgânico é caracterizado pelo desenvolvimento de ações de marketing que não envolvem quaisquer custos inerentes à divulgação e/ou angariação de potenciais clientes. Por sua vez o tráfego pago manifesta-se pela necessidade de investimento em campanhas capazes de impulsionar e atrair novos clientes.

Inicialmente é feita uma abordagem do estado da arte sobre *business intelligence e data warehousing*, e apresentadas as suas principais vantagens as empresas. Os sistemas *business intelligence* são necessários, porque atualmente as empresas detêm elevados volumes de dados ricos em informação, que só serão devidamente explorados fazendo uso das potencialidades destes sistemas. Nesse sentido, o primeiro passo no desenvolvimento de um sistema *business intelligence* é concentrar todos os dados num sistema único integrado e capaz de dar apoio na tomada de decisões. É então aqui que encontramos a construção do *data warehouse* como o sistema único e ideal para este tipo de requisitos.

Nesta dissertação foi elaborado o levantamento das fontes de dados que irão abastecer o *data warehouse* e iniciada a contextualização dos processos de negócio existentes na empresa. Após este momento deu-se início à construção do *data warehouse*, criação das dimensões e tabelas de factos e definição dos processos de extração e carregamento dos dados para o *data warehouse*. Assim como a criação das diversas *views*. Relativamente ao impacto que esta dissertação atingiu destacam-se as diversas vantagem a nível empresarial que a empresa parceira neste trabalho retira com a implementação do *data warehouse* e os processos de ETL para carregamento de todas as fontes de informação. Sendo que algumas vantagens são a centralização da informação, mais flexibilidade para os gestores na forma como acedem à informação. O tratamento dos dados de forma a ser possível a extração de informação a partir dos mesmos.

Palavras-chave: *Data warehouse*, *Business Intelligence*, Marketing digital

Abstract

This thesis focuses on the problem of building a data warehouse for AdClick company who operates in the area of digital marketing. Digital marketing is a type of marketing that uses digital media, with the same purpose of the traditional marketing which results on a effective publicity of goods, services and business to attract new clients.

There are several digital marketing strategies in order to achieve these objectives, highlighting organic traffic and paid traffic. Organic traffic is characterized by the development of marketing actions that do not involve any costs related to promote and / or appeal new customers. In the other hand, paid traffic is manifested by the need to invest in campaigns to boost and attract new customers.

First, an approach of the state of the art of business intelligence and data warehousing is made and presented the advantages of them to the companies. The business intelligence systems are needed, because currently firms have a high volume of data rich in information, which will only be fully exploited by making use of the potential of these systems. The first step in developing a business intelligence system is to concentrate all the data in a single integrated data warehouse to support decision making. It is then that we find here the construction of the data warehouse as the unique and ideal for this kind of requirements.

This dissertation begins with a survey of the data sources that will supply the data warehouse and also starts with the contextualization of existing business processes in the company. After this, we begin the construction of the data warehouse, its dimensions and facts tables, and define the processes of data extraction and loading data into the data warehouse. Just as the creation of the views. Concerning the impact that this dissertation reached to the enterprise advantage that the partner in this work takes from the implementation of data warehouse and ETL processes for loading from all sources of information. Since some advantage are, the centralization of information, more flexibility for managers in how they access information. And the opportunity to extract intelligence from data that has been spread in the operational systems.

Keywords: Data Warehouse, Digital Marketing, Business Intelligence

Agradecimentos

Pretendo deixar aqui expressos os meus sinceros agradecimentos a um conjunto de pessoas que foram importantes para mim ao longo da minha vida e em especial na vida académica que culminou com este trabalho de dissertação de Mestrado.

Começo por agradecer á minha madrinha pela educação que me deu na minha infância.

Ao meus pais por me terem permitido tirar uma licenciatura pois sem essa possibilidade nunca teria conseguido chegar até aqui.

À minha irmã, à Inês e aos meus amigos, por terem estado ao meu lado neste momento decisivo da minha vida.

À AdClick por ter aceitado este trabalho de dissertação e me ter dado todos os meios para que o objetivo final fosse atingido.

Ao Professor Doutor Gabriel de Sousa David e Professor Doutor Ricardo Santos Morla pela orientação que me deram no início deste trabalho.

Ao Professor Doutor Paulo Jorge Oliveira por todo o tempo que dispensou com as minhas dúvidas.

À Professora Doutora Maria de Fátima Rodrigues por todo o apoio, conselhos, críticas, orientação e paciência que teve comigo ao longo desta dissertação.

Índice

1 Introdução 1

1.1 Enquadramento 2

1.2 Finalidade e objetivos do trabalho 3

1.3 Organização do documento 3

2 Estado da Arte 5

2.1 Business Intelligence 5

2.2 Construção de um data warehouse 8

2.3 O que é o data warehouse? 8

2.3.1 Orientado por assuntos 9

2.3.2 Integrado 9

2.3.3 Variável no tempo 9

2.3.4 Não volátil 9

2.4 ETL – Extraction, Transformation, Loading 10

2.4.1 Metadados 10

2.4.2 Staging area 10

2.5 Modelo Dimensional 11

2.5.1 Tabela de dimensão 11

2.5.2 Tipos de atualizações nas dimensões 11

2.5.3 Tabela de factos 11

2.5.4 Granularidade 12

2.5.5 Tipos de esquemas 12

2.5.6 Conformed dimensions 14

2.5.7 Data Mart 14

3 Desenho do Data Warehouse 15

3.1 Contextualização da AdClick 16

3.1.1 Apresentação da empresa 16

3.1.2 Gestão de tráfego 16

3.1.3 Angariação de contactos 17

3.1.4 Venda de contactos 18

3.2 Sistemas operacionais 18

3.2.1 Sistemas de tracking 19

3.2.2 Sistema de Armazenamento e integração de contatos 20

3.2.3 Sistema de e-mail marketing 21

3.2.4 Sistema de relatórios HurryUp 21

3.3 Processo de negócio *e-mail marketing* 22

3.3.1 Estrela e-mail sent 22

3.3.2 Estrela e-mail sent daily 25

3.3.3 Estrela e-mail sent conversion 27

3.3.4 Estrela e-mail sent conversion daily 30

3.4 Processo de negócio geração de tráfego 32
3.4.1 Estrela traffic tracking 33
3.4.2 Estrela traffic conversion 35
3.4.3 Estrela traffic overall 37
3.5 Processo de negócio venda de contacto 40
3.5.1 Estrela integration 40
3.5.2 Estrela integration overall 43
3.6 Cálculo do tamanho do *data warehouse* 45
4 Implementação do *data warehouse* 49
4.1 Extração e transformação dos dados 50
4.1.1 *Task* YellowCommand 51
4.1.2 *Task* AdSniperCommand 54
4.1.3 *Task* importAfilea 54
4.1.4 *Task* LeadCentreCommand 55
4.1.5 *Task* HurryUpCommand 59
4.2 Carregamento das dimensões 62
4.2.1 Dimensão dim_date 63
4.2.2 Dimensão dim_time 64
4.2.3 Dimensão dim_country 65
4.2.4 Dimensão dim_email_kit 67
4.2.5 Dimensão dim_email_list 68
4.2.6 Dimensão dim_mta 69
4.2.7 Dimensão dim_integration_response 70
4.2.8 Dimensão dim_campaign 71
4.2.9 Dimensão dim_webpage 74
4.2.10 Dimensão dim_traffic_source 76
4.2.11 Dimensão dim_tracking 77
4.2.12 Dimensão dim_user 78
4.2.13 Dimensão dim_user_extra 82
4.3 Criação das *view* 83
4.3.1 *View* view_user 83
4.3.2 *View* view_microsite 84
4.4 Carregamento dos factos 84
4.4.1 Grupo tabela de factos não agregada 85
4.4.2 Grupo tabela de factos agregadas 86
5 Conclusões e trabalho futuro 87
Referências 89

Lista de Figuras

Figura 1: Arquitetura de um Sistema de *Business Intelligence* (Adaptado de: [Han e Kamber,2001]). 7
Figura 2: Representação do esquema em estrela. 12
Figura 3: Representação do esquema em floco de neve. 13
Figura 4: Representação do esquema constelação de estrelas. 13
Figura 5: Representação do modelo de troca de dados aquando da visita de um utilizador a uma página *web*. 19
Figura 6: Representação da tabela de factos fact_send_email e das suas dimensões. 24
Figura 7: Representação da tabela de factos fact_email_sent_daily e das suas dimensões. 27
Figura 8: Representação da tabela de factos fact_email_sent_conversion e das suas dimensões 29
Figura 9: Representação da estrela fact_email_sent_conversion_daily. 32
Figura 10: Representação da tabela de factos fact_traffic_tracking e das suas dimensões. 35
Figura 11: Representação da tabela de factos fact_traffic_conversion e das suas dimensões. 37
Figura 12: Representação da tabela de factos fact_traffic_overall e das suas dimensões 39
Figura 13: Representação da tabela de factos fact_integration e das suas dimensões. 42
Figura 14: Representação da tabela de factos fact_integration_overall e das suas dimensões. 44
Figura 15: Tabela controller_task responsável por guardar a informação sobre as extrações e carregamentos. 51
Figura 16: Exemplo do JSON retornado com a informação do MTA. 52
Figura 17: Tabela mta existente na *staging area* 53
Figura 18: Exemplo do JSON retornado com a informação acerca do *e-mail* enviado. 53
Figura 19: Exemplo do JSON retornado com a informação acerca do utilizador. 55
Figura 20: Tabela user presente na *staging area* 57
Figura 21: Tabela integration presente na *staging area* 59
Figura 22: Tabela traffic_source presente na *staging area* 60
Figura 23: Representação em forma de grafo das relações Cliente - Conta - Campanha. 61
Figura 24: Representação da hierarquia existente na dimensão dim_date 64
Figura 25: Representação da hierarquia existente na dimensão dim_time. 65
Figura 26: Representação da hierarquia existente na dimensão dim_country. 66
Figura 27: Representação da hierarquia existente na dimensão dim_mta. 70
Figura 28: Representação da hierarquia existente na dimensão dim_campaign. 73
Figura 29: Representação da hierarquia existente na dimensão dim_webpage 75
Figura 30: Representação da hierarquia existente na dimensão dim_user. 82

Lista de Tabelas

Tabela 1 - Diferenças entre *data mart* e *data warehouse*. 14
Tabela 2 - Factos Relativos a tabela de factos fact_email_sent. 23
Tabela 3 - Factos Relativos a tabela de factos fact_email_sent_daily 26
Tabela 4 - Descrição dos factos escolhidos para a tabela de factos fact_email_sent_conversion. 29
Tabela 5 - Descrição dos factos escolhidos para a tabela de factos fact_email_sent_conversion_daily. 31
Tabela 6 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_tracking 34
Tabela 7 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_conversion. 36
Tabela 8 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_overall 38
Tabela 9 - Descrição dos factos escolhidos para a tabela de factos fact_integration. 41
Tabela 10 - Descrição dos factos escolhidos para a tabela de factos fact_integration_overall. 44
Tabela 11 - Matriz em bus com a representação de todos os processos de negócio. 45
Tabela 12 - Espaço ocupado por cada tipo de dados em MySql 46
Tabela 13 - Composição dos atributos da tabela de factos 46
Tabela 14 - Resumo do volume de dados para 1 e 5 anos. 46
Tabela 15 - Opções disponíveis em todas as *tasks* 50
Tabela 16 - Descrição dos atributos da dimensão dim_date 63
Tabela 17 - Descrição dos atributos da dimensão dim_time. 64
Tabela 18 - Descrição dos atributos da dimensão dim_country 66
Tabela 19 - Descrição dos atributos da dimensão dim_email_kit. 67
Tabela 20 - Descrição dos atributos da dimensão dim_email_list. 68
Tabela 21 - Descrição dos atributos da dimensão dim_mta. 69
Tabela 22 - Descrição dos atributos da dimensão dim_integration_response. 70
Tabela 23 - Descrição dos atributos da dimensão dim_email_campaign. 71
Tabela 24 - Descrição dos atributos da dimensão dim_webpage 74
Tabela 25 - Descrição dos atributos da dimensão dim_traffic_source 76
Tabela 26 - Descrição dos atributos da dimensão dim_tracking. 77
Tabela 27 - Descrição dos atributos da dimensão dim_user. 80
Tabela 28 - Descrição dos atributos da dimensão dim_user_extra. 82

Acrónimos e Símbolos

Lista de Acrónimos

DW	*Data Warehouse*
DM	*Data Mart*
ETL	*Extraction, transformation and loading*
BI	*Business Intelligence*
BD	Base de dados
API	*Application Program Interfaces*
OLAP	*On-line Analytical Processing*
ROLAP	*Relational On-Line Analytical Processing*
MOLAP	*Multidimensional On-Line Analytical Processing*
HOLAP	*Hybrid On-Line Analytical Processing*
DOLAP	*Desktop On-Line Analytical Processing*
JSON	*JavaScript Object Notation*
MTA	*Mail Transfer Agent*

1 Introdução

No presente trabalho de dissertação vamos analisar a descrição da implementação de um *data warehouse* para a empresa AdClick [AdClick., 2013a]. Como tal, vamos aqui verificar os processos de negócio que estão implementados na empresa. Sendo estes o *e-mail* marketing, o aumento de tráfego em websites (gestão de tráfego) e a venda de contactos.

Resumidamente os vários processos de negócio podem ser definidos como:

- Gestão de tráfego: é o processo pelo qual é feita a publicidade *onlin*e através dos diversos canais de comunicação com vista a aumentar o número de visitantes a um *website*;

- Venda de contactos: tal como o nome indica é a venda de um contacto que a empresa detém a um dos seus clientes;

- *E-mail* marketing: é um canal de comunicação que faz parte do processo de negócio de aumento de tráfego em *websites*. No entanto este encontra-se separado devido à grande importância que este tem para a empresa;

Todos os processos de negócio supramencionados serão detalhadamente descritos no capítulo 3.

Relativamente ao *data warehouse* (DW) podemos definir este como um sistema de suporte de decisão que permite aos gestores tomarem medidas de forma consistente com os dados históricos da mesma. Deste modo é possível evitar medidas que tenham sido tomadas no passado sem que os objetivos pretendidos tenham sido atingidos. Os sistemas de informação, e no caso o *data warehouse* vem prevenir tais situações. A realidade demonstra que a memória humana é limitada e tem tendência a esquecer, no entanto os sistemas informáticos nada esquecem pois tudo registam. Daqui advém a sua importância tal como iremos ver no presente capítulo sobre o enquadramento do *data warehouse* e a sua importância nas estruturas organizacionais.

1.1 Enquadramento

Com o decorrer dos anos um dos principais problemas com que as empresas se depararam é o elevado volume de dados que estas geram e a incapacidade de retirar informação útil acerca dos mesmos. Isto acontece muitas das vezes devido às empresas apenas possuírem sistemas operacionais onde o único intuito dos mesmos é responder rapidamente ao que está a ocorrer no instante presente e não o que ocorreu no passado. Tal informação histórica acerca do que já ocorreu é importante para se poder prever os acontecimentos vindouros. Em termos de análises existe naturalmente uma que se destaca facilmente em relação a todas as outras que é a análise económica de rentabilidade. Se tomarmos como exemplo uma empresa onde apenas existem sistemas operacionais em que a mesma tem um crescimento anual de 10% mas, no entanto o mês de Agosto e Setembro tem uma quebra de receita de 50% e onde as decisões sobre o futuro da mesma coincidam com um destes dois meses, as decisões tomadas sobre os dados dos sistemas operacionais serão sempre decisões erradas.

Para se tomarem decisões melhores e mais rapidamente deve ter-se em consideração a evolução histórica dos dados. Para tal existem os sistemas de suporte à decisão de onde se destaca a importância do *data warehouse* como o sistema que integra dados relativos a vários anos e permite consultas ágeis aos mesmos.

Nota-se que apesar dos *data warehouses* apenas começarem a surgir no início dos anos noventa os conceitos fundamentais e os seus princípios foram desenvolvidos no início dos anos setenta.

As grandes diferenças entre os sistemas tradicionais e o *data warehouse* assenta na separação de funções a que o *data warehouse* obriga, tais como:

- A extração da informação existente nas várias fontes de dados;
- Uma visão histórica sobre um ou vários processos de negócios ou até mesmo a visão global da empresa.
- A preparação dos dados para fornecer respostas imediatas a todas as questões do negócio.

Assim sendo é facilmente percetível os benefícios de um *data warehouse* em qualquer organização. No entanto estes requerem um elevado custo de tempo e recursos para a sua implementação, o que hoje em dia é considerado um grande fator de exclusão de qualquer projeto. Realidade esta que culmina com o número de empresas proprietárias de um *data warehouse* seja ainda reduzido.

1.2 Finalidade e objetivos do trabalho

Com o presente trabalho de dissertação pretende-se elaborar um *data warehouse* capaz de guardar a informação de todos os processos de negócio da empresa AdClick, tanto no presente momento assim como nos anos futuros.

A utilização de um *data warehouse* para armazenar a informação da empresa é importante em diversos níveis da organização. Antes do presente trabalho a informação encontrava-se espalhada por diversos sistemas operacionais que não estavam sincronizados e por vezes era difícil conseguir uma correspondência dos dados que se encontravam nos diversos sistemas. Assim com a organização da informação num repositório central será possível conferir a como um determinado processo está a evoluir. Como exemplo concreto, e que será minuciosamente explicado no decorrer desta dissertação, teremos o caso de utilizadores que são angariados por uma fonte de tráfego para se efetuar campanhas de *e-mail marketing*. As campanhas de *e-mail marketing* estão assentes no princípio básico que após a receção de um *e-mail* a ação mínima que se espera do utilizador seja o *click* num dos *link* apresentados. Com o implementar deste projeto será possível identificar os utilizadores que efetuam o *click*, quais os *microsites* que os angariaram e quais as palavras utilizadas para a angariação destes utilizadores assim como será possível identificar padrões de palavras utilizadas na angariação dos mesmos.

Ressalve-se, que o objetivo desta dissertação não é a implementação de uma ferramenta de descoberta de conhecimento mas sim a construção de uma base sólida (*data warehouse*) para permitir que estas ferramentas possam trabalhar sobre a mesma. O exemplo anteriormente exposto efetua a ligação entre dois processos de negócio da empresa (*e-mail marketing* e geração de tráfego) que, neste momento não se encontram integrados e trabalham independentemente. Assim, com a criação do *data warehouse* será possível aproximar estes processos sabendo por exemplo exatamente o tipo de palavras chave que se devem usar.

Para além deste tipo de benefícios que serão retirados com a implementação do *data warehouse* existem também outras vantagens muito importantes que são a capacidade de extrair a informação particular de cada processo de negócio, assim como a possibilidade de extrair a visão global de como a empresa está a evoluir.

1.3 Organização do documento

Esta dissertação encontra-se organizada em seis capítulos.

O primeiro capítulo refere-se ao enquadramento geral do âmbito da tese, os problemas que se pretendem solucionar e os objetivos da mesma.

O segundo capítulo destina-se à revisão do estado da arte no âmbito do *data warehousing* assim como uma breve descrição sobre *business intelligence*.

O terceiro capítulo está relacionado com a contextualização da empresa AdClick. Apresenta os sistemas operativos existentes na mesma, os processos de negócio definidos assim como a elaboração do desenho do *data warehouse* implementado e o volume de dados esperado para o *data warehouse*.

O quarto capítulo diz respeito ao trabalho de implementação desenvolvido. Inicia com a descrição dos processos de extração de dados dos sistemas operacionais seguido pelo carregamento dos dados para as dimensões do *data warehouse* assim como das tabelas de factos e finaliza com a criação das *views* a partir das dimensões existentes.

O quinto capítulo apresenta as conclusões, as dificuldades encontradas e suguestões para o trabalho futuro a ser desenvolvido.

2 Estado da Arte

Este capítulo tem como objetivo fornecer um enquadramento da área subjacente a esta dissertação de mestrado, referindo os conceitos relevantes para a compreensão do trabalho desenvolvido.

2.1 Business Intelligence

Business Intelligence (BI) é um termo popular que foi introduzido por Hans Luhn, da IBM em 1958 [Luhn,H. P., 1958]. Este termo reúne um conjunto de métodos e técnicas destinadas a melhorar o processo de tomada de decisão nas organizações, tendo como base um sistema de apoio a este processo [Power, 2007].

Os sistemas de *Business Intelligence* têm em comum um conjunto de objetivos fundamentais. Estes objetivos resumem as características que estão subjacentes a estes sistemas, sintetizando o que eles permitem aos seus utilizadores:

- Acesso a dados fiáveis – a fiabilidade dos dados, a sua fácil integração e compreensão entre áreas é essencial para um exercício consciente de gestão;

- Aumento da transparência e compreensão do negócio – a disponibilização do conhecimento em tempo real (o "quê", o "quanto", o "quando", o "onde" e o "como") permite aos gestores e decisores ter uma perspetiva das áreas que devem controlar com total transparência e aumentar a sua capacidade de compreensão (o "porquê");

- Suporte para a tomada de decisão – só uma compreensão oportuna da realidade pode permitir tomadas de decisões eficazes: como tal, o conhecimento produzido pelos sistemas de *Business Intelligence*, potenciados pelas tecnologias de

comunicação atuais, deve suportar e justificar as medidas tomadas pelos vários intervenientes no processo de gestão.

Os sistemas de *Business Intelligence* englobam um vasto conjunto de dados e aplicações de apoio à tomada de decisão, as quais possibilitam um acesso rápido, partilhado e interativo da informação disponível, bem como a sua análise e manipulação. Através destas ferramentas, os utilizadores podem identificar relações e tendências e transformar grandes quantidades de informação em conhecimento útil [Sezões et al., 2006]. Estas ferramentas não subsistem por si só, pelo que estão constantemente ligadas às fontes de dados subjacentes que residem nos sistemas transacionais das organizações. Para além disto, estes sistemas estão ainda associados a tecnologias como:

- *Data Warehouse* (DW) ou *Data Mart* (DM), repositórios onde ficam armazenados e integrados todos os dados históricos de cariz operacional e transacional extraídos dos sistemas operacionais ou sistemas fonte;

- *Extraction*, *transformation and loading* (ETL), para a seleção, transformação, limpeza, e carregamento dos dados a analisar;

- *Front-End* que é composto pelas tecnologias *On-line Analytical Processing* (OLAP) e *Data Mining*, sendo que o OLAP integra aplicações informáticas que permitem efetuar, de forma rápida e partilhada, a análise de informação sob diversas perspetivas (baseadas no modelo de dados multidimensional definido para o *Data Warehouse*/*Data Mart* que armazena os dados e, por último, a tecnologia de *Data Mining*, cujos algoritmos de análise exploratória de dados permitem identificar padrões ou tendências nos dados analisados.

Existem várias propostas para a arquitetura de um sistema de *Business Intelligence*, contudo a arquitetura aqui referida é defendida por Han e Kamber, e é aqui evidenciada devido à sua abrangência, figura 1.

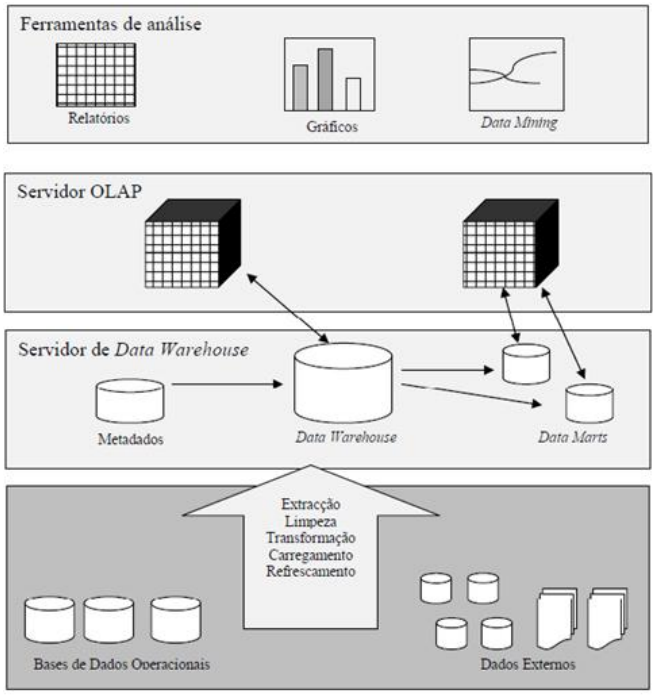

Figura 1: Arquitetura de um Sistema de *Business Intelligence* (Adaptado de: [Han e Kamber,2001]).

