Projeto Final em Engenharia Informática

Projeto de Data Warehouse

IMPLEMENTAÇÃO E ANÁLISE

Pedro Miguel Sequeira Narciso

Professor Luís Cavique

Lisboa 10 de Dezembro de 2020

Conteúdo

Lista de figuras

Lista de tabelas

Lista de abreviaturas

DM – Data Marts

DW – Data Warehouse

DWB – Data Warehouse Bus

ETL - Extract Transform and Load

OLAP – Online analytic processing

OLTP – Online transaction processing

RDBMS - Relational Database Management System

SQL – Structed Query Language

Introdução

Este trabalho foca-se numa implementação eficaz de uma data warehouse (DW) a qual será elaborada com foco na base de dados de exemplo *sakila*. (Oracle, 2020) Para implementação da DW irá proceder-se ao estudo de uma empresa fictícia, designada por VideoX cuja área de atividade é o aluguer de vídeos em lojas físicas. A VideoX possui nos sistemas informáticos três bases de dados operacionais: A base de dados *sakila*, dedicada ao foco negocial da empresa, uma base de dados de dedicada aos recursos humanos, e uma base de dados de fornecedores na qual se regista as compras e os contactos efetuados.

Ao longo do trabalho irá proceder-se à analise das fontes de dados disponíveis, e recorrendo a casos de estudo desenvolver uma DW com foco na análise das alternativas de implementação existentes.

ORGANIZAÇÃO DO RELATÓRIO

O trabalho está organizado por temas que visam descrever o processo de desenvolvimento. Um primeiro tema será dedicado ao levantamento bibliográfico, segue-se uma análise da empresa VideoX, com o levantamento de requisitos e em último os casos de estudo para implementar com eficácia a DW.

Neste relatório incidirá principalmente sobre o levantamento bibliográfico, e análise da empresa VideoX, ficando a parte de desenvolvimento e implementação para o relatório final.

CRONOGRAMA

Figura 1 : Cronograma

Capítulo 1 – Levantamento bibliográfico

Revisão da bibliografia relevante nos temas data warehouse e de extração tratamento e carregamento de dados, ETL na sigla em Inglês para *Extract Transform and Load*.

1.1. DATA WAREHOUSE

Definições e propósito de uma *data warehouse*

> *"A collection of integrated, subject-oriented databases designed to supply the information required for decision-making."* (Inmon, 2002)

O termo *data warehouse* foi primeiramente apresentado por Bill Inmon no livro *Building the Data Warehouse. 1st Edition. Wiley and Sons* publicado em 1992. Este livro e as edições subsequentes foram em conjunto com a série lançada por Ralph Kimball *The Data Warehouse Toolkit, Willey and Sons* publicado em 1996 os percursores de duas metodologias de implementação de uma data warehouse.

Data warehouse não é considerada uma nova tecnologia, mas sim uma nova forma de aproveitar as tecnologias existes para de forma concertada resolver um problema que a maior parte das empresas e instituições vinham a sentir relativamente aos dados que vinham acumulando nos seus sistemas. A necessidade de recolha de informação em sistemas de bases de dados operacionais levantava muitas dificuldades, e eram baseadas em relatórios extraídos segundo visões diferentes daqueles a quem eram solicitados. Não raras vezes relatórios solicitados a diferentes departamentos resultavam em informações dispares entre eles.

O conceito de *data warehouse* advém da necessidade de existir uma ferramenta dedicada ao suporte da decisão. As bases de dados tradicionais apresentam como principal caraterística o facto de serem estruturadas para transações e customizados para sistemas integrados, com o principal foco nos processos e requerimentos de determinado departamento dentro de uma empresa. Numa *data warehouse* pretende-

se ir além das visualizações tradicionais, concentrando-se nos assuntos que abrangem toda a empresa, fornecendo um quadro único e completo dos processos de negocio.
Bill Inmon e Ralph Kimball, pioneiros e visionários em data warehouse, criaram conceitos e princípios de modelagem que ainda hoje são vistos com referências. Mesmo a trabalharem de forma independente e com modelos distintos os conceitos e princípios apresentados possuem muitos pontos em comum. Fon Silvers [SILVERS 2008] descreve-esses conceitos e princípios da seguinte forma:

Orientado ao assunto

Os dados numa DW não refletem as transações operacionais, mas devem apresentar-se como um reflexo das áreas temáticas da empresa. Estes devem ser organizados de maneira a que reflitam os assuntos e de forma a que seja possível manter a sua arquitetura ao longo do tempo.

Integração de Dados

Numa data warehouse (DW) os dados devem ser uniformes de forma a permitir a sua exploração por diversas áreas de forma coerente e com o mesmo significado.

Não volatilidade

Os dados uma vez carregados em DW devem ser persistentes e não estarem sujeitos a atualizações, e não devem ser apagados destes sistemas.

Dados temporais

Os dados devem ser expressos com uma referência temporal de forma a expressar os eventos empresariais numa perspetiva e num contexto histórico.

Uma única versão dos dados

Os dados devem ser moldados e guardados de forma a apresentar uma única versão da realidade, e a sua origem deve de ser a versão única e não ter origem em cópias que por vezes são efetuadas para reaproveitamento. Restringe-se o seu valor a apenas uma realidade de forma a que só um tipo de resposta possa ser retribuído sobre essa realidade.

Investimento a longo termo

Porque se pretende que uma DW seja um registo histórico é importante que os dados tenham longevidade e estabilidade. Uma DW deve ser flexível o suficiente para puder acomodar o crescimento da empresa e crescer com a variante histórica.

1.2. WILLIAM (BILL) H. INMON

Bill Inmon (Inmon, 2002) apresenta um modelo de implementação de uma data warehouse segundo uma metodologia que segue os princípios de uma visão *top down*. Este método surge como uma evolução natural do modelo relacional ficando conhecido com *Third normal form data model* o qual permite refletir os dados representados nos DBMS's da empresa.

Metodologia

O desenvolvimento da data warehouse deve ser efetuado com base em 3 níveis de modelação dos dados. Num primeiro nível a que Bill Inmom designa pelo mais elevado as diferentes áreas de negócio da empresa ou departamentos devem definir por si os dados relevantes e apresentar um modelo de diagrama entidade-relação, (ERD - Entity Relation Diagram) o qual deve representar as entidades, os atributos e relacionamentos dentro dos dados disponíveis nas bases de dados que lhes são afetas. Estes modelos devem manter-se o mais fiáveis relativamente ao modelo relacional das bases de dados de origem., com as adaptações necessárias mantendo as caraterísticas relacionais. Do resultado da concatenação dos diferentes modelos de dados resultantes é criado um modelo empresarial.

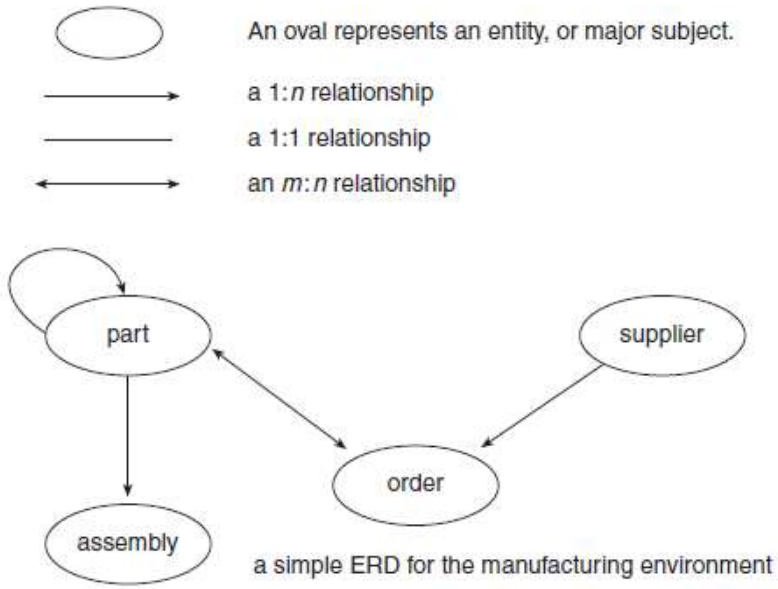

Figura 2: ERD departamental. (Inmon, 2002)

Num segundo nível de modelação designado por nível central, em que se estabelece o conjunto de dados (DIS – Data Item Set) para cada entidade do modelo ERD.