De acordo com Han e Kamber [Han e Kamber, 2001] um sistema de *Business Intelligence* muitas vezes adota uma arquitetura constituída por três níveis: o nível do servidor *do Data Warehouse*, o nível do servidor OLAP e o nível das ferramentas de análise.

O primeiro nível integra o servidor de *Data Warehouse*. Os dados das Bases de Dados operacionais e fontes externas são extraídos para um *Data Warehouse* através de API (Aplication Program Interfaces).

O segundo nível integra um servidor OLAP. Este servidor pode ser implementado de acordo com o método de armazenamento de dados escolhido, podendo ser implementado através do método *Relational On-Line Analytical Processing* (ROLAP), *Multidimensional On-Line Analytical Processing* (MOLAP), *Hybrid On-Line Analytical Processing* (HOLAP) ou *Desktop On-Line Analytical Processing* (DOLAP). Neste nível é possível a visualização dos dados conforme as necessidades dos utilizadores sob diversas perspetivas através de cubos de dados.

O terceiro nível integra um conjunto de ferramentas de consulta, ferramentas de análise, e/ou ferramentas de *Data Mining*. Estas ferramentas permitem aos utilizadores chegarem a um conhecimento acerca dos dados provenientes do *Data Warehouse*, através da identificação de padrões e tendências, e da geração de relatórios.

O trabalho desenvolvido nesta dissertação situa-se no primeiro nível da arquitetura *Business Intelligence*, pelo que serão apresentados apenas os principais conceitos relacionados com *Data Warehouse* e o processo ETL.

2.2 Construção de um data warehouse

Na construção de um Data Warehouse é necessário definir o método de implementação, ou seja, qual a metodologia a seguir para o seu desenvolvimento de acordo com as necessidades da empresa. A construção do Data Warehouse pode seguir duas abordagens distintas: top-down ou bottom-up.

A abordagem *top-down* é defendida por Inmon [Inmon, W. H., 2005]. De acordo com este autor a organização cria um *Data Warehouse* e depois parte para a segmentação, ou seja, divide o *Data Warehouse* em áreas mais pequenas (*Data Marts*). Este tipo de abordagem "é muito vantajoso quando a tecnologia é bem conhecida da organização, e quando os problemas de negócio estão bem identificados e são bem compreendidos" [Han e Kamber, 2001]. Os dados contidos no *Data Warehouse* constituem os dados da organização como um todo e tem o propósito de servir posteriormente de base de dados para os diversos *Data Marts*. Este tipo de abordagem, apesar de fornecer algumas vantagens, como uma única Base de Dados homogénea e integrada, sendo a sua manutenção mais simples, uma vez que o repositório de dados é centralizado, têm também alguns problemas tais como o alto custo e tempo de implementação, bem como à criação de expectativas em relação ao ambiente a ser construído, já que a implementação e a obtenção de resultados é demorada.

A abordagem *bottom-up* tem como seus principais defensores Kimball e Ross [Kimball e Ross, 2002], implica que as prioridades das empresas resultem primeiramente no desenvolvimento de *Data Marts*, para as áreas de negócio individuais, sendo assim possível através da integração de todos os *Data Marts* surgir o *Data Warehouse*, não sendo necessária a definição de uma infraestrutura do todo da empresa. Isto é possível devido a esta ser uma arquitectura em *bus* pois todos os *Data Marts* utilizam *conformed dimensions*. No desenvolvimento de uma abordagem *bottom-up* os custos são inferiores aos de um projeto de *Data Warehouse* organizacional, há uma maior rapidez de implementação, bem como a agilidade na apresentação dos resultados e a possibilidade de enfatizar primeiramente os principais sectores de negócio. Sendo que a principal desvantagem encontrada é a falta de padronização dos *Data Marts* que pode resultar na redundância dos dados e em dados inconsistentes, devido a diferentes representações das fontes de dados, trazendo problemas na integração dos dados e na sua fiabilidade.

De modo a obter uma perceção mais clara e pormenorizada desta arquitetura (*bottom-up*), os subcapítulos seguintes abordam os elementos pela qual ela é constituída, incluindo a tecnologia *Data Warehouse*, ETL, *Data Mart* e *Metadados*.

2.3 O que é o data warehouse?

Segundo a definição de Bill Inmon "Um *data warehouse* é uma coleção de dados orientada por assuntos, integrada, variável no tempo e não volátil que tem por objetivo dar suporte aos processos de tomada de decisão [Inmon, W. H., 2005]. "

2.3.1 Orientado por assuntos

Um *data warehouse* é um suporte de dados orientado por assunto o que significa que os dados estão organizados em torno de um processo de negócio em vez de estarem a representar a totalidade do negócio da empresa. Ficando estes disponíveis para análises concretas sobre temas particulares, dando assim a possibilidade do gestor poder tomar decisões suportadas no seu histórico de dados (longo prazo), ao invés dos sistemas operacionais que apenas permitem a análise para a tomada de decisões operacionais (curto prazo).

2.3.2 Integrado

O *data warehouse* caracteriza-se como sendo um sistema integrado pois é constituído pela integração de dados de vários sistemas. Não existem restrições relativamente aos sistemas que disponibilizam dados para o *data warehouse* contudo é importante, que exista uma consistência nos dados. Por exemplo, se num sistema o sexo de uma pessoa é representado por *Male*/*Female* e noutro é representado por M/F é necessário que seja efetuada uma harmonização para a forma que os dados devem assumir e assim serem representados no *data warehouse.*

2.3.3 Variável no tempo

Um *data warehouse* é variável no tempo pois todas as transações são registadas com um atributo de data. Assim sendo, por exemplo num *data warehouse* quando são registados os dados dos clientes (nome, nif, morada, telefone, etc) numa tabela é guardada a data em que o registo aconteceu no atributo created_date. No entanto se porventura o cliente mudar de nome, o registo é atualizado no atributo expired_date com a data de atualização e é criado um novo registo com os dados do cliente e com o atributo created_date com a data que a atualização ocorreu. Com isto, é possível efetuar uma consulta no *data warehouse* com um atributo do cliente que não tenha sido alterado e verificar a data em que foi alterado o nome do cliente.

2.3.4 Não volátil

Esta característica significa que os dados carregados para o *data warehouse* são transações/acontecimentos que ocorreram no passado e como tal não é expectável que sejam efetuadas atualizações sobre as mesmas. Ficando então definido que apenas são necessários dois tipos de operações sobre os dados, o carregamento e o acesso aos dados.

2.4 ETL – Extraction, Transformation, Loading

O processo denominado "Extraction, Transformation and Loading" - ETL (Extracção, Transformação e Carregamento) consiste na primeira fase do processo de obtenção de dados dos sistemas OLTP para o ambiente de data warehouse. As ferramentas de ETL que formam a base de um sistema de BI reúnem e combinam os dados provenientes das diversas fontes organizacionais num *data warehouse*, permitindo aos utilizadores trabalharem numa única plataforma de base de dados, como sendo uma única versão que integra um conjunto de dados.

O ETL, em regra implica três tipos de processos, a saber, a extração de dados dos sistemas operacionais, a transformação destes de forma a não existirem dados não inválidos (NULL) ou a transformação dos mesmos para se adequarem aos diferentes processos de negócio, e por fim, o carregamento dos dados no *data warehouse*. Cumpre ainda afirmar que o processo de extração e carregamento de dados reveste um carácter de obrigatoriedade, a par da fase da transformação que poderá ser facultativa ou opcional.

Dadas estas características, deve-se atribuir uma importância fundamental a este processo na fase de desenvolvimento de um *data warehouse*. De acordo com Inmon [Inmon, W. H., 2005]: "O processo de integração e de transformação de dados geralmente gasta até 80% dos recursos de desenvolvimento".

Pela experiência no corrente projeto, confirma-se que o processo de ETL consome grande parte do tempo de desenvolvimento de um *data warehouse*, pelo que se deve dar uma importância relevante a este processo, pois um processo de ETL mal delineado e desenvolvido pode por em causa a fiabilidade da implementação de todo o *data warehouse*, levando informação inconsistente e com fraca qualidade aos utilizadores finais, e pondo em causa todo o projeto.

2.4.1 Metadados

Os *metadados* é um catálogo de dados que representam um conjunto detalhado de informação sobre os dados. Neste, está guardada toda a informação sobre as características dos dados, tais como, local onde os dados são extraídos, o nome dos campos, o que significam, como foram agregados, quais as transformações que podem ocorrer antes de serem guardados no *data warehouse*, entre outras. Isto vem ainda permitir a consolidação dos dados pois estes têm que ter o mesmo significado independentemente do local onde estão a ser extraídos.

2.4.2 Staging area

A *staging area é* o sistema intermédio de ligação entre os sistemas operacionais onde os dados são extraídos e o *data warehouse*. É aqui que os dados são temporariamente

guardados antes de abastecerem o *data warehouse*. Como tal a *staging area* é composta pelos processos de extração transformação e carregamento dos dados (ETL) e pode ser composta por bases de dados e ficheiros de forma a armazenar a informação recolhida.

2.5 Modelo Dimensional

Na sua forma mais elementar o modelo dimensional consiste numa ou mais tabela de factos, situada no centro da estrela, que está interligada, num formato de estrela, a um conjunto de tabelas de dimensão que contêm a descrição dos factos (dados) armazenados na tabela de factos [Caldeira, C. P. , 2008].

2.5.1 Tabela de dimensão

A tabela de dimensão tem como objetivo a descrição dos factos medidos e armazenados nas tabelas de factos. Sendo que as tabelas de dimensão são altamente desnormalizadas e contêm muitos atributos para se conseguir facultar a maior quantidade de informação sobre os factos recolhidos. As tabelas de dimensão também contêm poucos registos por comparação com as tabelas de factos.

2.5.2 Tipos de atualizações nas dimensões

Tendo em conta que os atributos das dimensões são atualizados ao longo do tempo existem 3 tipos de métodos/técnicas mais usuais para lidar com esta situação. Sendo que destas três os métodos mais usados são o Tipo 1 e Tipo 2.

No caso de ser utilizado o método de Tipo 1 para registar as alterações, os valores dos atributos são reescritos e como tal perde-se o histórico das alterações.

Já no caso do método utilizado ser do Tipo 2 é mantido um histórico de todas as alterações efetuadas. Isto acontece porque em vez dos atributos dos registos serem atualizados o registo passa a estar inativo e é inserido um novo registo com as alterações. Permitindo-se assim ter um histórico de todas as alterações ocorridas.

Por fim o método de Tipo 3, implica que a estrutura das dimensões seja alterada e adicionado um novo atributo à tabela de dimensão.

2.5.3 Tabela de factos

A tabela de factos é o local onde são registados os acontecimentos, métricas ou factos que podem ser utilizados para se ter uma perspetiva do processo de negócio. Esta encontra-se ligada às tabelas de dimensões utilizando as suas chaves artificiais de forma a poder-se obter

informação acerca do facto ocorrido. Saliente-se que normalmente as tabelas de factos representam cerca de 90% do volume total do *data warehouse*.

2.5.3.1 Tipo de factos

No que concerne ao tipo de factos que podemos encontrar num *data warehouse* estes estão divididos em três tipos que são nomeadamente factos aditivos, factos semi-aditivos e factos não-aditivos.

Os factos aditivos são factos que podem ser somados em relação a qualquer dimensão num esquema em estrela. Por sua vez, os factos semi-aditivos apenas podem ser somados em relação a algumas ou até a uma única tabela como por exemplo as contagens. Relativamente aos factos não-aditivos estes não podem ser somados em relação a nenhuma dimensão, exemplo disso são factos que contêm percentagens.

2.5.4 Granularidade

A granularidade representa os dados no seu nível mais elementar, isto é, o grão de atomicidade que os mesmos representam.

É preciso ter em consideração que não é possível responder a questões que estão para além da granularidade escolhida. A granularidade pode variar por assunto, no entanto, para uma tabela de factos todas as métricas têm de ter a mesma granularidade.

2.5.5 Tipos de esquemas

Relativamente aos esquemas que podemos encontrar num *data warehouse*, existem três tipos de esquemas. O primeiro carateriza-se por uma tabela de factos ligada às tabelas de dimensão (figura 2).

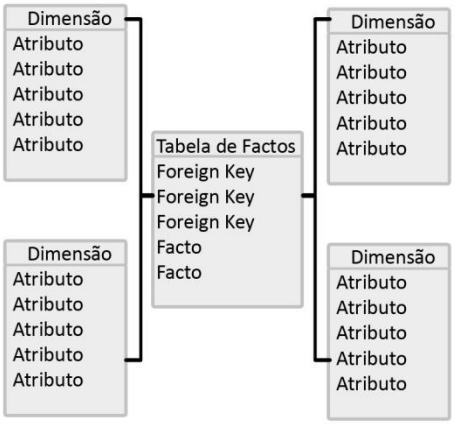

Figura 2: Representação do esquema em estrela.

O esquema em floco de neve é um esquema onde se encontra uma tabela de factos rodeada pelas dimensões e onde estas se encontram normalizadas (figura 3). O que se traduz no desaparecimento do esquema de estrela e toma lugar um esquema onde as várias dimensões se encontram ligadas entre si e a tabela de factos. Este tipo de esquema é desaconselhado devido ao elevado grau de complexidade que acarreta devido aos *joins* que se têm de fazer entre várias tabelas e que com isto aumenta a complexidade de computação [Kimball e Ross, 2008].

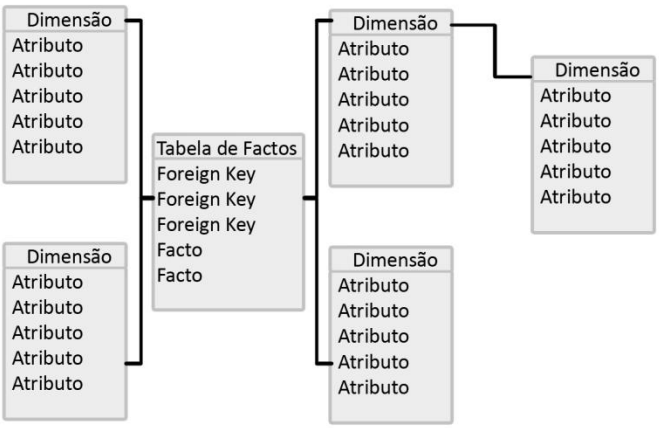

Figura 3: Representação do esquema em floco de neve.

E por fim, a constelação de estrelas caracteriza-se através de um esquema onde existem pelo menos duas tabelas de factos e estas partilham pelo menos uma tabela de dimensão (*conformed dimensions*) como ilustrado na figura 4.

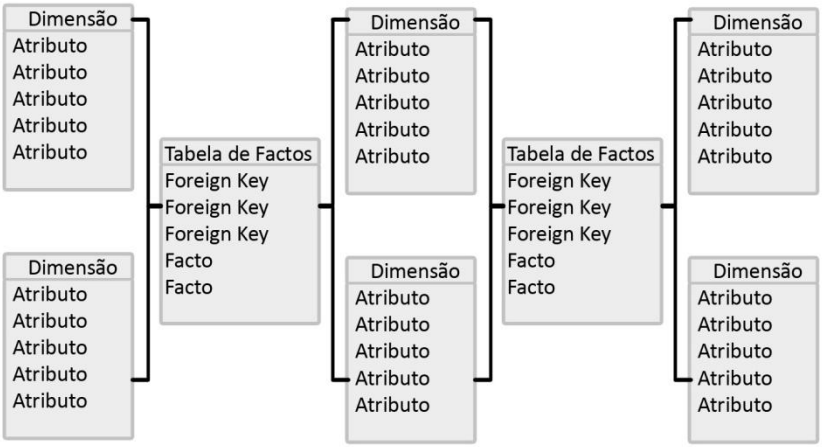

Figura 4: Representação do esquema constelação de estrelas.

2.5.6 Conformed dimensions

As *conformed dimensions* são tabelas de dimensão que se encontram representadas em várias tabelas de factos que atribuem o mesmo significado aos factos relacionados. De notar, que estas dimensões podem estar representadas em vários *data marts* mas têm que manter todos os atributos iguais à dimensão inicial ou um subconjunto dos mesmos, incluindo a mesma chave primária.

2.5.7 Data Mart

Os *data marts* são pequenos *data warehouse* que contêm informação relativa a pelo menos um processo de negócio da empresa. Estes são limitados no seu âmbito devido ao facto de serem muito específicos, e normalmente são utilizados apenas por um departamento ou por conjunto de utilizadores.

Algumas das diferenças entre os *data marts* e os *data warehouse* podem ser visualizados na tabela seguinte.

Tabela 1 - Diferenças entre *data mart* e *data warehouse*.

Data Mart	***Data warehouse***
Orientado ao assunto.	Contém múltiplos assuntos.
Informação por departamento.	Informação da empresa.
Dados agregados (granularidade alta).	Dados no nível mais elementar possível (baixa granularidade).
Histórico parcial dos dados.	Histórico total dos dados.
Um processo de negócio	Múltiplos processos de negócio

3 Desenho do Data Warehouse

Após a introdução dos principais conceitos a abordar serão descritos os principais processos de negócio a implementar no *data warehouse*. Sendo que primeiro definimos a abordagem a adotar como a defendida pelo autor Ralph Kimball. Para o autor, o *data warehouse* é "desenvolvido com base numa arquitetura de *data marts*." [Caldeira, 2008].

A escolha da metodologia recaiu sobre o facto do data warehouse de arquitetura em bus ter por base um desenvolvimento progressivo e por processo de negócio. Sendo certo que este tem um tempo de desenvolvimento mais rápido, uma vez que assenta num conceito que se desenvolve sob uma arquitetura de *data marts*. Assim, torna possível apresentar resultados de uma forma mais célere para se necessário se poder corrigir algumas situações que não tenham sido inicialmente definidas ou detetadas.

Assim, o *data warehouse* será elaborado de forma faseada e segundo o modelo dimensional. Sendo que, este será elaborado através de um conjunto de *data marts* a serem desenvolvidos segundo uma linha congruente para que o somatório de todos os *data marts* represente o *data warehouse*. Na verdade, cada *data mart* deve ser desenvolvido individualmente. Porém, é fulcral ter sempre em consideração que para existir um *data warehouse* todos os *data marts* devem estar ligados através da partilha das suas dimensões (*conformed dimensions*), sob pena de não se construir um *data warehouse* consolidado.

Para alcançar esse objetivo foram promovidas reuniões na empresa. Inicialmente foi efetuada uma reunião geral para se compreender quais os processos de negócio que existem na empresa assim como averiguar a necessidade de separar alguns deles ou mesmo agregá-los. As reuniões seguintes foram apenas efetuadas com os intervenientes diretos em cada processo de negócio.

Os processos de negócio a implementar no presente projeto de investigação são o *e-mail marketing*, geração de tráfego e venda de contactos.

Neste capítulo, será apenas apresentada a estrutura do modelo de dados que nos propusemos a desenvolver definindo as tabelas de factos, juntamente com as dimensões que irão representar os dados de cada estrela. Cumpre salientar, que as dependências entre as dimensões não vão ser representadas nos modelos, como por exemplo na dimensão dim_user a data de nascimento é uma *foreign key* da tabela dim_date. Também irão ficar desde já definidas as respetivas métricas que são necessárias para a correta análise de cada processo de negócio.

3.1 Contextualização da AdClick

3.1.1 Apresentação da empresa

O primeiro passo que iremos dar no âmbito deste trabalho de modelação e construção do *data warehouse* é precisamente, compreender a empresa onde o mesmo será implementado.

Assim, e antes de mais, é fundamental compreender o tipo de negócio da empresa assim como os sistemas existentes. A compreensão dos mesmos é importante pois são estes que armazenam os dados que à posteriori serão extraídos para abastecer o *data warehouse*. Neste sentido, passo a fazer uma breve apresentação da empresa bem como os seus sistemas.

A AdClick [AdClick., 2013] é uma empresa de marketing digital que tem por objetivo gerar negócio para os seus clientes. Este objetivo é alcançado essencialmente através dos seguintes meios, a saber, aumento de tráfego em *websites* e venda de contatos de e para potenciais interessados.

Para melhor alcançar os seus objetivos, a empresa, encontra-se dividida em várias unidades de negócio que interagem entre si que vamos analisar de seguida.

3.1.2 Gestão de tráfego

A gestão de tráfego surge com o intuito fundamental de aumentar o tráfego em *websites*.

Para tal, existem vários canais cujo objetivo é potenciar o aumento de tráfego em *websites*, sendo que os utilizados na AdClick são, os sistemas de publicidade online paga (tráfego pago) e publicidade online gratuita (tráfego orgânico), sendo que o tráfego pago é efetuado pelos serviços da Google AdWords, Bing Ads, *e-mail marketing* ou afiliados.

Um afiliado é uma entidade coletiva ou pessoal que é proprietária de um recurso de *e-mail marketing*, *banner*, entre outros, e que ganha uma comissão por cada ação [About.com, 2013]. Onde a ação definida pode ser a necessidade de redirecionarem um utilizador para um *website*.

Por sua vez o tráfego orgânico traduz-se numa angariação de utilizadores através de motores de busca, redes sociais, *blogs*, entre outros. Estas fontes de tráfego têm como função publicitar *microsites* através de anúncios, imagens ou conteúdos com *links* para os *microsites* que se pretende dinamizar. De salientar, que estes poderão ser externos no caso dos *microsites* não pertencerem à AdClick ou internos, se pelo contrário, os *microsites* forem propriedade da AdClick.

Assim, após o direcionamento de um utilizador para o *microsite*, este pode ser considerado válido, inválido ou pendente no caso de existir uma ação definida para além do simples redireccionamento. Esta validação pode ser operacionalizada de duas maneiras, imediatamente ou então do tipo *server-to-server*. A validação acontece imediatamente quando apenas é necessário o redireccionamento onde assume o valor de válido ou inválido (*real-time).* Já no caso de validação do tipo *server-to-server,* primeiro é enviada uma resposta em *real-time* como pendente e depois de se verificar as ações do utilizador é feita uma comunicação entre servidores com a resposta se esta assume o valor de válido ou inválido.

Em todo o caso, consideramos sempre a existência de uma conversão independentemente do estado que esta venha assumir.

3.1.3 Angariação de contactos

A angariação de contactos é o processo sobre o qual um dado contacto é recolhido com vista a poder ser corretamente armazenado na base de dados da AdClick. Para que tal seja exequível os contactos são enviados para os *microsites* através das fontes de tráfego supra-mencionadas e recolhidos pelos métodos que seguidamente passo a explicar.

O método mais usual para a recolha de contatos acontece através de *microsites* que são, por si, constituídos por várias páginas *web*. Sendo certo que, cada *microsite* é dedicado a um tema específico.

Um *microsite* é normalmente constituído por um conjunto finito de páginas, entre 2 e 6, estando as mesmas dedicadas à recolha de informação de um utilizador para um tema, sendo certo que após a recolha da informação o utilizador passa a ser considerado um contacto e o tema do *microsite* é definido como um vertical. Sendo que um vertical é uma área especifica de um negócio do qual os contactos têm necessidades específicas (exemplos de verticais Finanças, Entretenimento ou Seguros).

Normalmente, em cada página existe a submissão dos dados do utilizador para o sistema operacional. No caso de existirem múltiplas páginas serão efetuadas várias submissões com a informação recolhida na(s) página(s) anterior(es) sobre o mesmo utilizador, o que para o sistema operacional se entende como vários contactos diferentes.

No entanto um *microsite* pode encontrar-se dentro de um *website,* o que significa naturalmente, que um *website* pode angariar um ou mais *microsites*. Outro método de

angariação de contatos que temos ao dispor é a compra de bases de dados externas à AdClick. Após a compra é necessário proceder à avaliação dos interesses subjacentes a cada contacto que se encontra nesta base de dados externa. Para que tal avaliação possa ser concretizada é enviado um, ou vários *e-mails* para o contacto de forma a suscitar o seu interesse. Desse modo, cada *e-mail* pode ser constituído por vários *links* que estão a redirecionar o contacto para vários *microsites* da AdClick. Assim, será possível avaliar se o contacto é válido ou não, e começar a recolher informação específica sobre cada um de forma a avaliar os seus interesses e necessidades.

Neste sentido, será possível de identificar qual a melhor relação a criar entre um contacto com as campanhas dos clientes da AdClick.

3.1.4 Venda de contactos

No que respeita à venda de contactos este processo pode ser explicado como sendo a venda de contactos que a própria AdClick possui na sua base de dados aos seus clientes. Ainda de salientar, que existem vários processos que se interligam para que tal seja possível, a saber, o processo de angariação de contactos que se encontra intimamente ligado com o processo de gestão de tráfego.

Quando existe a entrega (venda) de um contacto a um cliente isto é definido como sendo uma integração. Esta pode vir a assumir vários estados que são, pendente, integrado, invalidado e rejeitado.

Após este momento o cliente tem a possibilidade de enviar uma ou várias respostas acerca da integração ocorrida.

Para cada cliente existe ainda a hipótese de serem definidas várias contas no caso de este operar em diferentes países. Sendo que para cada conta existem várias campanhas onde podem estar a ser integrados contatos com interesses distintos e para campanhas distintas, ou no limite, o mesmo contacto possa ter interesse em duas campanhas distintas.

De notar que, para cada campanha está definido um limite máximo de invalidações, das integrações a serem efetuadas, consoante o contrato previamente acordado entre as partes.

3.2 Sistemas operacionais

Existem vários sistemas que são utilizados na AdClick para uma concretização eficaz dos seus objetivos empresariais.

De seguida iremos aqui verificar, de forma breve, cada um deles assim como a explicação da sua função na empresa.

3.2.1 Sistemas de tracking

Os sistemas de *tracking* de uma forma geral são utilizados para armazenar a informação dos utilizadores que visitam um dado *website*.

Na AdClick são utilizados dois sistemas de *tracking*, a saber, o AdSniper [AdClick., 2013] e o Afilea [Afilea, 2013], que são configurados para serem utilizados quando existe um processo de geração de tráfego pago. Estes sistemas não são do tipo convencional pois existe a necessidade de se registar um identificador único para cada utilizador sendo assim possível efetuar uma relação com a integração ou conversão final que venha a ocorrer.

No caso de se utilizar este canal de propaganda (tráfego pago) é colocado um *link* no anúncio que nos vai redirecionar para um sistema que guarda a informação sobre o visitante que corresponde ao nome do browser, versão, língua, sistema operativo, página de origem, página de destino, endereço de IP, data e hora do pedido, e um identificador único (subid). Este processo encontra-se representado na figura 5.

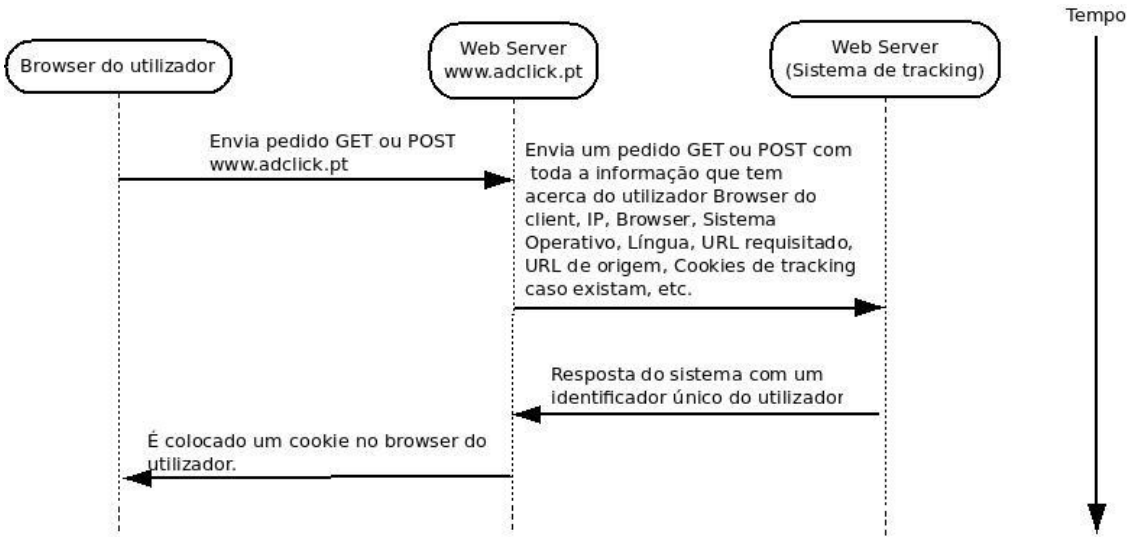

Figura 5: Representação do modelo de troca de dados aquando da visita de um utilizador a uma página *web*.

Após o registo desta informação o visitante é redirecionado para a página de destino que realmente foi anunciada, juntamente com o identificador único correspondente ao registo efetuado no sistema de *tracking*.

O identificador único (subid) serve, precisamente para, identificar a informação do utilizador no caso de se vir a verificar a ocorrência de uma conversão ou do utilizador se registar no *microsite* e este ser propriedade da AdClick.

Cumpre ainda salientar que no caso de se utilizar a ferramenta de *tracking* AdSniper é também, possível efetuar uma distribuição de utilizadores na forma de A/B teste. O A/B teste

é uma técnica utilizada em marketing para mostrar a dois grupos de utilizadores o mesmo conteúdo apresentando-o de forma diferente. Como por exemplo:

- Exemplo 1: um formulário de 50 questões booleanas em que o teste A é constituído por 5 folhas e o teste B é constituído por 1 folha devido ao tamanho da letra utilizado ser menor.

- Exemplo 2: na internet temos como exemplo duas páginas com um formulário onde num caso o formulário está à direita e noutra o formulário está à esquerda.

No exemplo 1 será interessante medir o tempo médio de resposta dos utilizadores ao formulário. Enquanto no exemplo 2 será interessante avaliar o número de utilizadores que preenchem o formulário.

A importância deste tipo de distribuições de A/B teste prende-se com o facto de a mesma poder avaliar *microsites* iguais na sua essência mas diferentes na sua forma (visual), sendo certo que, por exemplo, a cor de fundo pode influenciar acima de 30% [37signals, 2013] na taxa de conversão final.

O sistema de *tracking* Afilea é um *software* externo da empresa Hasoffers [HasOffers, 2013], por sua vez, permite não só efetuar o *tracking* de utilizadores, como também a possibilidade de gerir campanhas. Facto que se concretiza num alargar de possibilidades de angariação de tráfego para os *microsites*. Este sistema opera através de uma rede de afiliados compostos por empresas ou pessoas individuais que angariam contactos utilizando o *link* do sistema de *tracking* juntamente com o seu identificador de afiliado permitindo assim identificar o afiliado como sendo uma fonte de tráfego.

Uma outra forma de se efetuar o *tracking* aos visitantes é a utilização do *access log* que existe nos servidores *web*. No entanto, este tipo de sistema de *tracking* não é utilizado na AdClick porque não seria possível efetuar a correspondência entre os dados do *access log* e as integrações e/ou conversões.

3.2.2 Sistema de Armazenamento e integração de contatos

Ao nível do armazenamento e integração de contatos, o sistema operacional que encontramos na AdClick é a ferramenta proprietária LeadCentre [AdClick., 2013]. Este é um repositório que guarda a informação relativa aos contactos e é responsável pela sua integração com as campanhas dos clientes, termos e condições, local de recolha de contactos, entre outros. Devido ao facto deste ser o local onde são efetuadas as integrações, também aqui estão presentes os dados relativos aos clientes e às suas campanhas.

Cumpre salientar que os contactos nunca são atualizados, mesmo em *microsites* onde acontece várias vezes a submissão de dados sobre um mesmo contacto. Para que a integração possa ocorrer é necessário que exista uma associação entre a página visitada pelo contacto e

as campanhas. Sendo que este tipo de associação é definida previamente pelos gestores de campanhas e é operada através deste sistema.

São também aqui definidas todas as regras para que um contato possa ser integrado numa campanha. A título de exemplo, cumpre referir que para o mesmo *microsite* uma campanha pode ter uma regra que só aceita contatos com idade superior a 18 anos e inferior a 22 anos e outra campanha que só aceita contatos em que o *e-mail* do utilizador é do domínio Gmail.com. De notar que não é possível validar a veracidade dos dados, logo a regra é efetuada sobre os valores recolhidos.

3.2.3 Sistema de e-mail marketing

Relativamente ao sistema usado para responder às necessidades de *e-mail marketing* foi igualmente desenvolvido ao longo do tempo, uma ferramenta proprietária que se encontra madura e detém a denominação de Yellow [AdClick., 2013]. Neste sistema são definidos os envios de *e-mails* compostos pelas listas de *e-mails*, o conteúdo do *e-mail* e é guardada a informação sobre cada *e-mail*. Aqui, é também armazenada informação correspondente ao número de vezes que o *e-mail* foi aberto (número de aberturas), número de *clicks* nos *links* do conteúdo do *e-mail* e se o envio foi considerado um *soft bounce*, *hard bounce* ou *unsubscribe,* ou seja, se o contacto envia um pedido para deixar de pertencer à lista onde está registado.

O *soft bounce* é a resposta do servidor de e-mail no caso da caixa de e-mail se encontrar cheia ou o servidor de e-mail estar com problemas e ao terceiro *soft bounce* o endereço de *e-mail* é considerado um *hard bounce.*

Um *hard bounce* é quando o servidor de *e-mail* não consegue encontrar a caixa de *e-mail* e então retorna um erro permanente.

3.2.4 Sistema de relatórios HurryUp

O HurryUp [HurryUp, 2013] é atualmente a ferramenta que centraliza a informação dos sistemas da AdClick. Para tal, este agrega a informação efetuando os pedidos dos dados através das API's existentes nos outros sistemas. Este sistema é igualmente utilizado para efetuar relatórios da informação dos sistemas operacionais.

É ainda neste sistema que são definidas relações muito relevantes tais como o conjunto de páginas *web* que são consideradas como um *microsite* bem como o conjunto de *microsites* que são considerados como um *website* e as relações entre os clientes com as suas contas e quais as suas campanhas.

No *data warehouse* que vai ser desenvolvido no âmbito desta dissertação deverá conseguir responder, entre outras questões, a todos os relatórios que atualmente se encontram disponíveis no sistema HurryUp.

3.3 Processo de negócio *e-mail marketing*

Este processo de negócio tem como objetivo fundamental efetuar o *tracking* relativo ao envio de *e-mails* assim como contabilizar o lucro a partir dessa análise. Aqui, pretende-se que todas as métricas relativas a este processo de negócio estejam devidamente representadas, pois só assim, se poderá efetuar uma análise clara e eficaz e calcular devidamente os proveitos associados a este processo de negócio.

Neste *data mart* poderá encontrar as estrelas *e-mail sent*, *e-mail sent daily*, *e-mail sent conversion* e *e-mail sent conversion daily*. As estrelas cujo nome termina em *daily* são constituídas por tabelas de factos agregadas ao dia, como por exemplo, a tabela de factos fact_email_sent_daily. Neste caso, esta tabela é idêntica à fact_email_sent, diferenciando-se apenas, a primeira da segunda, no que respeita ao facto de a informação se encontrar mais agregada. O que se traduz que nesta situação à tabela de factos fact_email_sent_daily vai ter uma granularidade superior (menos detalhe). Sendo que para a sua concretização algumas dimensões deixarão de estar representadas no esquema.

No caso da tabela de factos fact_email_sent pretende-se que esta contenha a informação relativa a todos os envios de *e-mails* que ocorreram. Para que tal se suceda, esta será particionada pela date_key que é a chave artificial da dimensão dim_date e que representa na tabela de factos a data do envio do *e-mail*.

Por sua vez, a estrela *e-mail sent conversion* tem por objetivo medir os proveitos que foram registados em cada *e-mail* enviado. Assim sendo, esta será particionada com a date_key que corresponde à chave artificial da dimensão dim_date que, por si, representa a data em que ocorreu o envio. Também existe a *foreign key* conversion_date_key que representa a data em que ocorreu a conversão. De notar que neste caso, podemos definir que a conversão acontece quando um contacto efetua uma dada ação através do *e-mail* que recebeu.

3.3.1 Estrela e-mail sent

Comecemos então pela estrela *e-mail sent* que tem como objetivo registar a informação relativa a todos os envios de *e-mails*.

3.3.1.1 Granularidade

A granularidade escolhida para este processo de negócio é o envio do *e-mail*. Sendo que este é relativo a data de envio, instante em que este foi enviado (hora-minuto-segundo) o utilizador para quem foi enviado, *kit* de *e-mail* utilizado, a lista a que o utilizador pertence e o MTA que foi utilizado para efetuar a entrega do *e-mail*. Nesse sentido, deverá ter-se em atenção que o conjunto que define univocamente cada registo na tabela de factos é constituído pelo conjunto de *foreign keys* date_key, time_key, user_key, email_kit_key, email_list_key e mta_key que é a chave primária da tabela de factos.

3.3.1.2 Escolha das dimensões

As dimensões escolhidas para esta estrela são:

- dim_date;
- dim_time;
- view_user;
- dim_email_kit;
- dim_email_list;
- dim_mta;
- dim_traffic_source;

Estas dimensões representam todas as ligações com as *foreign keys* que irão estar presentes na nossa tabela de factos, cobrindo assim todas as necessidades dos futuros relatórios que venham a ser efetuados.

3.3.1.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 2 que se encontra abaixo.

Tabela 2 - Factos Relativos a tabela de factos fact_email_sent.

Facto	Tipo	Descrição
sent	facto aditivo	Número total de envios.
opens	facto aditivo	Contém o total de vezes que o *e-mail* foi aberto.
click_total	facto aditivo	Total de vezes que foi efetuado um *click* sobre um dos *links* apresentados.
click_unique	facto aditivo	Total de *clicks* distintos por utilizador. Isto é mesmo que um utilizador *click* 20 vezes num *link* apenas iremos contabilizar 1 *click*.
soft_bounce	facto aditivo	Contém o total de erros do tipo *soft bounce*.
hard_bounce	facto aditivo	Contém o total de erros do tipo *hard bounce*.
unsubscribed	facto aditivo	Campo booleano que contém 1 se existiu um *click* sobre o *link* para cancelamento do registo na base de dados.
open_rate	facto não-aditivo	Percentagem que define o total de aberturas (opens) a dividir pelo total de envios.
soft_rate	facto não-aditivo	Percentagem que define o total de *soft bounces* a dividir pelo total de envios.
hard_rate	facto não-aditivo	Percentagem que define o total de *hard*

		bounces a dividir pelo total de envios.
soft_hard	facto não-aditivo	Percentagem que define o total de *soft bounces* a dividir pelo total de *hard bounces*.
unsub_rate	facto não-aditivo	Percentagem que define o total de cancelamentos de registos na base de dados (*unsubscribed*) a dividir pelo total de envios.
unsub_open_rate	facto não-aditivo	Percentagem que define o total de cancelamentos de registos na base de dados (*unsubscribed*) a dividir pelo total de aberturas (*opens*).

3.3.1.4 Atributos extra

De forma a simplificar o acesso aos dados foram ainda registados na tabela de factos alguns atributos que se encontram nas dimensões que são o user_traffic_source_key, user_subscription_in_email_list_date_key e o user_country_key.

3.3.1.5 Esquema apresentado

Após a definição das dimensões e dos factos que foram anteriormente apresentados (Tabela 2), foi elaborada a representação do esquema da tabela de factos com as suas dimensões que pode ser analisado na figura 6.

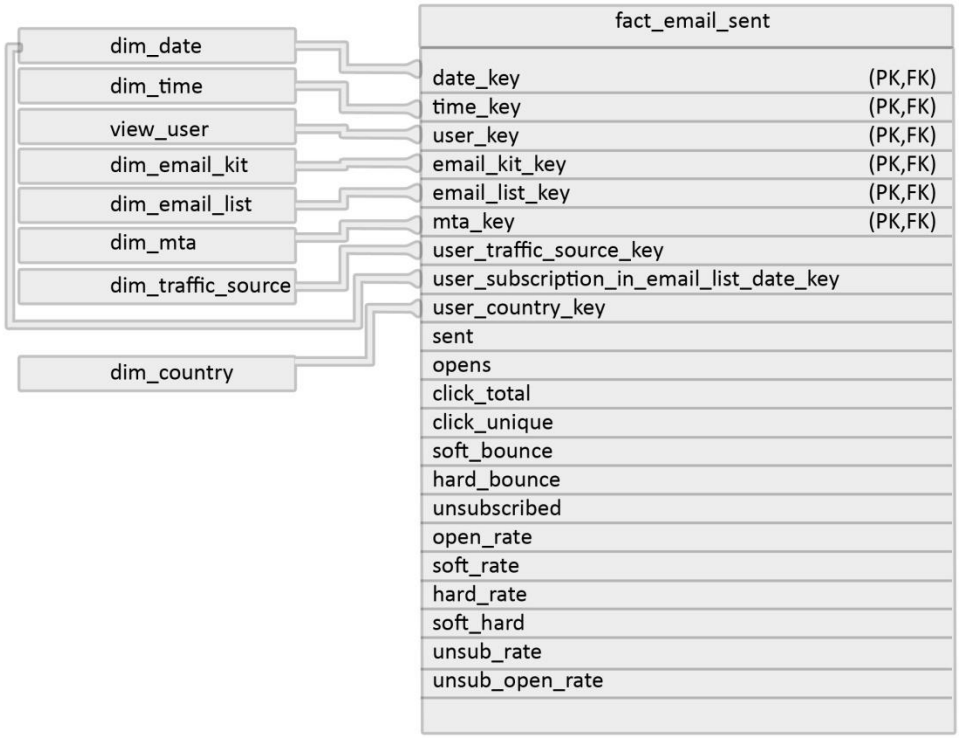

Figura 6: Representação da tabela de factos fact_send_email e das suas dimensões.

3.3.1.6 Volume de entradas diárias na tabela de factos

Neste subcapítulo vamos proceder à análise relativa ao volume de entradas diárias para a tabela de factos fact_email_sent. Para que tal possa ser possível, foram efetuadas as devidas consultas aos sistemas operacionais.

Concluindo-se através das consultas efetuadas aos sistemas operacionais que o total de envios é aproximadamente de 1 257 824 novos registos por dia.

3.3.2 Estrela e-mail sent daily

Foi igualmente construída uma estrela para armazenar os dados agregados relativamente ao dia, denominada de estrela *e-mail sent daily*. Para que tal possa acontecer, esta última, foi construída a partir da estrela supramencionada (estrela *e-mail sent*) de forma acelerar as consultas de acesso aos dados.

3.3.2.1 Granularidade

Tal como foi anteriormente referido, a informação que consta nesta estrela *e-mail sent daily*, tem um nível de granularidade superior (menos fino). Sendo que a granularidade é definida pelos envios de *e-mails* num dia com um *kit e-mail* (email_kit_key) para uma lista de contactos (email_list_key) por data de subscrição do contacto na lista (subscription_in_email_list_date_key) , por um agente que efetuou o envio (mta_key) e pela fonte de tráfego que angariou o contacto (traffic_source_key).

3.3.2.2 Escolha das dimensões

As dimensões escolhidas para a estrela em apreço são as mesmas que foram anteriormente escolhidas para a estrela *e-mail sent*. Desta última excceciona-se a dimensão dim_time e a *view* view_user. Uma vez que agora não pretendemos saber o instante nem chegar à individualidade do contacto. Convém ainda ressalvar que apesar de se perder a individualidade do contacto é importante manter a fonte de tráfego que o angariou, assim como, a data de subscrição do contacto na lista de *e-mails* (subscription_in_email_list_date_key). Assim, a traffic_source_key e a subscription_in_email_list_date_key passam a fazer parte da nossa chave primária e no caso estas são *foreign keys* da tabela dim_traffic_source e dim_date.

Ficando assim determinadas as seguintes dimensões:

- dim_date;
- dim_email_kit;
- dim_email_list;

- dim_mta;
- dim_traffic_source;

3.3.2.3 Escolha dos factos

Relativamente aos factos escolhidos, são os mesmos que foram definidos para a tabela fact_email_sent e que se encontram explicados no ponto 3.1.1.3 com o acréscimo das métricas total_new_subscriber e total_subscriber. Todos os factos encontram explicação na tabela 3.

De salientar que existem factos não-aditivos, porque esta tabela tem uma taxa de crescimento muito elevada, no entanto a taxa de atualização apenas é elevada nos primeiros dias. Isto porque é nos primeiros dias que existem mais *clicks* e aberturas nos *e-mails* por parte dos utilizadores. Note-se que os factos não-aditivos são atualizados a cada modificação que ocorre nos registos.

Tabela 3 - Factos Relativos a tabela de factos fact_email_sent_daily.

Facto	Tipo	Descrição
sent	Facto aditivo	Número total de envios.
opens	Facto aditivo	Contém o total de vezes que o *e-mail* foi aberto.
click_total	Facto aditivo	Total de vezes que foi efetuado um *click* sobre um dos *links* apresentados.
click_unique	Facto aditivo	Total de *clicks* distintos por utilizador. Isto é mesmo que um utilizador *click* 20 vezes no *link* apenas iremos contabilizar 1 *click*.
soft_bounce	Facto aditivo	Contém o total de erros do tipo *soft bounce*.
hard_bounce	Facto aditivo	Contém o total de erros do tipo *hard bounce*.
unsubscribed	Facto aditivo	Campo booleano que contém 1 se existiu um *click* sobre o *link* para cancelamento do registo na base de dados.
open_rate	Facto não-aditivo	Percentagem que define o total de aberturas (*opens*) a dividir pelo total de envios.
soft_rate	Facto não-aditivo	Percentagem que define o total de *soft bounces* a dividir pelo total de envios.
hard_rate	Facto não-aditivo	Percentagem que define o total de *hard bounces* a dividir pelo total de envios.
soft_hard	Facto não-aditivo	Percentagem que define o total de *soft bounces* a dividir pelo total *hard bounces*.
unsub_rate	Facto não-aditivo	Percentagem que define o total de cancelamentos de registos na base de dados (*unsubscribed*) a dividir pelo total de envios.
unsub_open_rate	Facto não-aditivo	Percentagem que define o total de cancelamentos de registos na base de dados (*unsubscribed*) a dividir pelo total de aberturas (*opens*).

total_new_subscriber	Facto não-aditivo	Total de contactos na lista.
total_subscriber	Facto aditivo	Número total de novos contactos na lista.

3.3.2.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 7.