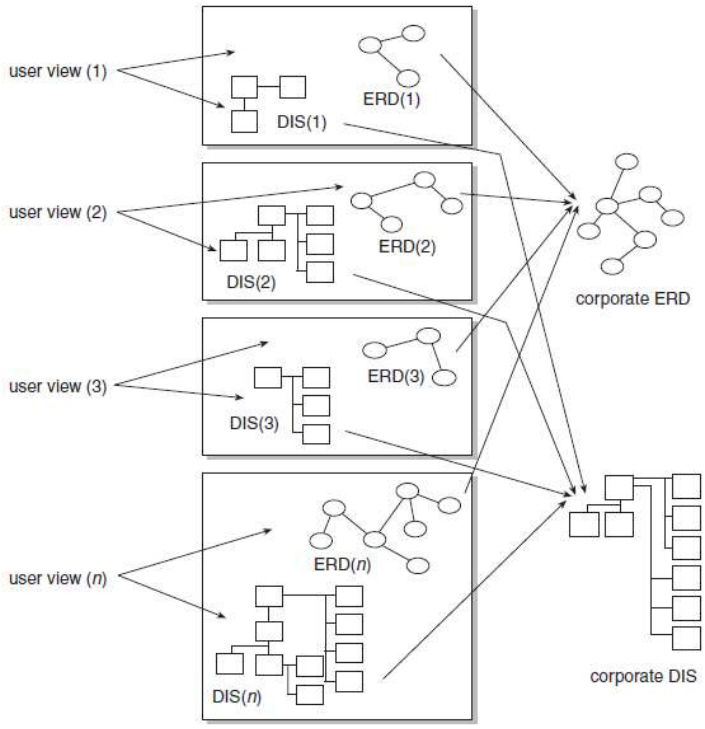

Figura 3: O DIS da empresa. (Inmon, 2002)

Os vários DIS são integrados através de 4 construtores sequenciais, em que o primeiro efetua um agrupamento dos dados nos atributos que não se repetem para aquela entidade, uma segunda fase de construção na qual irão constar atributos que podem existir várias vezes para cada entidade, e uma terceira fase em que são criados os conetores que irão relacionar os dados. Por último, uma quarta fase de construção em que é identificado os atributos dos dados.

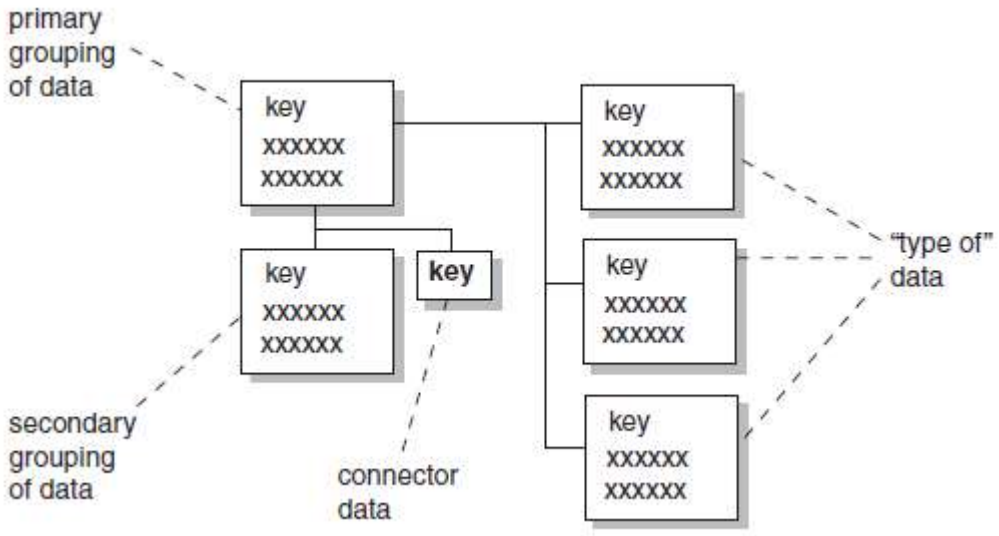

Figura 4: Construtores de segundo nível de modelação (Inmon, 2002)

Por último, um terceiro nível de modelação, a um nível mais baixo, também designado por modelo físico, que reflete o resultado do segundo nível ao qual é adicionado as chaves principais.

Este modelo requer um conhecimento elevado do negócio da empresa, assim como uma grande capacidade de entendimento e capacidade de modelagem.