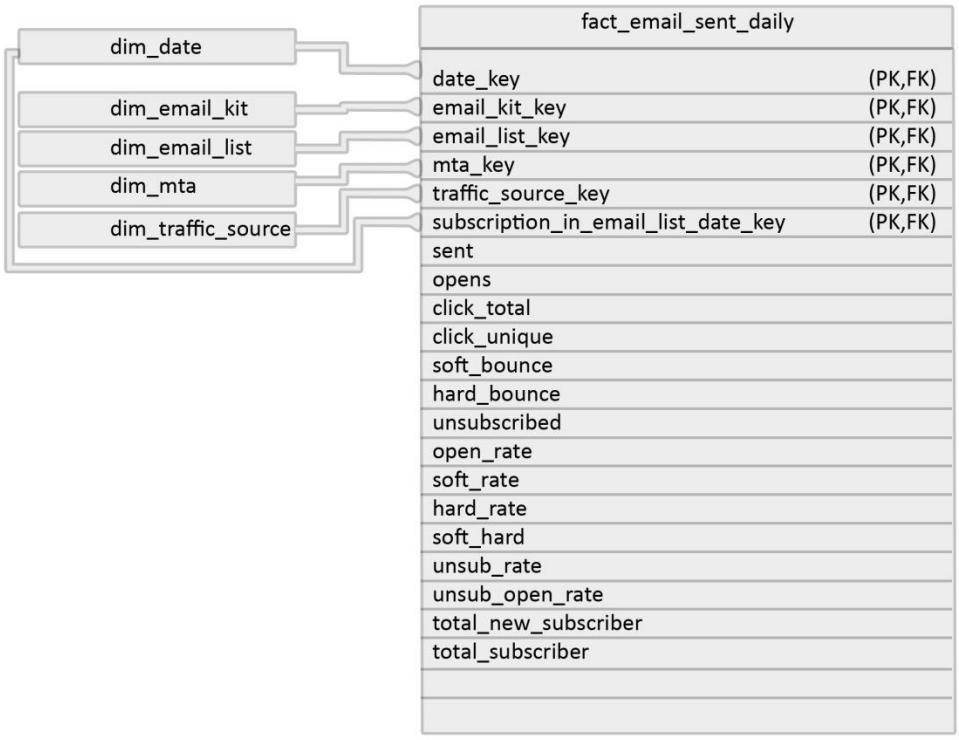

Figura 7: Representação da tabela de factos fact_email_sent_daily e das suas dimensões.

3.3.2.5 Volume de entradas diárias na tabela de factos

Neste subcapítulo vamos proceder à análise relativa ao volume de entradas diárias para a tabela de factos fact_email_sent_daily, tendo em conta as consultas aos sistema operacionais efetuadas no subcapítulo 3.1.1.6. No entanto para esta estrela apenas necessitamos efetuar a agregação relativa à nossa chave primária e não de todos os envios.

Como tal, concluímos através de várias análises das quais foi feita uma média que o total de novos registos diários a serem inseridos na tabela de facto é de 419 274.

3.3.3 Estrela e-mail sent conversion

Com o desenho da estrela *e-mail sent,* tal como a sua designação refere, esta tem como objetivo registar a informação relativa a todos os envios de e-mails. Surgiu então a

necessidade de avaliar os resultados económicos dos envios efetuados. A verificação destes resultados económicos é conseguida através da verificação das conversões, que podemos definir como o acontecimento resultante de uma ação efetuada por um contacto após a receção de um *e-mail*. Tais conversões são registadas de acordo com as campanhas dos clientes onde ocorre a ação desejada. A titulo de exemplo a ação pode ser a visita ao site anunciado no *e-mail,* efetuar a visita a um site anunciado no *e-mail* e uma compra no decorrer da visita, efetuar a visita a um site anunciado no *e-mail* e o preenchimento de um formulário no site, entre outros.

3.3.3.1 Granularidade

Para podermos avaliar os resultados dos envios, a granularidade a definir na tabela de factos fact_email_sent_conversion deve ser igual à escolhida na tabela fact_send_email. Isto visa permitir que, no limite, se consigam efetuar análises tendo em vista descobrir os ganhos exatos provenientes de cada envio.

Assim sendo, a granularidade escolhida é a chave primária da tabela fact_sent_email juntamente com a data de conversão, o instante em que ocorreu e a companha associada.

Deste modo, conclui-se que o conjunto que define univocamente cada registo nesta tabela de factos é constituído pelo seguinte conjunto de *foreign keys* date_key, time_key, user_key, email_kit_key, email_list_key, mta_key, conversion_date_key, conversion_time_key e campaign_key.

3.3.3.2 Escolha das dimensões

As dimensões escolhidas para esta estrela são:

- dim_date;
- dim_time;
- view_user;
- dim_email_kit;
- dim_email_list;
- dim_mta;
- dim_campaign;
- dim_traffic_source;
- dim_country;

Estas dimensões e view, foram escolhidas de acordo com a necessidade de resposta aos relatórios que se pretendem criar, bem como com as *foreign keys* que foram escolhidas para estarem presentes na tabela de factos.

3.3.3.3 Atributos extra

De notar que para simplificar o acesso aos dados foram ainda registados na tabela de factos alguns atributos que se encontram nas dimensões que ligam a tabela de factos e que são o user_traffic_source_key, user_subscription_in_email_list_date_key e o user_country_key.

3.3.3.4 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 4.

Tabela 4 - Descrição dos factos escolhidos para a tabela de factos fact_email_sent_conversion.

Facto	Tipo	Descrição
revenue	Facto aditivo	Lucro pela conversão ocorrida na campanha.

3.3.3.5 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 8.

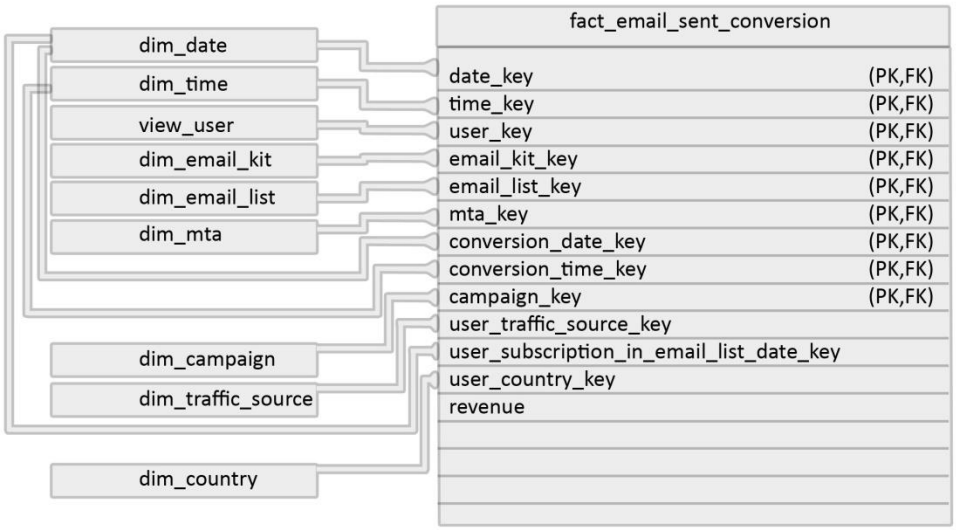

Figura 8: Representação da tabela de factos fact_email_sent_conversion e das suas dimensões.

3.3.3.6 Volume de entradas diárias na tabela de factos

Relativamente a este ponto, vamos efetuar as análises relacionadas com o volume de entradas diárias para a tabela de factos fact_email_sent_conversion. Para ser possível executar esta análise, foram efetuadas consultas através dos dados existentes nos sistemas operacionais.

Assim sendo, sabemos, pelas consultas efetuadas, que diariamente são recebidas aproximadamente 3 233 conversões por cada campanha.

Como tal, para as conversões recebidas ao longo de um dia temos, 3233 * 20 (do total das campanhas existentes este é o valor médio de campanhas que diariamente recebem conversões).

Logo, espera-se um total de 64 660 novos registos por dia.

3.3.4 Estrela e-mail sent conversion daily

De forma a concluir o *data mart* de *e-mail marketing* foi desenhada uma última estrela, a *e-mail sent conversion daily*. Esta estrela foi criada de forma a ser possível emparelhá-la com a estrela *e-mail sent daily* e assim efetuarmos as respetivas análises diárias económicas de forma mais rápida e com uma granularidade mais elevada.

3.3.4.1 Granularidade

Tendo em conta que a granularidade desta estrela deve ser equivalente a *e-mail sent daily* com a adição das respetivas campanhas para ser assim possível registar a campanha onde ocorreu a conversão. Com isto, será então possível a análise da rentabilidade diária dos envios efetuados.

Logo, a atomicidade dos dados será composta pela representação da conversão que ocorreu num dado dia (conversion_date_key) para uma campanha, (campaign_key) relativa aos envios de um *kit e-mail* (email_kit_key) *e-mails* esses que foram enviados num dia, (date_key) para uma lista de contactos, (email_list_key) com uma data de subscrição na lista, (subscription_in_email_list_date_key), pelo agente que efetuou o envio (mta_key) e recolhidos por uma fonte de tráfego que os angariou (traffic_source_key).

Ficando então definido que a constituição da chave primária da nossa tabela de factos é composta pelas *foreign keys* date_key, email_kit_key, email_list_key, mta_key, conversion_date_key, campaign_key, subscription_in_email_list_date_key e traffic_source_key.

3.3.4.2 Escolha das dimensões

As dimensões escolhidas para esta estrela são:

- dim_date;
- dim_email_kit;
- dim_email_list;
- dim_mta;
- dim_campaign;
- dim_traffic_source;

Estas dimensões representam todas as ligações com as *foreign keys* que irão estar presentes na nossa tabela de factos, cobrindo assim, todas as necessidades dos futuros relatórios a serem efetuados.

3.3.4.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 5.

Tabela 5 - Descrição dos factos escolhidos para a tabela de factos fact_email_sent_conversion_daily.

Facto	Tipo	Descrição
revenue	Facto aditivo	Lucro pelas conversões ocorridas na campanha.

3.3.4.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 9.

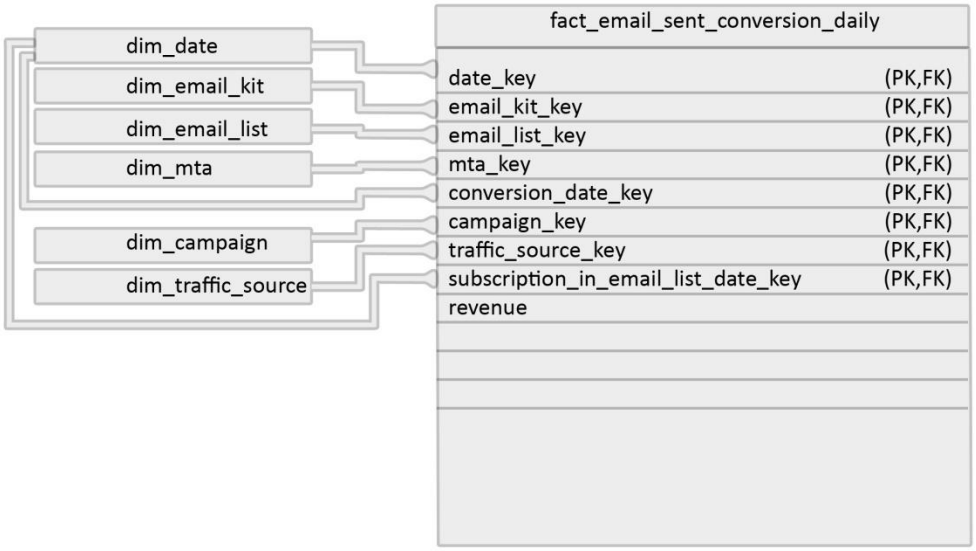

Figura 9: Representação da estrela fact_email_sent_conversion_daily.

3.3.4.5 Volume de entradas diárias na tabela de factos

No que respeita a este subcapítulo iremos proceder às análises relativas ao volume de entradas diárias para a tabela de factos fact_email_sent_conversion_daily.

Atualmente são diariamente recebidas aproximadamente 3 233 conversões por campanha. Como para a tabela de factos em questão apenas necessitamos saber a data de envio do *e-mail,* o *e-mail kit* utilizado, a lista de *e-mail* em que o contacto foi registado, o agente que efetuou o envio, a data de subscrição do utilizador na lista e a fonte de tráfego que angariou o contacto conclui-se que o valor de 3 233 conversões por campanha decresceu para 179 conversões.

Logo, para as conversões recebidas ao longo de um dia temos, 179 * 20 (do total das campanhas existentes este é o valor médio de campanhas que diariamente recebem conversões).

Assim, espera-se um total de 3 580 novos registos por dia.

3.4 Processo de negócio geração de tráfego

Relativamente a este processo de negócio pretende-se conseguir efetuar análises relativas ao redireccionamento de utilizadores para os *microsites*. Estas análises, têm por objetivo observar o comportamento dos utilizadores que chegam aos *microsites* através das várias fontes de tráfego pagas, com a exceção do *e-mail marketing* que foi separado como um processo à parte. Para além da observação dos comportamentos dos utilizadores, pretende-se também conseguir efetuar as respetivas análises de rentabilidade relativas à recolha e

conversão dos utilizadores. De notar, que neste caso é considerada como uma conversão a ação que está estipulada para a campanha, que pode ser considerada a visita do utilizador à página *web* ou para além da visita este tenha que executar uma ou várias ações após a entrada na página *web*, como por exemplo o preenchimento de um formulário, uma compra, visitar duas páginas do *microsite* em menos de 3 minutos, etc.

Em todo o caso, e de forma a conseguirmos atingir o objetivo da análise deste processo de negócio, será criado um *data mart* constituído por três estrelas, sendo estas compostas por várias dimensões. No que respeita às tabelas de factos estas serão denominadas por fact_traffic_tracking, fact_traffic_conversion e a última de fact_traffic_overall.

No âmbito da análise da estrela *traffic tracking* que contém a tabela de factos fact_traffic_tracking encontraremos a informação relativa ao *tracking* dos utilizadores. Isto só é possível graças aos sistemas operacionais de *tracking* registarem um identificador único por utilizador que visita as páginas, tal como explicado no subcapítulo 3.2.1.

No caso de serem *microsites* externos à AdClick o registo será guardado na tabela de factos com a *foreign key* referente ao contacto *unknown* que é o valor predefinido para o caso de não existir submissão dos dados do visitante para o repositório de contactos da AdClick.

No entanto, caso se tratem de *microsites* cuja propriedade pertence à AdClick e exista a submissão da informação do utilizador para o repositório de contactos (LeadCentre) este é guardado com a *foreign key* do contacto angariado. Porém, caso não haja submissão dos dados do utilizador este deve ser guardado com a *foreign key* referente ao contacto *unknown*.

Por sua vez, a tabela de factos fact_traffic_conversion irá registar as conversões dos utilizadores que chegaram pelas diversas fontes de tráfego. Claro está, que para além de registar a data de conversão, esta estrela regista também a respetiva campanha em que ocorreu a conversão.

E por último mas não menos importante, temos a tabela de factos fact_traffic_overall. Esta tabela de factos será construída a partir da junção das duas tabelas de factos anteriormente apresentadas, de forma a existir uma tabela de factos para este *data mart* onde as análises de rentabilidade sejam simples de serem efetuadas, tendo em conta que o custo esta registado na tabela de factos fact_traffic_tracking e a receita na tabela de factos fact_traffic_conversion.

3.4.1 Estrela traffic tracking

Tal como anteriormente descrito esta estrela tem como objetivo registar o *tracking* dos utilizadores. Sendo que o *tracking* irá ser responsável, neste caso, por registar os *microsites* visitados por cada utilizador/contacto.

3.4.1.1 Granularidade

Em relação à granularidade escolhida, esta estrela irá representar a vista num dia, num instante, de um utilizador/contacto, a um *microsite*, que foi direcionado por uma fonte de tráfego através de um anúncio.

Sendo que a chave primária da tabela de factos é constituída pelas *foreign keys* date_key, time_key, user_key, microsite_key, traffic_source_key, tracking_key e country_key.

3.4.1.2 Escolha das dimensões

As dimensões presentes na estrela são:

- dim_date;
- dim_time;
- view_user;
- dim_traffic_source;
- dim_tracking;
- view_microsite;
- dim_country;

Estas dimensões foram escolhidas de acordo com a necessidade de dar uma resposta aos relatórios que se pretendem criar, e com os dados escolhidos para estarem presentes na tabela de factos.

3.4.1.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 6.

Tabela 6 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_tracking.

Facto	Tipo	Descrição
operative_system	facto não-aditivo	Sistema operativo do utilizador/contacto.
browser	facto não-aditivo	*Browser* do utilizador/contacto.
browser_version	facto não-aditivo	Versão do *browser* do utilizador/contacto.
browser_language	facto não-aditivo	Língua do *browser* do utilizador/contacto.
clicks	facto aditivo	Clicks efetuados pelo utilizador/contacto.
cost	facto aditivo	Custo pago pelo redireccionamento do utilizador/contacto para o *microsite*.
cost_per_click	facto não-aditivo	Custo por redireccionamento.

3.4.1.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 10.

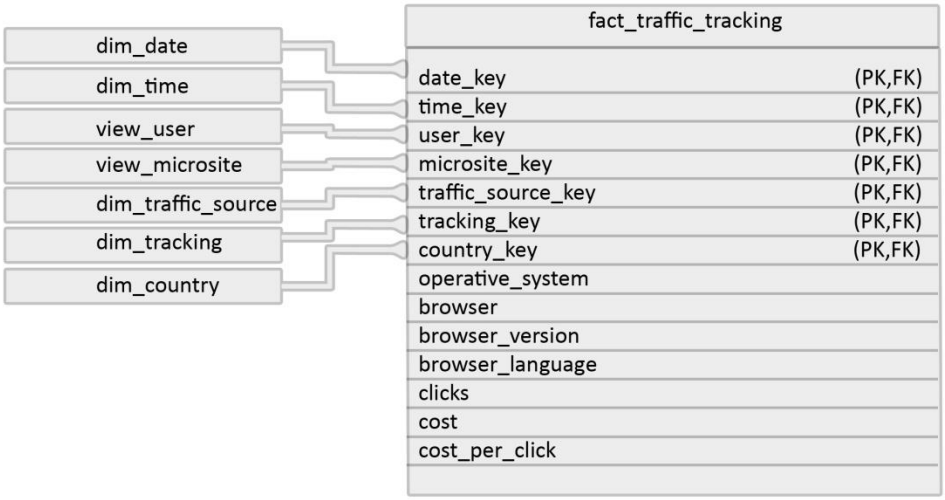

Figura 10: Representação da tabela de factos fact_traffic_tracking e das suas dimensões.

3.4.1.5 Volume de entradas diárias na tabela de factos

Vamos então efetuar as análises relativas ao volume de entradas para a tabela de factos fact_traffic_tracking. Para isso foram efetuadas as devidas consultas aos sistemas operacionais. Neste caso aos sistemas de *tracking*.

Sendo que por minuto (60*24 de forma a termos os valores diários) são registadas em média 78 novas visitas a *microsites* anunciados pela AdClick.

Como tal, as visitas registadas ao longo de um dia serão, 78*60*24, o que perfaz um total de 112 320 novas entradas na tabela de factos por dia.

3.4.2 Estrela traffic conversion

No âmbito da análise desta estrela será efetuada a contabilização das conversões que ocorrem devido ao direcionamento de utilizadores/contactos para um *microsite*, sendo certo que podem ainda estar pendentes de uma ação, tal como explicado no início do subcapítulo 3.4.

3.4.2.1 Granularidade

Relativamente à granularidade, a tabela de factos irá representar as conversões ocorridas num dia num determinado instante. No caso, irá estar representada na tabela de factos

através da chave primária da tabela de factos que é composta pela data (date_key) de visita de um utilizador (user_key), de um País (country_key), num instante (time_key), a um *microsite* (microsite_key), que foi direcionado através de uma fonte de tráfego (traffic_source_key) por um anúncio (tracking_key) e que converteu num dia (conversion_date_key), num instante (conversion_time_key) para uma campanha (campaign_key).

3.4.2.2 Escolha das dimensões

As dimensões presentes na estrela são:

- dim_date;
- dim_time;
- view_user;
- dim_traffic_source;
- dim_tracking;
- view_microsite;
- dim_campaign;
- dim_country;

Estas dimensões representam todas as ligações com as *foreign keys* que irão estar presentes na nossa tabela de factos, cobrindo assim todas as necessidades dos futuros relatórios a serem efetuados.

3.4.2.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 7.

Tabela 7 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_conversion.

Facto	Tipo	Descrição
conversions	Facto aditivo	Total das conversões.
revenue	Facto aditivo	Lucro pela conversão ocorrida.

3.4.2.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que podemos analisar na figura 11.

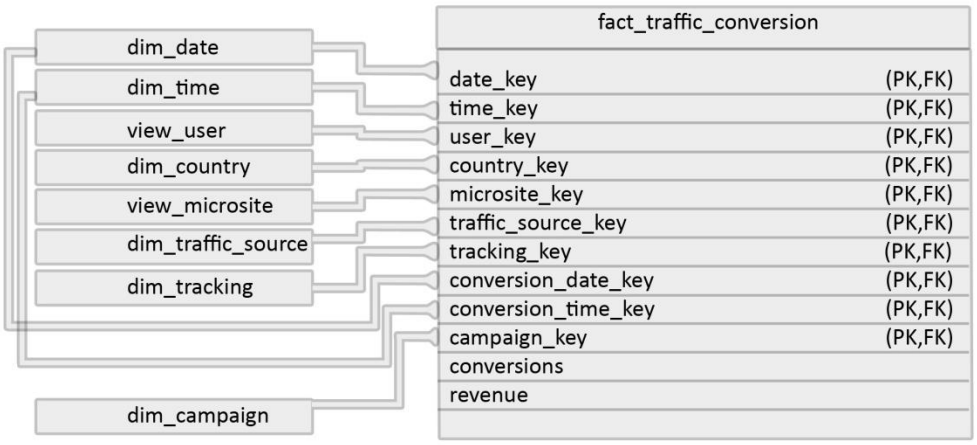

Figura 11: Representação da tabela de factos fact_traffic_conversion e das suas dimensões.

3.4.2.5 Volume de entradas diárias na tabela de factos

Efetuamos então as análises relativas ao volume de entradas diárias para a tabela de factos fact_traffic_conversion. Para que tal aconteça foram executadas as devidas consultas aos sistemas operacionais. Neste caso aos sistemas de *tracking*.

Sendo que, por minuto (60*24 de forma a termos os valores diários) são registadas 78 novas visitas a *microsites* que a AdClick está a anunciar, e cuja taxa de conversão média é de 8.8%.

Logo, podemos deduzir que o total de novos registos por dia que irão dar entrada na tabela de factos é de 78*60*24*0.088, o que perfaz um total de aproximadamente 9 884 novos registos por dia.

3.4.3 Estrela traffic overall

Relativamente à estrela *traffic overall*, temos como principal objetivo a elaboração das respetivas análises de rentabilidade.

Esta estrela contém uma tabela de factos com o nome de fact_traffic_overall que será construída a partir da junção das tabelas de factos fact_traffic_tracking e a fact_traffic_conversion. Com isto, será então possível efetuar diversas análises, tais como a taxa de conversão de um determinado *microsite*, custo por conversão e até mesmo o retorno de investimento por campanha.

De salientar, que nesta tabela de factos os dados estarão agregados ao dia e que diariamente são sempre recalculados os últimos 7 dias. Também ao dia 3 de cada mês é feito o

reprocessamento do mês anterior para esta tabela. Estes processos serão descritos no subcapítulo 4.4 que explica a implementação do *data warehouse* incluindo o carregamento da tabela de factos.

3.4.3.1 Granularidade

Na tabela de factos fact_traffic_overall vamos encontrar a representação dos direcionamentos num dia, por uma fonte de tráfego, para um *microsite* com uma data de conversão e a respetiva campanha onde a conversão ocorreu. De notar, que caso não exista conversão a data de conversão e a campanha irão assumir os valores pré-definidos como *unknown*.

Deste modo podemos então deduzir/encontrar a chave primária que é composta por date_key, microsite_key, traffic_source_key, conversion_date_key, campaign_key.

3.4.3.2 Escolha das dimensões

As dimensões presentes na estrela são:

- dim_date;
- view_microsite;
- dim_traffic_source;
- dim_campaign;

Estas dimensões foram escolhidas de acordo com a necessidade de dar resposta aos relatórios a criar e de acordo com os dados escolhidos que estão presentes nas tabelas de factos.