1.3. RALPH KIMBALL

Ralph Kimball (Kimball & Ross, 2002) apresenta um modelo de implementação diferente da implementação tradicional de uma base de dados relacional. Usam uma metodologia designada por Modelação Dimensional, e a utilização de múltiplas Bases de Dados na

sua arquitetura designadas de *Data Marts, (DM)* que são integradas entre si através de um *Data Warehouse Bus (DWB)* de forma a garantir a coerência entre os diferentes *data marts*. Resulta assim uma metodologia que contrasta com o modelo de Imnon, principalmente na forma de construção da DW que resulta numa interpretação *Bottom-up* dos sistemas de dados da empresa.

Metodologia

Kimball (Kimball & Ross, 2002) identifica 4 passos no processo de desenho de uma data warehouse.

Selecionar a área de negocio, e o processo que se pretende modelar.

Declarar o nível de detalhe.

Escolher as dimensões que relevantes.

Identificar os factos numéricos.

O resultado será uma base de dados composta por uma tabela de factos, ligada por tabelas de dimensões. Esta carteiristas levou a que o modelo dimensional se designe também por *star schema*, uma vez que quando representadas em gráfico a sua aparência assemelha-se ao desenho de uma estrela.

Figura 5:Tabelas de facto e dimensões num modelo dimensional (Kimball & Ross, 2002)

As principais caraterísticas de uma *data warehouse* com base nesta implementação é que ela apresenta uma construção vocacionada para a consulta. As consultas tornam-se mais fáceis de elaborar, pois a implementação evita junções sucessivas entre tabelas

relacionadas, e proporciona um local central onde as medidas constam em tabelas de factos que se encontram relacionadas com tabelas de dimensões que lhes dão o contexto. Tudo isto torna o trabalho de pesquisa menos elaborado, pois exige menos conhecimentos técnicos aos utilizadores na área de pesquisa os quais normalmente têm outras aptidões em áreas de interesse diferentes de interesses informáticos.

Esta implementação, sendo muito simplista adquire uma dimensão mais complexa no momento em que vários factos são incluídos de forma a permitir diferentes data-marts que irão coexistir na mesma data warehouse. Esta evolução resulta em schemas cuja configuração vai adquirir um formato de constelação, na qual as diversas tabelas de factos relacionam-se com as tabelas dimensão que lhes dão contexto.

As tabelas de dimensões são por regra tabelas desnormalizadas, com muitos campos. São tabelas que considerando a sua relação com as tabelas de factos possuem registos que sofrem menos inserções do que os factos. No entanto existem registos que por uma questão de organização podem ser alvo de normalização, evitando assim surjam repetidas em elevado número. Normalmente são campos cujos dados ocupam um espaço maior. Nesta situação pode ser útil normalizar a tabela de dimensão, resultando a DW numa configuração conhecida por *snow flake*.

1.4. IMPLEMENTAÇÃO TOP-DOWN VERSUS BOTTOM-UP

Paulraj Ponniah (Ponniah, 2001) descreve de forma sucinta as vantagens e desvantagens relativamente a cada uma destas abordagens: Numa adaptação livre pode-se descrever da seguinte forma:

Vantagens de uma abordagem tipo *Bottom-up*

Implementação mais fácil e rápida.

Provas de conceito e retorno de investimento mais favoráveis.

Menor risco de falha na implementação.

Incremental e escalável, o que permite a implementação de *Data Marts* após primeiras implementações.

Permite que a equipa do projeto aprenda e cresça com o mesmo.

Desvantagens:

Cada *data mart* possui a sua visão própria dos dados.

Permite data redundante nos *data mart's*

Dados inconsistentes e irreconciliáveis são perpetuados

Prolifera interfaces não gerenciáveis.

Vantagens de uma abordagem tipo *Top-Down*

Transmite uma visão corporativa dos dados.

Arquitetura unida, e não junção de diversos *data mart's*.

Armazenamento central de dados relacionados ao conteúdo.

Regras de controle centralizadas.

Se implementado com iterações, pode-se ter resultados rápidos.

Desvantagens

Demora mais tempo para ser implementado.

Maior exposição ao risco de falha na implementação.

Necessita de uma equipa de técnicos com maiores conhecimento e experiência.

Despesas mais elevadas e dificuldade de implementar ambiente de prova de conceito.