3.4.3.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 8.

Tabela 8 - Descrição dos factos escolhidos para a tabela de factos fact_traffic_overall.

Facto	Tipo	Descrição
clicks	Facto aditivo	Total de conversões clicks.
conversions	Facto aditivo	Total de conversões.
cost	Facto aditivo	Custo pago pelo redireccionamento para a *webpage*.
revenue	Facto aditivo	Valor recebido pelo redireccionamento para a *webpage* após a conversão.
profit	Facto aditivo	Diferença entre o revenue e o custo. Este cálculo é feito a partir fórmula revenue - cost.

roi	Facto não-aditivo	Retorno de investimento. Este cálculo é feito a partir fórmula profit / cost_total.
cost_per_click	Facto não-aditivo	Custo por click. Este cálculo é feito a partir fórmula revenue / clicks.
revenue_per_click	Facto não-aditivo	Valor recebido por click. Este cálculo é feito a partir fórmula revenue / clicks.
cost_per_conversion	Facto não-aditivo	Custo por conversão. Este cálculo é feito a partir fórmula cost / conversions.
revenue_per_conversion	Facto não-aditivo	Valor recebido por conversão. Este cálculo é feito a partir fórmula revenue / conversions .
conversion_per_click	Facto não-aditivo	Número de clicks necessários para que exista uma conversão. Este cálculo é feito a partir fórmula conversions / clicks.

3.4.3.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 12.

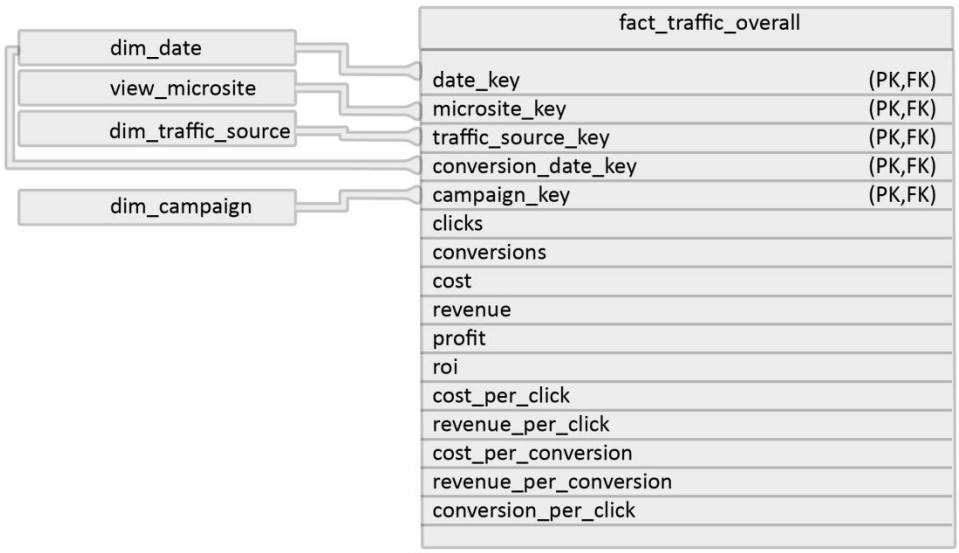

Figura 12: Representação da tabela de factos fact_traffic_overall e das suas dimensões.

3.4.3.5 Volume de entradas diárias na tabela de factos

Vamos então efetuar as análises relativas ao volume de entradas diárias para a tabela de factos fact_traffic_overall. Para isso foram efetuadas as devidas consultas aos sistemas

operacionais, neste caso aos sistemas de *tracking*. Particularmente podemos e devemos utilizar a informação que existe nos subcapítulos 3.4.1.5 e 3.4.2.5. Como esta tabela é construída a partir da junção das tabelas de factos fact_traffic_tracking e fact_traffic_conversion e, como sabemos pela informação do ponto 3.4.2.5 que normalmente uma conversão num *microsite* ocorre apenas para uma campanha. Logo a consulta a efetuar é a mesma da 3.4.1.5 mas agregada de acordo com a chave primária da tabela de factos fact_tracking_overall.

Sendo que o total visitas diárias é de 2 884 novas visitas pelas fontes de tráfego pagas aos *microsites* que a AdClick está a anunciar.

3.5 Processo de negócio venda de contacto

O presente processo de negócio, venda de contactos, tem como missão representar as vendas dos contactos e informação relacionada com a venda de contactos. Relembrando, que para existir uma venda de contacto é necessário que o contacto seja recolhido num *microsite* da AdClick.

Devemos também ter noção que a confirmação ou invalidação das vendas dos contactos é feita através de integrações *server-to-server*, tal como explicado no subcapítulo 3.1.2. Após o envio de um contacto para um cliente através da sua campanha, este pode ser instantaneamente considerado válido ou pode ser invalidado futuramente, o que pode criar a existência de várias respostas para uma integração.

Ainda de salientar que este *data mart* irá ser constituído por duas estrelas. Assim sendo, será uma respetiva às integrações, onde também estarão definidas as respostas do cliente e, outra estrela, que é a representação da primeira mas com um nível de agregação superior, que no caso é ao dia.

3.5.1 Estrela integration

No que respeita à estrela *integration*, esta é constituída por uma tabela de factos central que se chama fact_integration. Esta tabela irá proceder ao registo de todas as integrações ocorridas assim como de todas as respostas enviadas pelos clientes.

3.5.1.1 Granularidade

A granularidade desta estrela representa o envio numa data, num instante, de um contacto, para uma campanha, com uma resposta, num dia e num instante.

Sendo certo que a chave primária da tabela de factos, fact_integration, é constituída pelas *foreign keys* date_key, time_key, user_key, campaign_key, integration_date_key, integration_time_key e integration_response_key.

3.5.1.2 Escolha das dimensões

As dimensões presentes na estrela são:

- dim_date;
- dim_time;
- dim_campaign;
- view_user;
- dim_integration_response;
- dim_traffic_source;
- view_microsite;
- dim_country;

Estas dimensões representam todas as ligações com as *foreign keys* que marcam presença na nossa tabela de factos, cobrindo assim todas as necessidades dos futuros relatórios a ser efetuados.

3.5.1.3 Atributos extra

De notar que para simplificar o acesso aos dados foram ainda registados na tabela de factos alguns atributos que se encontram nas dimensões que ligam a tabela de factos e que são o user_microsite_key e user_traffic_source_key.

3.5.1.4 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 9.

Tabela 9 - Descrição dos factos escolhidos para a tabela de factos fact_integration.

Facto	Tipo	Descrição
cost	Facto aditivo	Custo pago pela recolha do contacto.
revenue	Facto aditivo	Valor recebido pela integração do contacto.
revenue_invalidated	Facto aditivo	O valor para anular o valor recebido pela integração do contacto.
profit	Facto aditivo	Diferença entre o valor recebido e o custo. Este cálculo é feito a partir fórmula revenue - cost.
roi	Facto não-aditivo	Retorno de investimento.

		Este cálculo é feito a partir fórmula profit / cost.
integrations	Facto aditivo	Total de tentativas de integração.
integrations_accepeted	Facto aditivo	Total de integrações com sucesso.
integrations_lost	Facto aditivo	Total de integrações perdidas.
integrations_rejected	Facto aditivo	Total de integrações rejeitadas.
integrations_invalidated	Facto aditivo	Total de integrações invalidadas.

3.5.1.5 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 13.

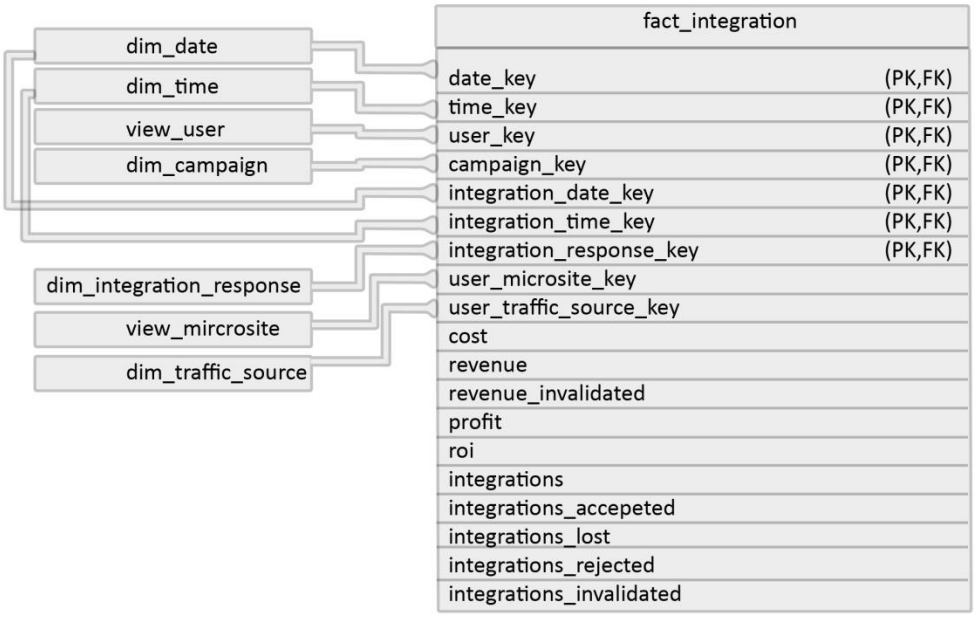

Figura 13: Representação da tabela de factos fact_integration e das suas dimensões.

3.5.1.6 Volume de entradas diárias na tabela de factos

Vamos então efetuar as análises relativas ao volume de entradas diárias para a tabela de factos fact_integration. Para isso foram efetuadas as devidas consultas aos sistemas operacionais.

Sendo que atualmente, por minuto (60*24 de forma a termos os valores diários) são recebidos aproximadamente 6 contactos os quais são integrados com aproximadamente 4 campanhas. O que perfaz um total de 34 560 contactos integrados por dia. De notar que neste caso, a resposta é síncrona pois o contacto é enviado e existe logo uma resposta por parte do cliente.

No entanto temos que tomar em consideração as respostas enviadas pelos clientes quando pretendem cancelar uma integração ou enviar informação adicional após a receção da integração. Estes casos atualmente são apenas verificados num universo de 13% do total das integrações efetuadas o que levará a um incremento no total de registos.

Como tal, as integrações registadas ao longo de um dia são, 6*4*60*24*1.13, o que perfaz um total de 39 053 registos por dia.

3.5.2 Estrela integration overall

No que respeita à estrela *integration overall* que será aqui representada, esta tem como objetivo principal representar a informação sobre as integrações ocorridas, de forma mais agregada em relação aos factos ocorridos diariamente. Como este é o nosso objetivo, algumas dimensões deixaram de estar presentes como iremos explicar em seguida.

3.5.2.1 Granularidade

A granularidade presente na tabela de factos fact_integration_overall é referente às integrações ocorridas numa data, de contactos recolhidos para uma campanha, através de uma fonte de tráfego, num *microsite*.

O que é representado pela chave primária da tabela de factos como date_key, campaign_key, traffic_source_key, microsite_key.

3.5.2.2 Escolha das dimensões

As dimensões presentes na estrela são:

- dim_date;
- dim_campaign;
- dim_traffic_source;
- view_microsite;

Estas foram escolhidas de acordo com a necessidade de responder aos relatórios que se pretendem criar e com os dados escolhidos para estarem presentes na tabela de factos.

3.5.2.3 Escolha dos factos

As métricas escolhidas para esta tabela de factos são as que se encontram descritas na tabela 10.

Tabela 10 - Descrição dos factos escolhidos para a tabela de factos fact_integration_overall.

Facto	Tipo	Descrição
cost	facto aditivo	Custo pago pela recolha do contacto.
revenue	facto aditivo	O valor recebido pela integração do contacto com a campanha do cliente.
revenue_invalidated	facto aditivo	O valor para anular o valor recebido pela integração efetuada.
profit	facto aditivo	Diferença entre o valor recebido e o custo. Este cálculo é feito a partir fórmula revenue - cost.
roi	facto não-aditivo	Retorno de investimento. Este cálculo é feito a partir fórmula profit / cost.
integrations	facto aditivo	Total de tentativas de integração.
integrations_accepeted	facto aditivo	Total de integrações com sucesso.
integrations_lost	facto aditivo	Total de integrações perdidas.
integrations_rejected	facto aditivo	Total de integrações rejeitadas.
integrations_invalidated	facto aditivo	Total de integrações invalidadas.

3.5.2.4 Esquema apresentado

Após a definição das dimensões e dos factos, foi elaborada a representação do esquema que pode ser analisado na figura 14.

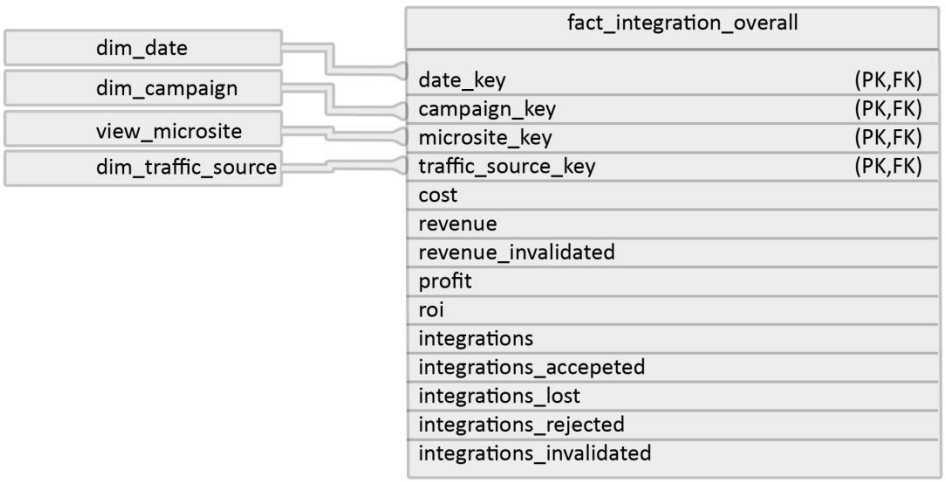

Figura 14: Representação da tabela de factos fact_integration_overall e das suas dimensões.

3.5.2.5 Volume de entradas diárias na tabela de factos

Vamos então efetuar as análises relativas ao volume de entradas diárias para a tabela de factos fact_integration_overall. Neste sentido, reutilizou-se as consultas que foram efetuadas aos sistemas operacionais no subcapítulo 3.5.1.6 e adaptou-se as mesmas à nossa nova

granularidade, que é definida pela chave primária da tabela de factos fact_integration_overall.

Concluindo-se através de várias análises das quais foi feita uma média que o total de novas entradas diárias na tabela de factos fat_integration_overall por dia é de 28 321.

3.6 Cálculo do tamanho do *data warehouse*

Neste ponto iremos elaborar o cálculo do volume esperado de dados de cada tabela de factos que foram anteriormente apresentadas.

Relativamente às dimensões, vamos estabelecer que estas ocupam 10% do volume total do *data warehouse* [Oliveira, P., 2013]. Sendo certo que as dimensões são todas *conformed dimensions*.

Assim sendo, foi então construída uma matriz em bus (tabela 11), foi também feito um levantamento do espaço ocupado por cada tipo de dados (tabela 12) [Oracle, & affiliates, 2013a] e por fim analisadas as tabelas de factos (tabela 13). Isto para ajudar no processo de cálculo da estimativa do volume do *data warehouse* representado na tabela 14.

Tabela 11 - Matriz em bus com a representação de todos os processos de negócio.

Data mart (Processo de negócio)	**Estrela**	**Dimensão**	**dim_date**	**dim_time**	**dim_country**	**dim_campaign**	**view_user**	**dim_email_kit**	**dim_email_list**	**dim_mta**	**dim_traffic_source**	**dim_tracking**	**view_microsite**	**dim_integration_response**
Email Marketing	Email sent		x	x	x		x	x	X	x	x			
	Email sent daily		x		x			x	X	x	x			
	Email sent conversion		x	x	x	x	x	x	X	x	x			
	Email sent conversion daily		x		x	x		x	X	x	x			
Geração de tráfego	Traffic tracking		x	x	x		x				x	x	x	
	Traffic conversion		x	x	x	x	x				x	x	x	
	Traffic overall		x			x					x		x	
Venda de contactos	Integration		x	x		x	x				x		x	x
	Integration overall		x			x					x		x	

Tabela 12 - Espaço ocupado por cada tipo de dados em MySql.

Tipo de dados	Espaço necessário
varchar(n)	L+1, se 0 < L <= 255 bytes onde L representa o tamanho em bytes da *string*.
tinyint	1 byte
integer	4 bytes
datetime	5 bytes
timestamp	4 bytes

De seguida iremos ver o número de registos atributos existentes em cada tabela de factos (tabela 13).

Tabela 13 - Composição dos atributos da tabela de factos.

Tipo de dados Tabela de Factos	VARCHAR(N) N+1 bytes	INT (10) 4 bytes	FLOAT(20) 4 bytes	Total em bytes
fact_email_sent (22)		16*4	6*4	88
fact_email_sent_daily (21)		13*4	8*4	84
fact_email_sent_conversion (13)		12*4	1*4	52
fact_email_sent_conversion_daily (9)		8*4	1*4	36
fact_traffic_tracking (14)	4*21	8*4	2*4	124
fact_traffic_conversion (12)		11*4	1*4	48
fact_traffic_overall (16)		7*4	9*4	64
fact_integration (19)		14*4	5*4	76
fact_integration_overall (14)		9*4	5*4	56

Foi então efetuada a estimativa do volume de dados esperado para 1 ano (estimativa de registos diários * 365 * tamanho do tuplo em bytes) e 5 anos (estimativa de registos diários * 1825 * tamanho do tuplo em bytes) que pode ser analisado na tabela 14.

Tabela 14 - Resumo do volume de dados para 1 e 5 anos.

Tabela de Factos	Estimativa de registos diários	Tamanho do tuplo em bytes	Volume esperado para 1 ano em GB	Volume esperado para 5 anos em GB
fact_email_sent	1 257 824	88	37.63	188.13
fact_email_sent_daily	419 274	84	11.97	59.86
fact_email_sent_conversion	64 660	52	1.14	5.71
fact_email_sent_conversion_daily	3 580	36	0.04	0.22
fact_traffic_tracking	112 320	124	4.73	23.67
fact_traffic_conversion	9 884	48	0.16	0.81
fact_traffic_overall	2 784	64	0.06	0.30
fact_integration	39 053	76	1.01	5.04
fact_integration_overall	28 321	56	0.54	2.70
Total			**57.29**	**286.45**

Pelo cálculo do volume de dados esperado para o *data warehouse* efetuado na tabela 14 verificamos que existem duas tabelas de factos (fact_email_sent e fact_email_sent_daily) com um volume grande de dados para o intervalo de 1 ano, e 3 tabelas de factos com um volume de dados relativamente grande para 5 anos. No entanto para 5 anos à exceção das tabelas fact_email_sent e fact_email_sent_daily podemos considerar um volume de dados normal, pois o principal objetivo do *data warehouse* é permitir uma análise histórica e como sabemos um histórico de dados tende sempre a ocupar um elevado volume de espaço.

De forma a mitigar o elevado crescimento da tabela de factos fact_email_sent esta pode apenas conter os dados relativos ao último mês, pois a mesma informação está presente na tabela de factos fact_email_sent_daily mas com um nível de agregação diário.

Por sua vez, uma possível forma de controlar o crescimento da tabela fact_email_sent_daily é criar uma nova tabela com uma agregação dos dados relativa ao mês tendo por base a tabela de factos fact_email_sent_daily. Onde este processo seria idêntico ao que acontece atualmente com a tabela fact_email_sent e fact_email_sent_daily. Após a criação desta tabela a fact_email_sent_daily passaria apenas a conter a informação relativa aos últimos 3 meses.

4 Implementação do *data warehouse*

Este é o ponto crucial no desenvolvimento do *data warehouse*, pois é aqui que passamos da conceção dos modelos à prática. Todo o processo de ETL foi desenvolvido de raiz, sem o recurso a qualquer tipo de *software* já existente. A base de dados foi implementada em MySQL. A linguagem de programação utilizada foi o PHP. Foi também utilizado como recurso a *framework* Symfony 2 [Symfony, 2013]. Esta *framework* gere as conexões com as bases de dados assim como dispõe de serviços que podem ser lançados através da linha de comandos que são identificados como tarefas (*tasks)*, e que serão usadas para todo o processo de ETL.

Uma *task* pode ser responsável por vários processos de extração ou de carregamento dos dados (opções). Para o efeito, irão ser criadas cinco *tasks* para a *staging area* que são responsáveis pela extração e transformação dos dados. Estas assumir-se-ão como, YellowCommand, AdSniperCommand, AfileaCommand, LeadCentreCommand e HurryUpCommand. A par destas serão criadas duas *tasks* adicionais que são responsáveis pelo carregamento dos dados para o *data warehouse*, que se denominam por LoadDimCommand e LoadFactCommand.

Foi também criada uma *task* para gerir as *views* que se denomina por LoadViewCommand.

Ressalve-se que as *tasks* partilham algumas opções transversais que podem ser visualizadas na tabela 15.

De notar que a importância das *tasks* prende-se com estes processos poderem ser executados através da linha de comandos. Como tal é possível através do *crontab,* que é um serviço do sistema operativo Unix que permite o agendamento de comandos, e que é utilizado para os processos de ETL serem lançados uma ou mais vezes por dia dependendo da taxa de atualização que se pretende manter no nosso *data warehouse*.

Tabela 15 - Opções disponíveis em todas as *tasks*.

Tabela de Factos	Descrição
date_start	Data e hora mínima a partir da qual se pretende que os registos sejam extraídos. (opcional)
date_end	Data-hora máxima para a qual se pretende que os registos sejam extraídos. (opcional)
start_id	Identificador do registo a partir do qual se pretende extrair os dados. (opcional)

No caso de haver omissão das opções date_start ou start_id, os valores assumidos (por omissão) são os que se encontram na tabela controller_task. Já no caso de não existir nenhuma entrada na tabela controller_task referentes a última extração ou carregamento, o valor assumido para a opção date_start é de "2013-12-01 00:00:00" e para o start_id é "0". Relativamente à opção date_end, o valor que esta assume corresponde à data e hora corrente no servidor, no caso das *tasks* serem de importação de dados para a *staging area* ou de carregamento de factos.

Assim, começamos por efetuar a extração e transformação dos dados que existem nos sistemas operacionais para a *staging area*.

Após a extração é feita a transformação dos dados de forma e estes estarem prontos a serem carregados para o *data warehouse* sendo que no final deste processo os mesmos são guardados na *staging area*. De notar, que todos os dados que mantenham um conjunto de respostas finitas como por exemplo os valores booleanos ou atributos apenas com dois valores possíveis (Sim/Não;Homem/Mulher) ou datas (Janeiro/Fevereiro…) são traduzidos para a língua Inglesa.

Após o processo de extração e transformação segue-se o carregamento das dimensões e dos dados para as tabelas de factos do *data warehouse*.

Sendo que todos os processos irão ser abordados nos subcapítulos seguintes do presente capítulo.

4.1 Extração e transformação dos dados

Comecemos por referir que a extração dos dados dos sistemas operacionais ocorre várias vezes por dia, 30 minutos em 30 minutos, devido ao elevado volume de dados que é recolhido. Para que apenas ocorra a extração dos dados que foram atualizados nos sistemas operacionais foi criada uma tabela controller_task na *staging area* (figura 15), que guarda a informação relativa ao processo usado (nome da *task* e do processo), a data e hora em que o processo concluiu a extração ou o carregamento e o identificador do último registo extraído. Isto para permitir que os processos usados apenas extraiam a informação que foi inserida ou os registos que foram atualizados.