1.5. ETL - EXTRACT TRANSFORM LOAD

O processo de ETL é um dos principais pilares de uma data Warehouse, e corresponde aos processos de extrair dados das bases de dados e outras fontes relevantes o conteúdo significante, transformar através de ações de limpeza e verificação de conformidade os

dados segundo os requerimentos estabelecidos e inserir esses mesmos dados numa data warehouse.

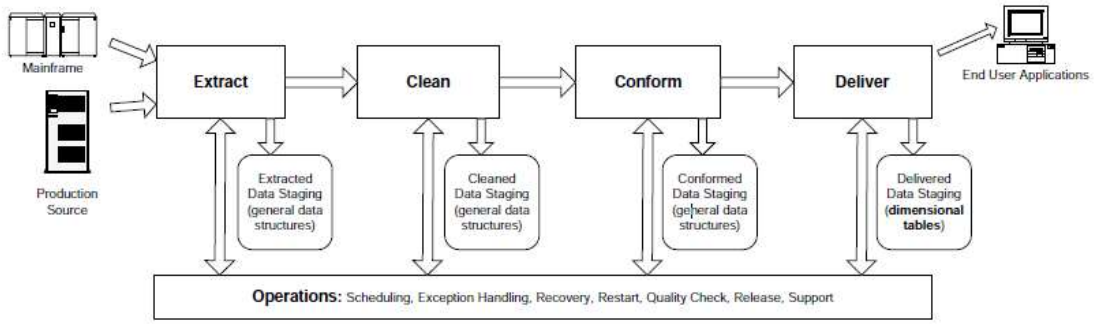

Figura 6:O processo de ETL – Fonte: (Kimball, Ralph; Caserta, Joe, 2004)

A estruturação de um processo de ETL deve começar por estabelecer os requisitos do processo. Ralph Kimball identifica os seguintes no livro Data Warehouse ETL Toolkit (Kimball, Ralph; Caserta, Joe, 2004): as necessidades do negócio; a conformidade dos dados; o perfil dos dados; a segurança dos dados; a integração dos dados; a latência dos dados; os interfaces do utilizador final; os conhecimentos técnicos; o licenciamento de sistemas; e por fim a escolha da arquitetura

Estes requisitos devem de ser definidos com clareza, de forma a que o processo de extração, transformação e carregamento de dados seja efetuado de forma transparente e os processos de ETL ficarem bem definidos.

O processo de extração deve ser incidir sobre os dados empresarias e guardados em tabelas independentes de forma serem trabalhados sem causar impactos nos sistemas operacionais.

O processo de transformação dos dados vai assim lidar com dados com origem em diferentes sistemas, cada qual com as suas próprias regras e tipos de dados que podem não obedecer ao padrão definido para a DW. Este processo incidirá assim na limpeza e transformação dos dados que não estão em conformidade com os requisitos, levando-os a um ponto de consistência. Como exemplo podemos encontrar moradas e números de telefone com diferentes formatos, códigos de produtos com diferentes anotações ou até mesmo campos numéricos com diferentes implementações. Também registo

incompletos e sem interesse podem ser alvo de transformação ou até mesmo descartados se não tiverem significado relevante. Outro dos fatores importante nesta fase é a colocação dos dados na granularidade definida. Nem sempre a caraterística da data warehouse requer que os dados sejam transpostos na granularidade com que se apresentam nas bases de dados operacionais, ou potras vezes o mesmo dado tem origem de sistemas com diferentes granularidades. Um exemplo prático é por exemplo medidas que podem estar registadas em diferentes sistemas métricos.

Segue-se o processo de carregamento dos dados em data warehouse, quando estes cumprirem os requisitos e estarem em conformidade com as necessidades.de pesquisa do utilizador final. Este passo é importante num sistema de data warehouse, porque pretende-se que sejam carregados uma única vez com o seu significado temporal, e não serem alvo de futuros atualizações e, ou correções.

1.6. STRUCTED QUERY LANGUAGE - SQL

SQL é a linguagem padrão das bases de dados, e como tal não deixa de ter um papel relevante em todo o processo relativo a data warehousing.

Para o processo de ETL recomenda-se a utilização de ferramenta desenhadas para esse propósito, uma vez que têm integradas funcionalidades que irão facilitar em muito todo o processo, tornando-o mais ágil, rápido, resultando num processo mais fácil. Este facto torna-se ainda mais relevante em sistemas de maior complexidade, e com volume de dados de diversas origens e por vezes dispares entre si na sua forma.