De notar que a extração dos registos extraídos tem de configurar uma de duas situações:

- no caso de existirem atributos do tipo *timestamp* nas tabelas em que os dados estão a ser extraídos, guarda-se a data e o instante da última atualização do conjunto de registos extraídos;

- no caso da importação ser sequencial, guarda-se o último identificador (id) do registo extraído;

Em todo o caso, o tipo de extração utilizada é o que for definido no processo de atualização que na maioria dos casos coincide com a primeira das duas situações anteriormente apresentadas.

Figura 15: Tabela controller_task responsável por guardar a informação sobre as extrações e carregamentos.

Em seguida serão analisados os processos de extração relativamente aos sistemas operacionais que irão abastecer os dados da *staging area*.

4.1.1 *Task* YellowCommand

Tal como anteriormente descrito, é no sistema Yellow (subcapítulo 3.2.3) que se encontra armazenada toda a informação relativa ao envio de *e-mails*. Assim sendo, foi criada uma *task* responsável pela extração da informação que aí consta. Sendo que nesta *task* se encontram definidas várias opções que têm a responsabilidade de extrair os dados para tabelas específicas da *staging area*.

De notar que inicialmente foi efetuada uma consulta ao sistema operacional a fim de identificar que tipo de extração podem ser usadas nas opções. Sendo que foi identificado que em todas as tabelas do sistema de onde vão ser extraídos os dados contêm o atributo updated_at que guarda a data da última atualização do registo. Logo, para todas as opções o processo a ser usado após a primeira extração é o de apenas extrair os novos registos ou aqueles que foram atualizados desde a última extração.

Também é de ter em consideração que o meio de extração dos dados é efetuado através de um pedido ao sistema operacional Yellow através de um URL onde é possível passar o parâmetro last_updated de forma a se obter apenas os dados que foram atualizados a partir dessa data. Sendo que a resposta ao pedido é sempre uma resposta no formato para a troca de dados JSON [JSON, 2013]).

Iremos então analisar algumas das opções que podem ser invocadas para lançar os processos de extração dos dados.

4.1.1.1 Opções mta, email_kit e email_list

Neste ponto iremos apenas analisar a opção mta isto porque as opções email_kit e email_list são processos de extração de dados muito semelhantes a que vamos análisar.

Como anteriormente verificado a extração dos dados é efetuada através de um pedido ao sistema operacional Yellow através de um URL e obtém-se uma resposta no formato JSON (figura 16) contendo a informação relativa aos MTA.


```
{
  statistics: {
    results:
    [
      {
        mta_id: "1",
        name: "Example 1",
        ip: "192.168.1.1",
        return_path: "Yes",
        updated_at: "2013-05-20 15:52:84"
      },
      {
        mta_id: "2",
        name: "Example 22",
        ip: "192.168.1.12",
        return_path: "Yes",
        updated_at: "2013-07-02 15:52:84"
      }
    ]
  },
  success: "1",
  message: "Data Found"
}
```


Figura 16: Exemplo do JSON retornado com a informação do MTA.

Sendo certo que após a validação da informação recebida esta é armazenada na tabela mta da *staging area* (figura 17).

Figura 17: Tabela mta existente na *staging area*.

4.1.1.2 Opção email_sent

Tal como foi referido no subcapítulo anterior, esta opção da *task* YellowCommand tem como objetivo recolher a informação que se encontra no sistema operacional Yellow, e no caso da opção email_sent, mais concretamente, a informação dos envios de *e-mails* efetuados.

Para que tal objetivo seja atingido, é efetuado um pedido através de um URL ao sistema operacional, onde se obtém como resposta um JSON (figura 18). Sendo que após a receção, é validado o conteúdo da informação recebida e caso existam campos enviados com o valor *null* ou que estejam vazios, estes assumem o valor pré-definido de *unknown* ou 0.


```
{
  statistics: {
    results:
    [
      {
        email_sent_id: "154666",
        date: "2013-06-05",
        kit_id: "1",list_id: "1",mta_id: "1",
        list_subscription_date: "2013-01-01",
        email: "albuquerque.joao.filipe@gmail.com",
        cost: "1.52",
        subid: "3125y85888",
        send_total: "1",open_total: "22",
        click_total: "10",click_unique: "1",
        unsubcribed: "0",soft: "0",hard: "0",
        updated_at: "2013-05-20 15:52:84"
      },
      {...}
    ]
  },
  success: "1",
  message: "Data Found"
}
```


Figura 18: Exemplo do JSON retornado com a informação acerca do *e-mail* enviado.

Após a validação, a informação é guardada na tabela email_send na *staging area*.

4.1.2 *Task* AdSniperCommand

Relativamente ao processo de importação de dados presentes no sistema de *tracking* AdSniper, vamos extrair a informação relativa aos dados do utilizador quando este visita uma página *web*. Este processo será ativado através da invocação da opção user_tracking da *task* AdSniperCommand.

Também seria possível extrair deste sistema, os dados inerentes às estatísticas das fontes de tráfego. Ou seja, poderíamos obter a informação relativa ao número de *clicks*, número de conversões, entre outros, no entanto, esta informação estaria incompleta, porque não contém o valor pago à fonte de tráfego nem o valor recebido pelo redireccionamento. Assim iremos optar por extrair toda a informação de uma só vez utilizando para isso o sistema operacional HurryUp que já contém todos os valores.

Como tal, nesta *task* (AdSniperCommand) iremos apenas encontrar a opção user_tracking que nos propomos analisar em seguida.

4.1.2.1 Opção user_tracking

No que respeita à opção user_tracking, esta é invocada para se efetuar a recolha de informação relativa à visita de um utilizador. Como este sistema é proprietário da AdClick e não existe nenhuma API definida para se efetuar o pedido das estatísticas, é necessário que seja feito o acesso direto à base de dados do sistema AdSniper.

Assim, é recolhida a informação que consta na base de dados (adsniper) do sistema através de uma consulta direta às tabelas, log_redirect e log_conversions. De notar que em ambas as tabelas está presente o atributo updated_at que guarda a data da última atualização como tal iremos tirar partido do mesmo para efetuar as extrações. Após a extração dos dados, é o momento de se efetuar a validação que os dados extraídos se encontram todos preenchidos e não existem valores vazios caso contrário iremos definir os mesmos com o valor de 0 ou "Unknown". Sendo que após esta verificação os mesmos são armazenados em duas tabelas da *staging* a*rea* nomeadamente user_adsniper_tracking_redirect para os dados extraídos da tabela log_redirect e user_adsniper_tracking_conversion para os dados extraídos da tabela log_conversion.

Ressalve-se que este processo é feito com a extração dos dados das duas tabelas (log_redirect e log_conversion) devido ao elevado número de registos que estas contêm.

4.1.3 *Task* importAfilea

Relativamente ao sistema Afilea este permite efetuar o serviço de *tracking* de utilizadores, disponibilizar campanhas para outras entidades gerarem tráfego e, ainda, gerar tráfego para campanhas de outras entidades. Apesar de podermos retirar informação relativa às campanhas, fontes de tráfego, estatísticas das campanhas, aqui, apenas iremos retirar os

dados relativos aos visitantes das páginas *web*. Isto sucede porque, tal como foi referido anteriormente, existe o sistema operacional HurryUp que já contém toda a informação restante e que permite configurações mais avançadas relativamente às campanhas e fontes de tráfego. Neste sentido, é nossa opção efetuar apenas a exportação dos dados relativos aos visitantes utilizando a opção user_tracking.

4.1.3.1 Opção user_tracking

De forma a efetuarmos a exportação dos dados do sistema Afilea é feito um pedido através de um URL à API do mesmo. Como resposta, é enviado um JSON com a informação relativa a cada utilizador (figura 19).


```
{
  request: {...},
  response: {status: 1,httpStatus: 200,data:
    {
    page: 1,current: 50,count: 1192,pageCount: 24, data: [
      {
        Stat:{ datetime: "2012-12-01 00:00:22",
            ad_id: "1024ea51320998edd320e93f7c32d3",
            advertiser_id: "18",goal_id: "0",
            refer: "http://www.bing.com/search?q=tarotistas en Oviedo&go=&qs=bs&form=QBRE",
            pixel_refer: "http://videncias-espana.com/esmeraldavidente/p4a.php",
            user_agent: "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
            browser_id: "2",
            affiliate_info1: "",affiliate_info2: "",affiliate_info3: "",affiliate_info4: "",affiliate_info5: ""
        },
        Browser: {display_name: "Firefox"}
      },{...}
    ],
    dbSource: "branddb"
  },
  errors: [ ],
  errorMessage: null
  }
}
```


Figura 19: Exemplo do JSON retornado com a informação acerca do utilizador.

Sendo que após a receção da informação esta é validada de forma a não ser armazenada informação com erros na tabela user_tracking da *staging area*.

4.1.4 *Task* LeadCentreCommand

No que se refere à importação dos dados existentes no sistema LeadCentre, estes serão relativos aos contactos recolhidos bem como às integrações que foram efetuadas dos contactos com as campanhas dos clientes.

No decorrer desta análise, foi encontrado um problema relacionado com o facto de a informação recolhida acerca do contacto ser dinâmica. Este problema será melhor abordado e explicado no carregamento da dimensão contacto (dim_user) para o *data warehouse*. Por ora apenas necessitamos saber que a informação relativa ao contacto se encontra em duas

tabelas, numa das tabelas constam atributos que todos os contactos devem recolher, mas não obrigatórios, e na outra consta informação adicional sobre o contacto.

Tal como acontece à informação extraída do sistema AdSniper e Afilea, no LeadCentre também se encontram campanhas e dados relativos aos clientes. Contudo, esta mesma informação é redundante por comparação com a mesma que se pode encontrar no sistema HurryUp. Devido à informação sobre as campanhas que se encontra definida no sistema HurryUp conter um maior nível de detalhe, optamos por apenas extrair a informação que se encontra no sistema HurryUp.

Encontra-se, deste modo, assente que a informação a retirar do LeadCentre é apenas relativa aos contactos, detalhes extra dos contactos, definição dos campos extra dos contactos e as integrações ocorridas.

Para ser possível a extração desta informação foi criada uma *task* com o nome LeadCentreCommand, em que estão definidas as opções user, user_extra, user_extra_attribute e integration, que pela mesma ordem correspondem aos contactos, detalhes extra dos contactos, nome dos campos extra dos contactos e as integrações.

Assim, cumpre referir que todo o processo de extração ocorre através da chamada da API do LeadCentre. Como resposta é devolvido um ficheiro CSV com a informação.

4.1.4.1 Opção user

A presente opção consta da *task* LeadCentreCommand e foi definida com o intuito de extrair o contacto armazenado no sistema LeadCentre. Porém, é importante ressalvar que neste sistema os contactos nunca são atualizados assim, o tipo de extração a ser usada nesta opção é a extração sequencial de acordo com o identificador (id) do contacto do sistema.

Após a extração dos dados, é necessário efetuar as devidas validações para que os mesmos possam ser guardados na tabela user da *staging area* (figura 20).

Figura 20: Tabela user presente na *staging area*.

4.1.4.2 Opções user_extra

Na opção, user_extra, tal como acontece na opção anterior, user, é do tipo sequencial resultado dos motivos anteriormente mencionados. Também aqui é efetuada a consulta dos dados que se encontram no sistema LeadCentre e após a recolha e validação dos mesmos, estes são guardados na tabela user_extra da *staging area*.

4.1.4.3 Opção user_extra_attribute

Relativamente à opção user_extra_attribute, esta existe para recolher o nome dos campos extra (dinâmicos) que existem para a recolha de informação adicional sobre o contacto. Para esta extração é necessário efetuar a extração de todos os dados da tabela porque a mesma não contém nenhum atributo com a data da última atualização dos registos.

4.1.4.4 Opção integration

No que respeita a esta opção cumpre referir que a mesma é responsável pela extração das integrações que ocorreram entre os contactos existentes no repositório LeadCentre e as campanhas ativas. Tal como referido anteriormente, as integrações são compostas pelo envio dos contactos para as campanhas dos clientes. Ou seja, o cliente recebe o envio e retorna uma resposta, sendo a mesma guardada no LeadCentre e feita a respetiva correspondência para um dos quatro pares possíveis:

- não repetido (*NOT REPEATED*) e integrado (INTEGRATED);
- não repetido (*NOT REPEATED*) e não integrado (NOT INTEGRATED);
- repetido (*REPEATED*) e integrado (INTEGRATED);
- repetido (*REPEATED*) e não integrado (NOT INTEGRATED);

De salientar que, o conjunto que define uma integração com sucesso, é composto pelo par não repetido e integrado.

Para além do conjunto de respostas definidas no primeiro envio o cliente pode facultar informação sobre a integração *a posteriori*. Sendo que, nesses envios deve constar o identificador do contacto enviado, assim como, a informação que o cliente pretende transmitir à AdClick. A informação enviada é guardada e mapeada com um dos seguintes campos:

- Inicio das conversações com o contacto (*OPENED*);
- Contacto inválido (*INVALID*);
- Contacto não interessado na campanha (*NOT INTERESTED*);
- Contacto converteu na campanha (*CONVERTED*);
- Contacto cessou as conversações (*CLOSED*);

De notar que no caso da resposta ser mapeada com o termo contacto inválido, a integração deve ser considerada invalidada.

Tal como nas opções anteriores, a forma de extrair a informação acontece através da invocação de um URL que envia como resposta um ficheiro separado por vírgulas (CSV). Após a receção deste último, a informação é devidamente tratada e guardada na tabela integration (figura 21) da *staging area*. Ainda de salientar que o tratamento da informação na extração visa a existência de valores coerentes, não vazios ou nulos e pretende acelerar todo o processo de carregamento de dados para as tabelas de factos que poderá ser visto no subcapítulo 4.4.

Figura 21: Tabela integration presente na *staging area*.

4.1.5 *Task* HurryUpCommand

Como foi já anteriormente referido, o HurryUp funciona como um sistema geral que guarda todas as definições que se encontram espalhadas pelos outros sistemas. Para além da funcionalidade de armazenamento, este permite que estas (definições) possam ser alteradas de uma forma centralizada e permite ainda, que os dados tenham um maior nível de detalhe. Este facto ocorre principalmente quando os sistemas são externos à AdClick e as configurações que são possíveis efetuar são de um âmbito limitado.

Assim sendo, no que diz respeito aos dados recolhidos pelo sistema HurryUp, estes serão relativos aos clientes, contas e campanhas. Igualmente iremos proceder à recolha das informações relativas às fontes de tráfego, páginas *web, microsites* e *websites* existentes.

Para além dessa informação é também aqui que se encontram os dados relativos às conversões que ocorreram, como tal esta informação também será recolhida.

A recolha da informação ocorre através de consultas diretas à base de dados do sistema HurryUp, onde o nome da base de dados é o nome do próprio sistema mas com todas as letras em minúsculas (hurryup).

De seguida iremos analisar cada um destes cinco processos de extração que são despoletados pela invocação das opções traffic_source, campaign, webpage, stat_traffic e stat_email_conversion na *task* HurryUpCommand.

4.1.5.1 Opção traffic_source

Iniciamos a análise pela extração da informação quanto às fontes de tráfego. Cumpre referir que esta informação encontra-se na tabela traffic_sources da base de dados hurryup. De forma a cumprir o proposto, efetuamos então a extração dos dados e a devida validação dos mesmos para que, desta forma, não existam valores sem informação na *staging area*. Após a validação dos dados, os mesmos são guardados na tabela traffic_source (figura 22) que consta na *staging area*.

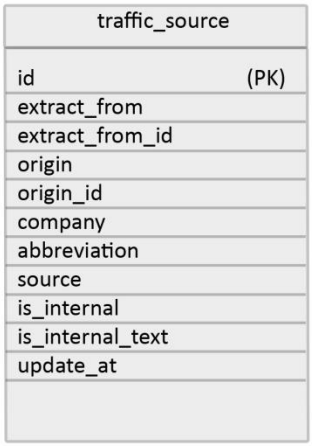

Figura 22: Tabela traffic_source presente na *staging area*.

4.1.5.2 Opções campaign e webpage

Relativamente às opções campaign e webpage, que podem ser encontradas na *task* HurryUpCommand, estas são bastante semelhantes devido ao facto de em ambos os casos os dados destas serem constituídos por múltiplas tabelas na base de dados hurryup.

Assim sendo, iremos apenas aqui abordar o caso mais complexo, que é o da opção campaign.

No que concerne a esta opção, campaign, convém ressalvar que apesar do seu nome são extraídos vários dados, tais como, campanhas dos clientes, contas dos clientes, e dados dos clientes.

Quanto aos dados dos clientes, e tal como o nome indica representa a informação referente ao cliente (tabela client da BD hurryup).

Por sua vez, as contas dos clientes disponibilizam informação referente a um cliente num determinado País, por exemplo o cliente ABC está presente em vários países, Portugal, Espanha e França, e no caso tem vários dados para contacto assim como moradas dependendo do país. Sendo certo que esta informação se encontra na tabela client_account na base de dados hurryup.

Relativamente às campanhas dos clientes, estas encontram-se associadas às contas dos clientes. Cumpre salientar que um cliente pode ter várias contas e várias contas podem ter a mesma campanha ou distintas, tal como confirma a figura 23 que contém a representação visual do exemplo.

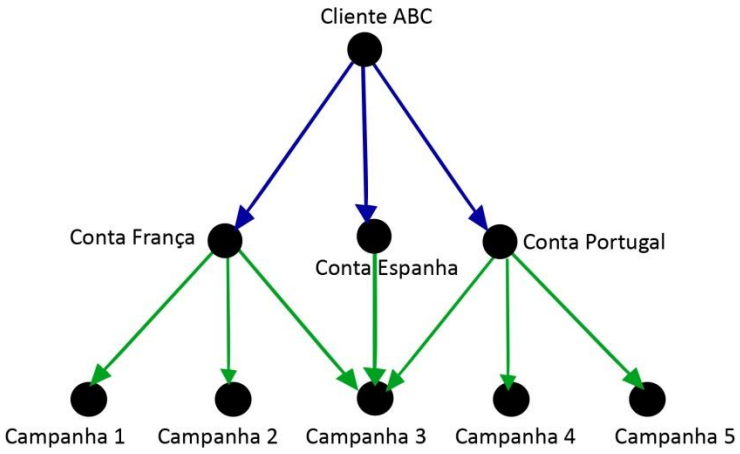

Figura 23: Representação em forma de grafo das relações Cliente - Conta - Campanha.

Após a compreensão do modelo onde estão presentes os dados, é feita a respetiva extração, validação e armazenamento dos dados na tabela campaign da *staging area*.

De notar que a extração é feita sobre todos os registos existentes devido a uma das tabelas usada para encontrar a relação entre as campanhas e as contas não conter o atributo updated_at com a data da última atualização.

4.1.5.3 Opções stat_traffic e stat_email_conversion

Tal como no subcapítulo anterior, neste ponto também encontramos um elevado grau de similaridade entre as opções stat_traffic e stat_email_conversion. Este grau de similaridade está presente porque ambas as opções têm como função extrair os dados respetivos das conversões que ocorreram, sendo que num caso se refere às conversões dos redireccionamentos para os *microsites* e noutro caso é relativo às conversões dos *e-mails* enviados. Devido a este facto, iremos apenas apresentar a opção stat_email_conversion que diz respeito às conversões registadas pelo envio dos *e-mails*.

A opção stat_email_conversion é responsável pela extração dos dados relativos às conversões que ocorreram após o envio dos *e-mails*. No entanto e devido ao facto de por vezes as conversões serem anuladas (isto é os registos serem removidos da base de dados do sistema operacional) ao longo dos primeiros sete dias, este processo corre de acordo com a data de conversão.

Deste modo, sempre que o processo é lançado, o primeiro passo a ser efetuado é a remoção dos dados na tabela stat_email_conversion da *staging area* relativos às datas (data de

conversão) que vão ser extraídos. Após a fase de extração, os dados são devidamente validados e guardados na tabela stat_email_conversion na *staging area*.

4.2 Carregamento das dimensões

Neste subcapítulo iremos explicar cada dimensão individualmente, assim como os problemas encontrados no carregamento dos dados e, por fim, efetuar a análise ao significado de cada um dos seus atributos.

Iremos igualmente validar os campos que foram indexados e obter a justificação para a ocorrência desta indexação. Cumpre referir que as chaves primárias das tabelas se encontram indexadas, uma vez que é este o conjunto de atributos que será mais vezes requisitado. Como esta indexação está presente em todas as dimensões não será mencionada.

Também iremos ver as hierarquias, caso se apliquem, que estão presentes nas dimensões.

Convém relembrar que as dimensões são tabelas desnormalizadas, normalmente compostas por muitos atributos e poucas linhas, por comparação com a tabela de factos. Este é o local onde devem estar definidos o máximo de atributos possíveis, para uma correta análise sobre os dados, que se irão encontrar na tabela de factos. Quantos mais atributos as dimensões tiverem, mais análises diferentes poderão ser realizadas sobre as tabelas de factos, proporcionando assim diferentes perspetivas sobre os dados.

Saliente-se que após a criação das dimensões, foi inserido um registo em cada uma delas com todos os atributos a *Unknown* excepto os atributos is_current, created_date e expired_date. Esta decisão dá-se pois os dados nos sistemas operacionais podem estar incompletos e para assim podermos carregar os registos que dependam dessa dimensão.

Ressalva-se ainda, o caso das dimensões dim_date, dim_time e dim_country que apenas serão carregadas uma única vez, após a criação do *data warehouse*, tal como será explicado em seguida.

Como todo o processo de ETL foi criado de raiz, existiu a necessidade de se criar uma classe component onde se encontram definidos vários métodos que são utilizados durante todo o processo de carregamento de dados. Esta classe contém alguns métodos idênticos aos que podemos encontrar no *software* de ETL como por exemplo o método de *LookUp* para encontrar se o registo já se encontra inserido numa tabela.

Cumpre referir que a ordem de carregamento dos dados é a mesma ordem utilizada na apresentação das dimensões, de forma a evitar problemas de dependências entre as várias tabelas.

Vamos então partir para a análise das respetivas dimensões começando pela dim_date.

4.2.1 Dimensão dim_date

A dimensão dim_date foi criada de forma a conter todas as datas susceptíveis de aparecerem nos dados dos sistemas operacionais. Por isso mesmo, esta dimensão contém os dados desde a data 1900-01-01 até 2038-01-01. Neste sentido, foi também inserido um registo com a data 0000-00-00 de forma que se os dados nos sistemas operacionais não estiverem completos e a data seja um campo obrigatório no *data warehouse*, esta assuma o valor do registo com data 0000-00-00. No caso de existirem datas anteriores a 1900-01-01 esta deve ser tomada como o limite mínimo e a data 2038-01-01 como limite máximo. Atenda-se que as datas estão totalmente de acordo com o calendário (gregoriano) em vigor, sendo que o dia 1900-02-29 não existe contrariamente ao que alguns sistemas acreditam.

O carregamento da dimensão dim_date é efetuado apenas na primeira vez, após a criação do *data warehouse* ou quando esta tabela se encontra vazia.

Os atributos que compõem esta dimensão podem ser analisados na tabela seguinte.

Tabela 16 - Descrição dos atributos da dimensão dim_date.

Atributo	Descrição
date_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado em cada inserção de um novo registo.
fulldate	Representação da data no formato Ano-Mês-Dia (ex: 2013-02-25)
day_in_year	Número do dia no ano. Exemplo para a data 2013-02-25 o valor que este campo tome é 56.
day_in_month	Número do dia no mês. Exemplo para a data 2013-02-25 o valor que este campo tome é 25.
day_in_week	Número do dia na semana. De notar que a semana começa ao domingo. Exemplo para a data 2013-02-25 o valor que este campo tome é 2.
day_name	Nome do dia. Exemplo para a data 2013-02-25 o valor que este campo tome é *Monday*.
is_weekend	Valor booleano com o significado se a data faz parte do fim de semana ou não. Exemplo para a data 2013-02-25 o valor que este campo tome é *False*.
week_number	O número de semanas desde o primeiro dia do ano. Exemplo para a data 2013-02-25 o valor que este campo tome é 9.
month_number	O número do mês. Exemplo para a data 2013-02-25 o valor que este campo tome é 2.
month	Nome do mês. Exemplo para a data 2013-02-25 o valor que este campo tome é *February*.
quarter	Descrição do trimestre da data. Exemplo para a data 2013-02-25 o valor que este campo tome é 1st quarter.
semester	Descrição do semestre da data. Exemplo para a data 2013-02-25 o valor que este campo tome é 1st Semester.
year	Ano da data. Exemplo para a data 2013-02-25 o valor que este campo tome é 2013.