Não existe qualquer inconveniente numa implementação por código SQL, mas tornam-se um processo mais dispendioso e mais trabalhoso. Ainda relativamente ao SQL a extração de dados deve ser efetuada de forma a obter resultados corretos. Estão identificadas diversas lacunas que se tornam verdadeiras armadilhas no procedimento de extração de dados operacionais utilizando esta linguagem e que estão diretamente relacionados com junções entre tabelas. As mais comuns são as relacionadas com a existência de caminhos múltiplos de pesquisa sobre dados, principalmente quando se pretende a obtenção de dados agregados.

No artigo publicado no WorlsCIST 2019, 7th World Conference on Information Systems and Technologies, Cavique (Cavique, Cavique, & Gonçalves, 2019). Descreve dois principais problemas que podem ocorrer no processo de desnormalização de bases de dados relacionais com a extração de tabelas de factos. Identifica em primeiro lugar a problemática de existirem caminhos alternativo entre tabelas de factos e tabelas de pesquisa, passando por tabelas intermédias, evidenciando assim caminhos múltiplos de acesso, que o operador *join* vai retornar valores diferentes, correspondendo às relações existentes nas tabelas intermédias. qual existem tabelas intermédias poderem retornar resultados diferentes quando se pretende retirar um facto. Considerando que uma única tabela intermédia pode não devolver todos os resultados esperados. Identifica também nas relações ternárias entre três tabelas em que existe uma relação muitos para muitos, (N:1 – 1:1 – 1:N) com recurso a uma tabela intermédia, ocorre uma inflacionamento nos dados agregados, que resulta da projeção natural do operador *join*, tornando os valores irreais. É apresentado duas formas de desnormalização de forma a evitar os caminhos múltiplos, uma primeira solução, designada por 1DF em que é proposto uma estratégia de divisão do caminho de acesso múltiplo com recurso a uma estratégia de divisão em dois caminhos distintos de pesquisa de forma a cada tabela ficar virtualmente conectada às tabelas intermédias e de pesquisa por um só caminho de acesso, evitando assim os caminhos múltiplos na pesquisa. Isto é conseguido por um processo de duplicação das tabelas onde se regista o acesso por diferentes caminhos; uma segunda técnica nomeada 2DF de desnormalização em que o objectivo é encontrar para cada facto a sua própria árvore de pesquisa. Sem caminhos duplos. E é conseguido identificando árvores independentes para cada facto identificado na base de dados, e resulta na divisão de toda a árvore do schema em diferentes árvores, de forma a que cada tabela de factos esteja relacionada com tabelas de pesquisa por caminhos de acesso independentes no que diz respeito às tabelas intermédias.

1.7. CUBOS E OLAP

Uma data warehouse é uma base de dados relacional. Apresenta, no entanto, uma subtileza funcional que a distingue das bases de dados relacionais que são projetadas para serem utilizadas nos processos de negócio e desenhadas para serem eficazes quando sujeitas a uma utilização exaustiva em processos transacionais que incluem

pesquisas, inserções ou alterações em um ou vários registos, conhecidas por serem bases de dados operacionais, OLTP systems (OnLine Transaction Processing systems).

A distinção de uma data warehouse para um sistema OLTP é precisamente a sua função: fornecer informação estratégica para a tomada de decisão. Para se conseguir este objetivo os dados relevantes deixam de ser os registos atualizados como os existentes numa base de dados operacional, mas sim dados históricos, sumários de dados e dados que se podem derivar dos que se encontram online. A estrutura de dados deixa de ser relevante para transações, e ganha importância que sejam projetadas e otimizados para consultas, que normalmente são efetuadas com uma frequência menor, mas com tempos de resposta grandes.

Uma estrutura de uma data warehouse dimensional apresenta-se como um sistema, nas quais os dados encontram-se relacionados entre valores e dimensões de conteúdos. Esta configuração apresenta um desenho lógico que permite que sejam consultados os dados com o detalhe disponível ou sumarizados, e esta facilidade de consulta torna-se uma boa fonte de alimentação para sistemas informacionais, conhecidos pela sigla OLAP do inglês OnLine Analytical Processing System cuja estrutura reflete os dados tratados de forma a poderem explorados sob diversas perspetivas. A esta nova forma de apresentar os dados que invoca a figura geométrica de um cubo designa-se por cubo OLAP, e tem a vantagem de ter os dados já organizados em células individuais, relacionadas com as dimensões relevantes para extração de informação base de suporte à decisão. Os dados num cubo OLAP podem ser explorados sob diversas perspetivas, serem fatiados, agrupados e sumarizados de forma mais rápida, e suportam de forma geral as seguintes operações:

Slice, operação que permite selecionar dados por dimensões;

Dice, operação que permite obter cubos menores através de extração de dados por dimensões;

Drill Down, operação que permite desagrupar dados de forma a se ir obtendo maior detalhe;

Roll Up, operação inversa de Drill Down.

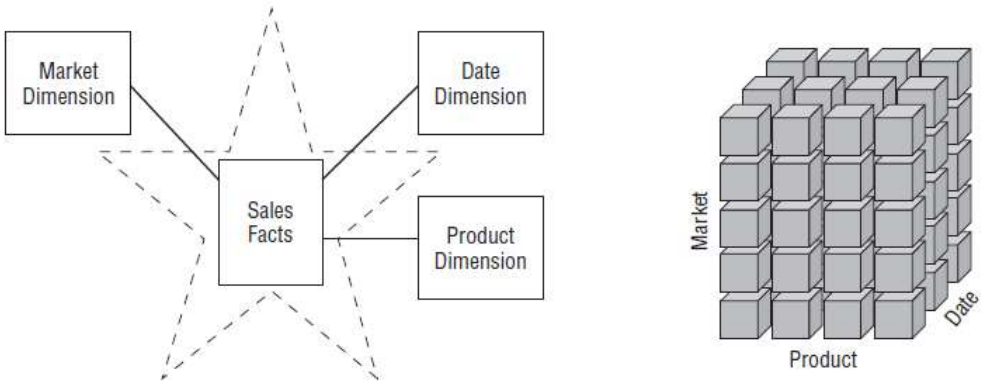

Figura 7: Star Schema versus cubo OLAP - fonte: (Kimball & Ross, 2002)

Os cubos OLAP são representações dos dados geridas por software que possuem motores próprios de indexação e agregação, estrutura essa que difere de fabricante para fabricante que concorrem entre si em inovação de forma a apresentar produtos capazes de oferecer melhores desempenhos na gestão de recursos de análise do que aqueles que são possíveis de retirar por consulta numa data warehouse que, apesar de ser vocacionada para a consulta, normalmente apresenta um volume de dados bastante elevado com tendência para crescer com o tempo, o que torna os processos de consulta, de agregação e de sumarização um processo mais demorado. Os cubos OLAP respondem com mais eficiência a estes problemas, pois podem ser estruturados com a sumarização ou detalhe pretendido para os dados factuais e relaciona-los diretamente com as dimensões relevantes para determinado tipo de questão, podendo serem oferecidos e ficarem disponíveis para exploração aos utilizadores finais de forma permanente.

Os cubos OLAP tem vindo a ganhar uma importância crescente na capacidade que têm de oferecer ao utilizador final a possibilidade de obter informação relevante para a tomada de decisão de forma mais célere e intuitiva, pois têm a possibilidades de serem construídos de forma a oferecer respostas às questões que costumam ser colocadas com maior frequência.

1.8. PRINCIPAIS CONCEITOS

Antes de continuarmos importa destacar alguns conceitos relacionados com a temática de data warehouse.

Data source, diz-se da origem de dados. Uma datawarehouse é uma base de dados relacional cujos dados podem ter origem em múltiplos repositórios, que podem ser de origens completamente dispares, como bases de dados relacionais originárias de diferentes SGD, bem como ficheiros de texto ou folhas de dados.

ETL, define-se como o processo pelo qual os dados passem do seu contexto de origem, para o contexto da data warehouse. A sua sigla em inglês designa o processo de extração (extract) dos dados das data sources, transformação (transform) qualitativa de forma a corresponderem aos requisitos da data warehouse, e carregamento (load) final em data warehouse.

Matriz/constelação, é o resultado final da implementação de data warehouse. Em termos simplistas uma tabela de facto dará origem a uma configuração em forma de estrela em que se pode observar um relacionamento com as tabelas que lhe fornece conteúdo. Quando os dados operacionais refletem a ocorrência de vários factos, uma matriz que relacione esses mesmos factos com as tabelas de dimensões fornece-nos a possibilidade de observar uma configuração formada por diversos conjuntos tipo estrela que partilham dimensões. Um desenho esquemático leva-nos a uma configuração com um formato a que se atribui a designação de constelação por analogia a um conjunto de diversas estrelas.