No caso desta dimensão foram definidos dois índices, um que é a chave primária (date_key) e o outro que é o atributo fulldate. O índice do atributo fulldate nesta tabela foi criado devido à grande quantidade de consultas que são feitas sobre este atributo. De ressalvar que este é um índice único uma vez que não podem existir datas repetidas.

Na figura 24 podemos visualizar a hierarquia existente na tabela dim_date.

Figura 24: Representação da hierarquia existente na dimensão dim_date.

Ficando assim concluída a apresentação da dimensão dim_date e o seu processo do carregamento.

4.2.2 Dimensão dim_time

A dimensão dim_time foi criada de forma a conter todas as horas, minutos e segundos possíveis de aparecerem nos dados dos sistemas operacionais. Como tal, os registos desta dimensão estão compreendidos no intervalo das 00:00:00 até às 23:59:59.

Como é de fácil compreensão, este intervalo é imutável, logo, o carregamento da dimensão da dim_time apenas poderá ser feito uma vez após a criação do *data warehouse* ou então quando esta tabela se encontrar vazia.

Relativamente aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 17 - Descrição dos atributos da dimensão dim_time.

Atributo	Descrição
time_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo.
fulltime	Representação do tempo no formato Hora:Minuto:Segundo (ex: 23:59:00)
hour	Hora registada. Exemplo para o tempo 23:59:00 o valor que este campo assume é 23.
minute	Minuto registado.

	Exemplo para o tempo 23:59:00 o valor que este campo assume é 59.
second	Segundo registado. Exemplo para o tempo 23:59:00 o valor que este campo assume é 00.
period	Nome do período, este atributo assume uma de três hipóteses que são *Morning*, *Afternoon* e *Evening*. Exemplo para o tempo 23:59:00 o valor que este campo assume é *Evening*.
launch	Valor booleano com o significado se a hora é a de almoço, entre as 12:00:00 até as 13:59:59, ou jantar, entre as 19:00:00 até as 20:59:59, Exemplo para o tempo 23:59:00 o valor que este campo assume é *No*.

No caso da presente dimensão foram definidos dois índices, um que é a chave primária (time_key) e o outro que é o atributo fulltime. Relativamente ao índice do atributo fulltime este foi criado devido à grande quantidade de consultas que são feitas sobre o mesmo, de forma a retornar a time_key. De ressalvar que este é um índice único uma vez que não podem existir dois instantes de tempo repetidos.

Na figura 25 podemos ver a hierarquia existente na tabela dim_time.

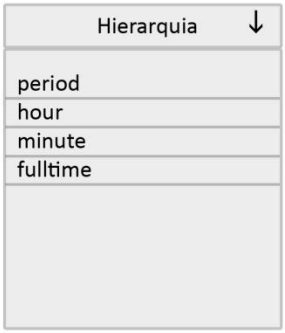

Figura 25: Representação da hierarquia existente na dimensão dim_time.

Ficando assim concluída a apresentação da dimensão dim_time bem como o seu processo de carregamento.

4.2.3 Dimensão dim_country

A dimensão dim_country foi criada com o intuito de esta conter todos países existentes. Para isso, foram efetuadas consultas ao ISO 3166-1 [Keopi, 2013] de forma a saber o nome e códigos corretos de cada país. Para além dos códigos que definem cada país foram ainda inseridas as siglas do continente a que este pertence assim como o nome do continente.

Tal como nos casos anteriores, o carregamento da dimensão da dim_country ocorre apenas uma vez, após a criação do *data warehouse* ou quando esta tabela se encontra vazia.

Relativamente aos atributos que fazem parte desta dimensão, estes poderão ser analisados na tabela seguinte.

Tabela 18 - Descrição dos atributos da dimensão dim_country.

Atributo	Descrição
country_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo.
country_code	Código de dois caracteres que definem o país de acordo com o ISO 3166-1 de dois caracteres. No caso é com este código que os países estão definidos no sistema operacional. Exemplo, para o país Portugal este campo assume o valor de PT.
country_name	Nome abreviado do país. Exemplo, para o país Espanha este campo assume o valor de Spain.
full_name	Nome completo do país. Exemplo, para o país Angola este campo assume o valor de Republic of Angola.
country_iso	Código de três caracteres que definem o país de acordo com o ISO 3166-1 de três caracteres Exemplo, para o país Angola este campo assume o valor de AGO.
country_iso_number	Código de três dígitos que definem o país de acordo com o ISO 3166-1 de três dígitos. Exemplo, para o país Angola este campo assume o valor de 024.
continent_code	Código de dois caracteres que definem o continente onde o país se encontra. Exemplo, para o país Angola este campo assume o valor de AF.
continent_name	Nome do continente onde o país. Exemplo, para o país Angola este campo assume o valor de Africa.

De salientar que no caso desta dimensão, não foi definido nenhum índice para além da chave primária (country_key).

Na figura 26 podemos visualizar a hierarquia existente na tabela dim_country.

Figura 26: Representação da hierarquia existente na dimensão dim_country.

Ficando assim concluída a apresentação da dimensão dim_country bem como o seu processo de carregamento.

4.2.4 Dimensão dim_email_kit

A dimensão dim_email_kit foi criada de forma a conter o nome dos kits de *e-mail* utilizados nos envios de *e-mails*. Tendo em conta que apenas existe um atributo que pode sofrer atualizações, iremos apenas encontrar atualizações do tipo 2 mantendo assim o histórico de todas as actualizações.

Relativamente aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 19 - Descrição dos atributos da dimensão dim_email_kit.

Atributo	Descrição
email_kit_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
email_kit_name	Nome do *e-mail* kit utilizado para o envio de *e-mails*.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

Para esta dimensão foram definidos dois índices em que, um é a chave primária (email_kit_key) e o outro é composto pelo conjunto de atributos extract_from e extract_from_id devido à elevada taxa de consultas a estes atributos durante o processo de importação de dados para a dimensão em causa.

Não existem hierarquias definidas nesta dimensão.

Ficando assim concluída a apresentação da dimensão dim_email_kit bem como o seu processo de carregamento.

4.2.5 Dimensão dim_email_list

A dimensão dim_email_list foi criada de forma a conter o nome das listas de *e-mail* utilizadas no envio de *e-mails*.

Relativamente aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 20 - Descrição dos atributos da dimensão dim_email_list.

Atributo	Descrição
email_list_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
email_list_name	Nome da lista utilizada para o envio de *e-mails*.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é relativamente simples, uma vez que apenas existem dois atributos possíveis de serem atualizados. Foi então definido que a atualização a ser efetuada é do tipo 2 para o único atributo. Portanto, significa que se este mudar é inserido um novo registo com as atualizações, sendo que o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante em que a *task* de carregamento iniciou a atividade.

No caso desta dimensão foram definidos dois índices, sendo o primeiro a chave primária (email_list_key). O segundo é composto pelo conjunto de atributos extract_from e extract_from_id, facto que se deve à elevada taxa de consultas a estes atributos durante o processo de importação de dados para a dimensão apresentada.

Não existem hierarquias definidas nesta dimensão.

Ficando assim concluída a apresentação da dimensão dim_email_list bem como o seu processo de carregamento.

4.2.6 Dimensão dim_mta

A dimensão dim_mta foi criada de forma a conter os dados sobre o MTA utilizado assim como o endereço de IP que efetuou o envio do *e-mail* e se este está certificado.

Relativamente aos atributos que fazem parte desta dimensão estes podem ser analisados na tabela seguinte.

Tabela 21 - Descrição dos atributos da dimensão dim_mta.

Atributo	Descrição
mta_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
mta_name	Nome do MTA que efetua o envio do *e-mail*.
mta_ip	Endereço de IP utilizado para efetuar o envio de *e-mail*.
mta_ip_return_path	Atributo que identifica se o endereço de IP utilizado um certificado da *return path* no envio de email. Este atributo pode assumir o valor de *Yes* caso utilize esteja certificado ou *No* caso contrário.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é uma tarefa simples, pois apenas existem três atributos possíveis de serem atualizados, a saber, o mta_name, mta_ip e o mta_ip_return_path. Foi então definido que a atualização a ser efetuada é de tipo 2 para todos os atributos, o que significa que se algum deles mudar é inserido um novo registo com as atualizações e o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante com que a *task* de carregamento iniciou a atividade.

No caso desta dimensão foram definidos dois índices em que um é a chave primária (email_mta_key) e o segundo é composto pelo conjunto de atributos extract_from e extract_from_id devido à elevada taxa de consultas a estes atributos durante o processo de importação de dados para a dimensão apresentada.

Na figura 27 podemos ver a hierarquia existente na tabela dim_mta.

Figura 27: Representação da hierarquia existente na dimensão dim_mta.

Ficando assim concluída a apresentação da dimensão dim_mta bem como o seu processo de carregamento.

4.2.7 Dimensão dim_integration_response

Relativamente à dimensão dim_integration_response esta foi criada tendo em vista guardar as respostas das integrações ocorridas. Esta dimensão pode ser considerada uma *junk dimension* pois guarda informação impossível de inserir em alguma das outras dimensões criadas, sendo que contém uma baixa cardinalidade e representa informação adicional sobre os eventos das tabelas de factos.

Relativamente aos atributos que fazem parte desta dimensão estes podem ser analisados na tabela seguinte.

Tabela 22 - Descrição dos atributos da dimensão dim_integration_response.

Atributo	Descrição
integration_response_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
integration_response	Nome da resposta da integração.

O processo de carregamento de dados para esta dimensão é um processo atípico tendo em conta que apenas são inseridos novos registos nesta dimensão durante o carregamento da tabela de factos e no caso de não existir um registo com o valor do atributo integration_response igual ao que se pretende carregar.

No caso desta dimensão não foi definido nenhum índice para além da chave primária (integration_response_key).

Não existem hierarquias definidas nesta dimensão.

Fica assim concluída a apresentação da dimensão dim_integration_response bem como o seu processo de carregamento.

4.2.8 Dimensão dim_campaign

A dimensão dim_campaign foi criada para agregar toda a informação relativa às campanhas, clientes e respetivas contas.

No que respeita aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 23 - Descrição dos atributos da dimensão dim_email_campaign.

Atributo	Descrição
campaign_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
campaign_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) para a campanha de onde os dados foram extraídos.
account_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) para a conta do cliente relacionada com a campanha de onde os dados foram extraídos.
client_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) para o cliente que esta relacionado com a conta e que por sua vez se encontra relacionada com a campanha de onde os dados foram extraídos.
campaign_extract_from	Este atributo contém o nome do sistema de onde os dados foram inicialmente extraídos. Isto acontece porque, os dados são extraídos do sistema operacional HurryUp mas existem factos que são extraídos de outros sistemas como tal convém saber a informação sobre o local original da extração dos dados no sistema operacional.
campaign_extract_from_id	Este atributo contém o identificador único (id) de onde os dados foram inicialmente extraídos. O princípio deste atributo encontra-se explicado no atributo origin_extract_from.
campaign_name	Nome da campanha.
campaign_payout	Valor definido por cada integração ou conversão efetuado para a campanha.
campaign_first_payout	Valor definido por cada integração ou conversão efetuado para a campanha na primeira vez que a

	campanha foi extraída.
campaign_is_active	Sinalizador se a campanha se encontra ativa. Caso esta se encontre ativa o atributo assume o valor de *Yes* ou *No* caso contrário.
campaign_rejection_rate	Taxa máxima de invalidações que podem ocorrer na campanha.
campaign_target	Número máximo de integrações ou conversões que podem ocorrer para a campanha.
campaign_target_revenue	Valor máximo a pagar pelo cliente pelas integrações ou conversões que podem ocorrer para a campanha.
campaign_vertical_id	Identificador único do sistema operacional do vertical em que a campanha se insere.
campaign_vertical	Vertical em que a campanha se insere.
campaign_currency_code	Moeda em que o *payout* da campanha está definido.
campaign_is_acquisition	Se a campanha é utilizada para a compra de contactos externos.
campaign_selling_to_business_unit_id	Identificador único do sistema operacional referente a unidade de negócio.
campaign_selling_to_business_unit	Unidade de negócio a que os contactos são adjudicados quando existe a sua compra.
account_name	Nome da conta a que a campanha pertence.
account_description	Descrição sobre a conta.
account_rejection_rate	Taxa máxima de invalidações que podem ocorrer na conta.
account_is_active	Identificador se a conta se encontra ativa ou não.
account_network_name	Nome da conta nas redes de afiliação.
account_network_username	Nome usado para efetuar o *login* na conta nas redes de afiliação.
account_type_name	Tipo de conta.
client_name	Nome do cliente a que pertence a campanha.
client_description	Breve descrição sobre o cliente.
client_is_active	Identificador se o cliente se encontra ativo ou não.
client_type_name	Tipo de cliente.
client_phc_id	Identificador do cliente no programa de facturação.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso do registo se encontrar ativo este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é um processo complexo devido ao elevado número de atributos que se encontram na dimensão em apreciação.

O carregamento de dados para esta dimensão é efetuado através da verificação da existência de algum registo ativo com os atributos extract_from, campaign_id, account_id e cliente_id. Devido ao sistema operacional permitir que uma campanha não possua conta ou cliente associados, é necessário verificar se existe algum registo que faça correspondência com os atributos extract_from, campaign_id e account_id ou extract_from, campaign_id. Em caso afirmativo, e portanto se existir a correspondência com os campos, é necessário verificar se o atributo account_id ou client_id estão definidos com o valor 0. Sendo que se assim for, é feita uma atualização do tipo 1 sobre todos os campos, isto porque pode ter sido carregado um registo de uma campanha sem associação de conta ou cliente(isto apenas pode acontecer neste caso). Caso exista correspondência e o valor do account_id e client_id seja diferente de 0 então é efetuada uma alteração do tipo 2 para todos os atributos. Isto significa que se algum deles mudar é inserido um novo registo com as atualizações e o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante com que a *task* de carregamento iniciou a atividade.

No caso da dimensão dim_campaign foram definidos quatro índices, em que um é a chave primária (campaign_key), o segundo é composto pelo conjunto de atributos extract_from, campaign_id, account_id e cliente_id devido à elevada taxa de consultas a estes atributos durante o processo de importação de dados para a dimensão apresentada. Já o terceiro índice é composto pelo conjunto de atributos origin_extract_from e origin_extract_from_id devido às elevadas consultas no carregamento das tabelas de factos. Por fim, o campaign_country_key que é a *foreign key* da dimensão dim_country.

Na figura 28 podemos ver a hierarquia existente na tabela dim_campaign.

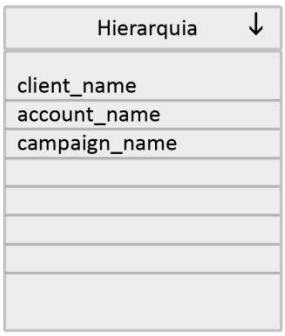

Figura 28: Representação da hierarquia existente na dimensão dim_campaign.

Ficando assim concluída a apresentação da dimensão dim_campaign bem como o seu processo de carregamento.

4.2.9 Dimensão dim_webpage

A dimensão dim_webpage foi criada para captar a informação sobre as páginas *web* que são utilizadas para a angariação de contactos e para onde o tráfego é redirecionado.

Relativamente aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 24 - Descrição dos atributos da dimensão dim_webpage.

Atributo	Descrição
webpage_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
webpage_origin_from	Nome do sistema operacional de onde a página *web* foi inicialmente recolhida.
webpage_origin_from_id	Identificador único de onde a página *web* foi inicialmente recolhida.
webpage_name	Nome da página *web* onde o contacto foi recolhido.
microsite_id	Identificador único do sistema operacional referente ao *microsite*.
microsite_name	Nome do *microsite*.
microsite_abbreviation	Abreviatura do nome do *microsite*.
microsite_description	Descrição acerca do *microsite*.
microsite_is_active	Identificador se o *microsite* se encontra ativo.
microsite_is_internal	Tipo de *microsite*, interno ou externo.
microsite_manager	Siglas do responsável pelo *microsite*.
microsite_business_unit_id	Identificador único do sistema operacional referente a unidade de negócio.
microsite_business_unit	Unidade de negócio que é responsável pelo *microsite*.
microsite_country_key	Este atributo é uma *foreign key* da dimensão dim_country e identifica o país em que o *microsite* pretende atuar.
microsite_vertical_id	Identificador único do sistema operacional referente ao vertical.
microsite_vertical	Vertical em que o *microsite* se insere.
website_name	Nome do *website* em que o *microsite* se encontra inserido.
website_description	Descrição sobre o *website*.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso do valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo

	de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é composto pela verificação da existência de um registo na tabela dim_webpage, através do conjunto de atributos extract_from, extract_from_id e microsite_id. Caso não exista nenhum registo é feita a inserção com os dados extraídos pelo sistema. No caso de existir um registo, é verificado se os seus atributos foram alterados e no caso de terem sido, é levada a cabo a operação de atualização do tipo 2 para todos os atributos. Isto significa que se algum deles mudar é inserido um novo registo com as atualizações e o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante em que a *task* de carregamento iniciou a atividade.

Relativamente aos índices que podemos encontrar nesta tabela, estes são a chave primária (webpage_key), o índice composto pelo conjunto de atributos extract_from, extract_from_id e microsite_id devido à elevada taxa de consultas a estes atributos durante o processo de carregamento dos dados para esta dimensão. E por fim, o índice que é o microsite_country_key que é a *foreign key* da dimensão dim_country.

Na figura 29 podemos ver a hierarquia existente na tabela dim_webpage.

Figura 29: Representação da hierarquia existente na dimensão dim_webpage.

De notar que foi criada uma *view* com o nome de view_microsite isto porque em alguns casos não existe a necessidade da granularidade ser ao nível da página *web* mas somente ao nível do *microsite*. Esta *view* pode ser consultada no subcapítulo 4.3.2.

Ficando assim concluída a apresentação da dimensão dim_webpage bem como o seu processo de carregamento.

4.2.10 Dimensão dim_traffic_source

Quanto à dimensão dim_traffic_source, esta foi criada de forma a armazenar a informação sobre as fontes de tráfego que são utilizadas para a angariação de contactos ou para o aumento do tráfego em *microsites*.

Relativamente aos atributos que fazem parte desta dimensão, estes podem ser analisados na tabela seguinte.

Tabela 25 - Descrição dos atributos da dimensão dim_traffic_source.

Atributo	Descrição
traffic_source_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
origin_extract_from	Este atributo contém o nome do sistema de onde os dados foram inicialmente extraídos. Isto acontece porque, os dados são extraídos do sistema operacional HurryUp mas existem factos que são extraídos de outros sistemas, como tal convém saber a informação sobre o local original da extração dos dados no sistema operacional.
origin_extract_from_id	Este atributo contém o identificador único (id) de onde os dados foram inicialmente extraídos. O princípio deste atributo encontra-se explicado no atributo origin_extract_from.
traffic_source_name	Nome da fonte de tráfego.
traffic_source_abbreviation	Abreviatura do nome da fonte de tráfego.
traffic_source_is_internal	Identificador que indica se a fonte de tráfego pertence a AdClick.
traffic_source_is_internal_flag	Valor boleano que identifica se a fonte de tráfego é interna. Onde 0 significa que não e 1 que é interna.
traffic_source_business_unit_id	Identificador único do sistema operacional referente a unidade de negócio.
traffic_source_business_unit	Unidade de negócio que é responsável pela fonte de tráfego.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o

	registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é composto pela verificação da existência de um registo na dimensão dim_traffic_source através dos atributos extract_from e extract_from_id. Caso não exista nenhum registo é feita a inserção do mesmo. Pelo contrário, e no caso de existir um registo é verificado se os atributos foram alterados e, em caso de terem sido é levada a cabo a operação de atualização do tipo 2 para todos os atributos. Isto significa que se algum deles mudar, é inserido um novo registo com as atualizações e o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante em que a *task* de carregamento iniciou a atividade.

Relativamente aos índices que podemos encontrar nesta tabela, estes são nomeadamente, a chave primária (traffic_source_key), o índice composto pelo conjunto de atributos extract_from e extract_from_id devido à elevada taxa de consultas a estes atributos durante o processo de carregamento dos dados para esta dimensão. E por fim, o índice que é composto pelo conjunto de atributos origin_extract_from e origin_extract_from_id devido ao número elevado de consultas no carregamento das tabelas de factos.

Na dimensão dim_traffic_source não se encontram definidas hierarquias.

Ficando assim concluída a apresentação da dimensão dim_traffic_source bem como o seu processo de carregamento.

4.2.11 Dimensão dim_tracking

No que respeita à dimensão dim_tracking esta foi criada com o objetivo de guardar informação sobre a forma como foram angariados os utilizadores para os *microsites*. Tal como a dimensão dim_integration_response, a dim_tracking pode ser considerada uma *junk dimension* porque guarda informação que não é possível inserir em nenhuma das outras dimensões criadas.

Relativamente aos atributos que fazem parte desta dimensão estes podem ser analisados na tabela seguinte.

Tabela 26 - Descrição dos atributos da dimensão dim_tracking.

Atributo	Descrição
tracking_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
utm_keyword	Nome da angariação de tráfego de publicidade utilizada. Exemplo nome da palavra-chave (*keyword*) utilizada ou nome do

	banner utilizado.
addicional_info_1	Informação adicional que exista sobre a *keyword* ou o local onde esta foi apresentada.
addicional_info_2	Informação adicional que exista sobre a *keyword* ou o local onde esta foi apresentada.

Por sua vez o processo de carregamento de dados para esta dimensão é um processo atípico tal como sucede com a dim_integration_response. É atípico uma vez que apenas são inseridos novos registos nesta dimensão durante o carregamento das tabelas de factos e caso não exista a correspondência dos dados com os atributos utm_keyword, addicional_info_1 e addicional_info_2.

Quanto aos índices definidos para esta dimensão, foram considerados nomeadamente, a chave primária (tracking_key) e o índice composto pelo conjunto de atributos utm_keyword, addicional_info_1 e addicional_info_2 devido à elevada taxa de consultas a estes atributos durante o processo de carregamento dos factos que tem esta dimensão como *foreign key*.

Não existem hierarquias definidas nesta dimensão.

Ficando assim concluída a apresentação da dimensão dim_tracking_type bem como o seu processo de carregamento.

4.2.12 Dimensão dim_user

No que concerne à dimensão dim_user, a mesma foi criada para guardar a informação sobre os contactos recolhidos. No entanto, deparamo-nos com um problema. Uma vez que a informação recolhida sobre o contacto é dinâmica. Logo, nunca é possível conhecer o número concreto de atributos que compõem a dimensão dim_user.

Como forma de resolução deste problema foram analisadas três hipóteses possíveis que nos propomos apresentar em seguida.

4.2.12.1 Hipótese 1 - Atributos extra na dim_user

Tendo em vista a resolução do problema da informação extra recolhida para os contactos, foi pensada uma solução que passava por se adicionar campos extra na dimensão dim_user onde seria possível inserir a informação extra recolhida. Isto seria possível tendo por base um par que é composto pelo conjunto do nome do campo extra e informação recolhida do contacto.

Para isso, foi feito uma análise da média de campos extra que são recolhidos por contacto e o máximo de campos extra que foram recolhidos para um contacto. Os valores que se encontraram foi 6 para a média de campos extra recolhidos por contacto e de 23 como o máximo de campos extra que foram recolhidos para um contacto.

O problema desta solução residiu no facto de que por um lado existiria sempre informação que iria ser perdida e por outro no mesmo atributo estariam assuntos díspares.

Pelos problemas apresentados esta solução foi descontinuada.

4.2.12.2 Hipótese 2 - Reconstrução da dim_user

A não adequação da primeira hipótese, e a necessidade de transportar toda a informação que consta nos sistemas operacionais acerca do contacto para o *data warehouse*, levou a considerar outras hipóteses

Assim, surgiu naturalmente a ideia de que se devia efetuar sempre uma alteração à estrutura da dimensão dim_user, o que é conhecido como uma atualização do tipo 3. Isto traduz-se no caso do atributo não se encontrar na dimensão dim_user. Então a estrutura da dimensão é alterada e são inseridos os novos atributo na mesma. Neste sentido, foram estudadas algumas técnicas de atualizações do tipo 3, em que se ressalvou a hipótese de renomear a tabela dim_user para dim_user_old e criar novamente a dimensão dim_user com os novos atributos. Com isto seria possível dar-se continuidade ao processo de abastecimento, assim como se abasteceria a nova tabela com os dados existentes na tabela dim_user_old.

No entanto, tal como na primeira hipótese deparámo-nos com um outro problema, que seria o tempo que a nova tabela iria demorar até conter toda a informação. Devido à elevada taxa de criação de novos campos extra para os contactos podia acontecer o paradoxo de o processo de carregamento dos dados da dim_user_old para a dim_user ainda não estar concluído e já ser necessário efetuar uma nova alteração à estrutura da dimensão dim_user.

Com isto esta opção foi igualmente abandonada.

4.2.12.3 Hipótese 3 - Dimensão dim_user_extra e *View* view_user

Após a insatisfação das hipóteses abordadas anteriormente, foi então encontrada a solução que pensamos ser a melhor para solucionar o problema principal.

Foi então definido que seria necessário uma tabela dim_user_extra, onde iria estar presente o conjunto do nome do campo, o id do campo extra e o valor introduzido pelo utilizador. Neste sentido aproximou-se esta tabela das encontradas nos modelos relacionais mas na sua forma mais desnormalizada possível.

Com isto ficaria resolvido o problema de guardar toda a informação extra dos contactos no *data warehouse*.

No entanto, saliente-se, que esta questão acrescenta um elevado nível de dificuldade na elaboração de *query's* para a consulta de dados. Assim sendo, e de forma a resolver este novo problema ficou então definido que o melhor seria criar uma *view* (view_user) que será

automaticamente (subcapítulo 4.3.1) gerada sempre que forem encontrados novos campos extra acerca dos contactos.

Sendo esta a hipótese que achamos mais adequada à resolução do problema existente, foi esta que optamos por implementar. Para tal, foi criada uma dimensão dim_user_extra que pode ser analisada no subcapítulo 4.2.13 e uma *view* que se encontra explicada no subcapítulo 4.3.1.

Sendo que, por agora continuamos a analisar a dimensão dim_user.

Relativamente aos atributos que fazem parte desta dimensão, estes encontram-se definidos na tabela seguinte.

Tabela 27 - Descrição dos atributos da dimensão dim_user.

Atributo	Descrição
user_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
extract_from	Este atributo contém o nome do sistema operacional de que os dados foram extraídos.
extract_from_id	Este atributo é a chave natural no sistema operacional, que é o identificador único (id) de onde os dados foram extraídos.
email	*E-mail* do contato.
email_domain	Domínio associado ao *e-mail* de contacto.
firstname	Nome próprio do contacto.
lastname	Apelido do contacto.
birthdate_key	Este atributo é uma *foreign key* da dimensão dim_date e identifica a data de nascimento do contacto.
gender	Identificador do sexo do contacto. Este pode assumir os valores *Male* se Homem, *Female* se Mulher e *Unknown* no caso de não existir esta informação.
phone	Número de telefone ou telemóvel do contacto.
postal_code	Código postal da morada do contacto.
city	Cidade da morada do contacto.
country_key	País da morada do contacto.
ip_address	Endereço de IP utilizado pelo contacto.
geo_postal_code	Localização geográfica do código postal em que o contacto se encontra através do endereço de IP no momento da recolha de informação do contacto.
geo_city	Localização geográfica da cidade em que o contacto se encontra através do endereço de IP no momento da recolha de informação do contacto.
geo_country_key	Localização geográfica do país em que o contacto se encontra através do endereço de IP no momento da recolha de informação sobre o mesmo.
activity_webpage_key	Este atributo é uma *foreign key* da dimensão dim_webpage e identifica a atividade do contacto numa página *web*.
activity_date_key	Este atributo é uma *foreign key* da dimensão dim_date e

	identifica a data de atividade do contacto.
tracking_system	Nome do sistema de *tracking* utilizado.
tracking_system_id	Identificador único que identifica o contacto no sistema de *tracking* utilizado.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo este atributo assume o valor de *NULL*.

Por sua vez, o processo de carregamento de dados para esta dimensão é diferente de todos os outros. Isto acontece devido ao facto de assumirmos que um contacto contém um *e-mail* que é único por pessoa. Assim, vamos ter de detetar se os contactos que estão a ser carregados já se encontram no *data warehouse* independentemente do sistema dos quais foram extraídos. Primeiro utilizamos os campos extract_from e extract_from_id para verificar a existência do contacto, isto porque pode ter existido a necessidade de criar a *foreign key* para este contacto no decorrer do carregamento da dimensão dim_user_extra que vamos ver no próximo subcapítulo. No caso de existir o registo são efetuadas alterações do tipo 1 sobre todos os atributos expecto os atributos extract_from e extract_from_id. No caso de não existir correspondência então é utilizada uma comparação direta através do *e-mail* do contacto recolhido com os registos que estão inseridos no *data warehouse*.

Tendo em conta que a informação recolhida é online, e que não existe nenhuma verificação física sobre a mesma, todos os atributos contemplam alterações do tipo 2 excepto o atributo email e email_domain.

Cumpre ainda salientar que o processo de carregamento desta dimensão é incremental tendo em conta que os dados sobre um utilizador nunca sofrem alterações nos sistemas operacionais.

No que respeita aos índices definidos para esta dimensão aqui vamos encontrar sete índices maioritariamente devido ao número de *foreign keys* existentes. Sendo que os índices são, a chave primária (user_key). O índice composto pelo atributo email, devido à elevada taxa de consultas a estes atributos durante o processo de carregamento da dimensão para encontrar os duplicados. O índice do atributo birthdate_key que é a *foreign key* da dimensão dim_date. O índice do atributo country_key que é a *foreign key* da dimensão dim_country. O índice do atributo geo_country_key que é a *foreign key* da dimensão dim_country. O índice do atributo activity_webpage_key que é a *foreign key* da dimensão dim_webpage. O índice do atributo activity_date_key que é a *foreign key* da dimensão dim_date.

Na figura 30 podemos ver a hierarquia existente na tabela dim_user.

Figura 30: Representação da hierarquia existente na dimensão dim_user.

Ficando assim concluída a apresentação da dimensão dim_user bem como o seu processo de carregamento.

4.2.13 Dimensão dim_user_extra

A dimensão dim_user_extra foi criada com o intuito de armazenar a informação adicional que existe sobre um dado contacto tal como explicado no subcapítulo 4.2.12.3.

Relativamente aos atributos que fazem parte desta dimensão estes podem ser analisados na tabela seguinte.

Tabela 28 - Descrição dos atributos da dimensão dim_user_extra.

Atributo	Descrição
user_extra_key	Chave primária da tabela. Este é um campo onde o valor é auto incrementado a cada inserção de um novo registo. É também conhecido como a chave artificial da dimensão.
user_key	Este atributo é uma *foreign key* da dimensão dim_user e identifica a informação sobre o contacto a que a informação pertence.
user_extra_field_id	Este atributo é a chave natural no sistema operacional do atributo extra.
user_extra_field_content	Este atributo contém o valor preenchido pelo contacto para o campo extra.
is_current	Este atributo é composto de uma de duas opções, *Yes* ou *No*, que significa se este registo se encontra ativo, no caso de *Yes* ou se o registo foi atualizado no caso de o valor ser *No*.
created_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados.
expired_date	Atributo de controlo que guarda a data e instante que o registo foi criado sendo este valor de acordo com o início do processo de importação dos dados. Em caso de o registo se encontrar ativo

este atributo assume o valor de *NULL*.

O processo de carregamento de dados para esta dimensão é composto primeiramente pela verificação da existência do user_key. No caso de este não existir é criado um registo com todos os campos a *Unknown* excepto o extract_from e extract_from_id. Após a criação é utilizado o user_key e o user_extra_field_id para verificar a existência de algum registo. Caso não exista nenhum registo, é feita a sua inserção. No caso de existir um registo, é verificado se os seus atributos foram alterados e no caso de terem sido é levada a cabo a operação de atualização do tipo 2 para todos os atributos. Isto significa que se algum deles mudar, é inserido um novo registo com as atualizações, sendo que o registo antigo passa a conter o atributo is_current como *NO* e o atributo expired_at com a data e instante em que a *task* de carregamento iniciou a atividade.

Relativamente aos índices que podemos encontrar nesta dimensão estes são, nomeadamente, a chave primária (user_extra_key e user_key). E o índice que é no atributo user_key que é a *foreign key* da dimensão dim_user.

Não existem hierarquias definidas nesta dimensão.

Ficando assim concluída a apresentação da dimensão dim_user_extra bem como o seu processo de carregamento.

4.3 Criação das *view*

Neste subcapítulo serão explicadas as *views* criadas, assim como o processo de criação da *view* caso esta seja dinâmica.

Relembrando que uma *view* é uma consulta pré-guardada na base de dados sobre uma ou várias tabelas e que quando consultada produz um conjunto de resultados. Neste sentido, uma *view* atua como tabela virtual [Oracle, & affiliates, 2013b]. Cumpre referir que os dados não são guardados, mas apenas a *query* usada para a consulta é armazenada. Assim, uma *view* não contém os registos físicos dos dados, porem apenas apresenta os dados de forma a facilitar a sua visualização. Por fim, saliente-se que as *view* são também usadas para limitar o acesso aos dados por parte dos utilizadores.

4.3.1 *View* view_user

Relativamente à *view* view_user compete referir que a mesma foi criada de forma a controlar o problema dos campos extra dos contactos recolhidos. Assim, esta *view* é criada na construção do *data warehouse* com os mesmos campos que se encontram na dimensão dim_user.

Após a construção do *data warehouse* e com o início da recolha de informação para a tabela user_extra_field na *staging area* é necessário reconstruir a view_user, de modo a que os novos campos dinâmicos recolhidos como informação adicional do contacto estejam disponíveis para consulta.

Assim sendo, configuramos a *task* LoadView com a opção view_user para efetuar a criação da vista.

O processo de criação da vista é feito através da consulta aos dados que se encontram na tabela user_extra_field na *staging area*. Estes dados passam a ser considerados como atributos da nossa *view* construindo-se assim a *query* para a criação da *view*. Após a criação da *query* para gerar a nova *view* é adicionada uma verificação para apagar a *view* já existente e só depois é criada a nova view.

Ficando assim concluída a apresentação e criação da view_user.

4.3.2 *View* view_microsite

No que respeita à view_microsite esta foi criada de forma a possibilitar a consulta dos dados da dimensão dim_webpage mas de forma mais agregada. Assim esta *view* é na sua essência uma consulta à dim_webpage sendo que o nível mais elementar é o *microsite* e não a página *web*.

4.4 Carregamento dos factos

Neste subcapítulo iremos analisar o carregamento dos dados que constam na *staging area para as* tabelas de factos.

De notar que como as tabelas de factos já foram apresentadas no capítulo 3 agora iremos apenas efetuar a análise de como os dados são carregados para as mesmas e que mecanismos foram preparados para que estas sejam tolerantes a falhas. Contudo não iremos abordar detalhadamente cada processo de carregamento individualmente devido a estes serem muito similares. Por isso mesmo foi decidido que iremos separar o carregamento dos factos em dois grupos os factos não agregados e os factos agregados e efetuaremos a análise de um carregamento para cada um dos grupos.

Cumpre salientar que os carregamentos dos dados para o *data warehouse* é feito às 00 horas, 8 horas e 16 horas pois pretende-se ter o *data warehouse* sempre atualizado ao longo do dia. De notar que antes de se efetuar o carregamento de qualquer tabela de factos é sempre efetuado o carregamento das dimensões. Esta abordagem foi efetuada de forma a mitigar os possíveis problemas de dependências de relações entre as tabelas de factos e os registos nas dimensões.

Note-se ainda que todas as tabelas de factos encontram-se particionadas por intervalo (pelo atributo date_key de forma a ser um ano por partição) de forma a otimizar o acesso aos dados.

Relativamente ao carregamento das tabelas de factos inicialmente é sempre efetuada a consulta há tabela controller_task da *staging area* para se saber a data e hora do último carregamento para a tabela de factos. No final do processo de carregamento o registo da controller_task é atualizado com a data-hora que foi utilizada como limitador superior na extração dos dados da tabela da *staging area*.

4.4.1 Grupo tabela de factos não agregada

Tal como anteriormente referido os processos de carregamento das tabelas de factos foram separados em dois grupos de processos similares. Sendo que para o grupo de factos não agregados os processos similares são o de carregamento das tabelas de factos fact_email_sent , fact_email_sent_conversion, fact_traffic_tracking, fact_traffic_conversion e fact_integration. Posto isto iremos ver como exemplo o carregamento da tabela de factos fact_email_sent.

Relativamente ao carregamento dos dados para a fact_email_sent o processo é iniciado pela extração dos dados que constam na tabela email_sent_dqp da *staging area* e que será aqui apresentada.

Após a extração é despoletado o processamento dos registos extraídos. Segue-se então a extração dos dados que constam na tabela email_sent da *staging area* através do atributo updated_at como limitador superior utilizando a data e hora da última extração.

Comecemos então pela tabela email_sent_dqp que foi criada para armazenar os dados quando estes não conseguem ser inseridos na tabela de factos. Esta tabela é idêntica à tabela email_sent mas com os campos adicionais fail_info e created_at que guardam o motivo pelo qual o registo não foi carregado para a tabela de factos e a data e hora de inserção do registo nesta tabela.

Quando iniciamos o método de carregamento dos factos é primeiro extraída a informação, caso exista, desta tabela e é tentado inserir o registo na tabela de factos segundo o mesmo princípio aplicado aos dados que constam na tabela email_sent e que veremos de seguida. Ressalvando-se que caso se conseguir efetuar o carregamento o registo que consta na tabela email_sent_dqp é apagado da tabela, mas caso contrário este é atualizado se necessário.

De seguida é iniciado o processo de carregamento dos dados extraídos da tabela email_sent. Este processo consiste primeiramente na verificação da existência de todas as *foreign keys*.

No caso de todas as *foreign keys* estarem presentes é criada a instrução de inserção do registo na tabela de factos com as *foreign keys* assim como as suas métricas. De notar que no caso do

registo já existir as métricas são substituídas. Isto acontece devido a latência de algumas das suas métricas como por exemplo a unsubscribed.

No caso de não se encontrar uma ou mais *foreign keys* o registo é inserido na tabela email_sent_dqp com a informação extraída da tabela email_sent, o motivo da falha no campo fail_info e a data e hora do início do carregamento no campo created_at.

Após estar concluído o carregamento dos dados é efetuada a verificação da existência de registos com mais de 1 dia na tabela email_sent_dqp através do campo created_at. No caso de existirem tais registos é enviado um *e-mail* de alerta para o administrador do sistema inspecionar o motivo destes ainda não terem sido carregados com sucesso.

4.4.2 Grupo tabela de factos agregadas

Relativamente ao segundo grupo de tabelas de factos com similaridades no seu carregamento este é composto pelo carregamento das tabelas de factos fact_email_sent_daily, fact_email_sent_conversion_daily, fact_traffic_overall e fact_integration_overall. Sendo que no seguimento do que foi especificado no subcapítulo 4.4 iremos verificar o carregamento de uma das tabelas, no caso da tabela de factos fact_email_sent_daily.

No que concerne ao carregamento dos factos para a tabela fact_email_sent_daily a informação que será guardada nesta tabela é a que se encontra na tabela de factos fact_email_sent mas agregada.

Como tal temos de ter em consideração que a tabela fact_email_sent sofre atualizações devido a algumas das suas métricas terem um tempo de carregamento tardio. Tendo em conta este facto a tabela fact_email_sent_daily irá sempre atualizar os registos relativamente aos últimos 30 dias. Assim sendo e por uma questão de performance no carregamento dos factos para esta tabela os dados dos últimos 30 dias são apagados e são inseridos os novos dados agregados.

O que se traduz que o carregamento desta tabela de factos é feito quase exclusivamente através de instruções na base de dados. Onde a primeira instrução é composta com a data a partir dos quais os dados devem de ser apagados e a segunda instrução sendo a consulta a tabela fact_email_sent mas com o nível de agregação pretendido.

5 Conclusões e trabalho futuro

O objetivo deste trabalho consistiu na implementação de um *data warehouse* capaz de armazenar a informação que se encontrava dispersa pelos vários sistemas operacionais da empresa AdClick e com isto ser possível a criação de um sistema para ajudar na tomada de decisões por parte da empresa.

Ao longo do trabalho da dissertação foram sempre apresentadas soluções que se encaixaram facilmente com as necessidades do desenvolvimento do *data warehouse*, no entanto a realidade de implementação do mesmo foi bem diferente.

Comecemos primeiro por falar na dificuldade de extração dos dados. A extração dos dados dos sistemas operacionais foi um dos maiores problemas encontrados no decorrer da implementação do *data warehouse.* Isto aconteceu porque a grande maioria dos sistemas operacionais não continham nenhuma forma de extração de dados e como tal foi necessário desenvolver métodos que devolvessem esta informação.

A segunda dificuldade encontrada foi a volatilidade dos dados nos sistemas operacionais. Note-se que quando se refere a volatilidade dos dados não se refere a rápida mudança dos mesmos mas sim a estes estarem a ser constantemente apagados ao longo de um dia. Isto causa um problema de integridade de dados no *data warehouse* devido ao mesmo ser construído tendo por base a não volatilidade dos factos.

A terceira dificuldade encontrada foi relativa aos campos extra dos contactos. Esta dificuldade prendeu-se com não ser possível construir uma dimensão com todos os atributos necessários tendo em conta que os mesmos são dinâmicos.

A quarta dificuldade encontrada no decorrer da implementação prendeu-se com o facto das *tasks* (rotinas) de carregamento das dimensões e dos factos estarem a consumir muita memória da máquina e ter momentos em que paravam a execução por falta de memória.

Contudo e apesar destas dificuldades foi sempre possível encontrar pelo menos uma solução para cada problema.

No primeiro caso a solução passou por criar API's para fornecerem os dados que eram necessários.

No segundo caso a solução passou por extrair sempre a informação relativa aos últimos seis dias da data corrente entre as 2 horas e às 6 horas da manhã de forma a consolidar os últimos 6 dias.

A terceira solução passou por criar um método de criação da *view* de forma dinâmica a partir de uma dimensão com campos estáticos e de outra com campos adicionais.

A solução para o quarto problema passou por limitar o volume de dados a carregar e de lançar novamente o processo sempre que o número de dados extraídos para carregar seja igual ao limite de dados a extrair para carregamento.

Relativamente as suguestões de trabalho futuro a desenvolver seria interessante a criação de uma tabela de factos fact_email_sent_overall em substituição das tabelas de factos fact_email_sent_daily e fact_email_sent_conversion_daily. Nesta tabela de factos fact_email_sent_overall estaria representado todo o processo de *e-mail marketing.*

Como trabalho futuro seria também intersante que fosse feita a implementação de uma ferramenta de relatórios (OLAP) e de uma ferramenta de *data mining* para trabalhar sobre o *data warehouse* pois só assim serão devidamente exploradas todas as potencialidades dos dados contidos no *data warehouse*.

Relativamente ao impacto que o projeto teve ressalva-se a vantagem estratégica que a empresa pode agora retirar tendo em conta que foi implementado o *data warehouse* e os processos de ETL para carregamento de todas as fontes de informação.

Estando esta agora munida de um sistema capaz de dar uma visão histórica a nível de cada processo de negócio ou se necessário da empresa no seu todo. A capacidade de se efetuarem análises mais rápidas e flexíveis tendo em conta a maior autonomia dos gestores. E destaca-se ainda a centralização dos dados num só sistema assim como a possibilidade de extrair informação a partir do que eram apenas dados soltos pelos diversos sistemas operacionais.

Referências

[37signals, 2013] Behind the scenes: A/B testing part 3: Finalé by Jamie of 37signals. http://37signals.com/svn/posts/2991-behind-the-scenes-ab-testing-part-3-final [último acesso: Maio 2013]

[About.com, 2013] Marketing – Information and Advice on Marketing Strategy, Marketing Plans, and Marketing Careers and Jobs http://marketing.about.com/od/marketingglossary/g/affiliatedef.htm [último acesso: Maio 2013]

[AdClick., 2013a] AdClick, Lda. http://www.adclickint.com/business-we-generate-business [último acesso: Maio 2013]

[AdClick., 2013] AdClick. http://www.adclickint.com/business-our-pillars [último acesso: Julho 2013]

[Afilea, 2013] Afilea. http://afilea.com/ [último acesso: Julho 2013]

[Caldeira, C. P. , 2008] Caldeira, C. P., Data Warehousing Conceitos e Modelos, Edição Silábos Lda. 2008

[Han e Kamber, 2001] Han, J., Kamber M., Data Mining: Concepts and Techniques. Morgan Kaufman Publishers, 2001.

[HasOffers, 2013] HasOffers. http://www.hasoffers.com/about/ [último acesso: Maio 2013]

[HurryUp, 2013] HurryUp. http://hurryup.adctools.com [último acesso: Julho 2013]

[Luhn,H. P., 1958] Luhn, H. P., A Business Intelligence System, IBM Journal of Research and Development , vol.2, no.4, 1958.

[Inmon, W. H., 2005] Inmon, W. H., Building the Data Warehouse, Fourth Edition. Wiley Publishing, Inc, 2005.

[JSON, 2013] JSON.
http://json.org/json-pt.html
[último acesso: Julho 2013]

[Keopi, 2013] Keopi, ISO 3166-1 Alpha-3 - CountryCodes.CO
http://countrycodes.co/country-codes/iso-3166-1-alpha-3/
[último acesso: Julho 2013]

[Kimball e Ross, 2002] Kimball R., Ross M. , The Data Warehouse Toolkit, Second Edition., John Wiley and Sons, Inc., 2002

[Kimball e Ross, 2008] Kimball R., Ross M. , The Data Warehouse Lifecycle Toolkit, Second Edition., Wiley Publishing, Inc, 2008

[Oliveira, P., 2013] Oliveira, P., Moodle Isep - Instituto Superior de Engenharia do Port.
https://moodle.isep.ipp.pt/file.php/234515/Teoricas/MDM/MultiDimensional_Modeling.pdf
[último acesso: Maio 2013]

[Oracle, & affiliates, 2013a] Oracle, & affiliates, O. C. (2013). MySQL :: MySQL 5.7 Reference Manual :: 11.6 Data Type Storage Requirements.
http://dev.mysql.com/doc/refman/5.7/en/storage-requirements.html
[último acesso: Maio 2013]

[Oracle, & affiliates, 2013b] *MySQL :: MySQL 5.7 Reference Manual :: 18.5 Using Views*.
http://dev.mysql.com/doc/refman/5.7/en/views.html
[último acesso: Junho 2013]

[Power, 2007] Power, D. J., "A Brief History of Decision Support Systems".
http://dssresources.com/history/dsshistory.html
[último acesso: Julho 2013]

[Sezões et al., 2006] Sezões C., Oliveira, J., Batista M., Business Intelligence. Sociedade Portuguesa de Inovação, 2006.

[Symfony, 2013] Symfony at a Glance - Symfony.
http://symfony.com/at-a-glance
[último acesso: Setembro 2013]