Consultas OLAP, método de analisar os dados. O objetivo de uma data warehouse é permitir a disponibilidade de dados relevantes para a tomada de decisão. Esses dados são tratados através de cubos agregadores e disponibilizados por relatórios resumidos disponibilizados através de consultas efetuadas sobre a os cubos de informação que se pode extrair dos dados disponíveis na data warehouse.

Capítulo 2 – Data Sources (Fontes dos dados)

Este trabalho desenvolve-se em torno de uma empresa que se dedica ao aluguer de vídeos, que será designada por VideoX.

Pretende implementar uma *Data Warehouse* que responda às necessidades da empresa de analisar o seu histórico e possuir uma ferramenta que lhe permita a tomada de decisão com bases solidas.

Como Data Sources, observamos a existência de três bases de dados relacionais;

Em primeiro lugar temos a base de dados de aluguer de vídeos, na qual se verifica a existência de uma relação de filmes disponíveis em dois estabelecimentos físicos, cada qual com a sua relação própria. Os filmes encontram-se catalogados por género e possuem diversas informações associadas de interesse, tal como ano de produção, atores intervenientes, custo de reposição, etc. Estes são alugados por clientes aos quais são cobrados o serviço disponibilizado, sendo este processo efetuado por um dos vários funcionários que a empresa dispões nas lojas físicas.

Depois existe duas bases de dados que foram implementadas na sede. Uma base de dados em que consta principalmente a aquisição de filmes, com relevância para quais foram os fornecedores e o funcionário responsável pela aquisição, e o preço de aquisição. Os contactos que são efetuados também constam desta base de dados, e o responsável pela sua implementação e gestão está relacionado com o departamento de aquisições; uma segunda base de dados cuja responsabilidade de gestão compete ao departamento de recursos humanos, e possui os dados biográficos de todos os funcionários.

2.1. REFERÊNCIA PARA O ESTUDO DE CASO

Luís Cavique, et al. num artigo publicado no WorlsCIST 2020, 8th World Conference on Information Systems and Technologies (Cavique, Cavique, & Santos, 2020) propõe um processo para extração de uma data warehouse de diversas bases de dados relacionais,

decompondo-as através de uma visão botton-up, objectivando a criação de um esquema de constelação (constelation schema).

Na extração de um data mart importa identificar a medida que lhe está subjacente e construir uma árvore sem caminhos duplicados para as tabelas que lhes proporcionam determinado significado. No sitio da internet "Kimbal Group" (Group, s.d.), uma equipa de especialistas em projetos de data warehouse define os factos em três categorias, relacionados com a existência de valores numéricos a eles associados. Estas categorias são classificadas como:

- Factos aditivos, quando os valores numéricos disponíveis podem ser somados em todas as dimensões com que se relacionam;

- Factos semi-aditivos, quando os valores numéricos numa tabela de factos podem ser somados em algumas das dimensões com que se relacionam, mas não em todas;

- Factos não aditivos, quando estes não podem ser somadas em relação às dimensões com que se relacionam. No entanto proporcionam interesse suficiente para serem utilizados em funções agregadoras, com por exemplo através de contagem de objetos.

Depreende-se ainda uma quarta categoria em que não ocorre a existência de um facto que se enquadre nesta definição uma vez que os dados disponibilizados não oferecem qualquer tipo de medida plausível se lhe ser atribuída.

A identificação dos factos é conseguida pela construção de tabelas de resumo onde se identifica o tipo de tabela e das medidas presentes, e pela elaboração de uma matriz que faz a associação entre os factos apurados e as dimensões que lhes fornecem conteúdo.

Parte deste trabalho terá como base os processos e princípios descritos nos artigos mencionados, nomeadamente na base à extração das tabelas de factos e dimensões das bases de dados.

2.2. ESTRUTURA DE DADOS

A VideoX possui na sua organização orgânica 3 departamentos, que gerem de forma independente 3 bases de dados transacionais.

O departamento cuja responsabilidade cabe o registo de alugueres de vídeo, utiliza uma réplica da base de dados sakila sample database, disponível no sitio da internet https://dev.mysql.com/doc/sakila/en/ (Oracle, 2020)

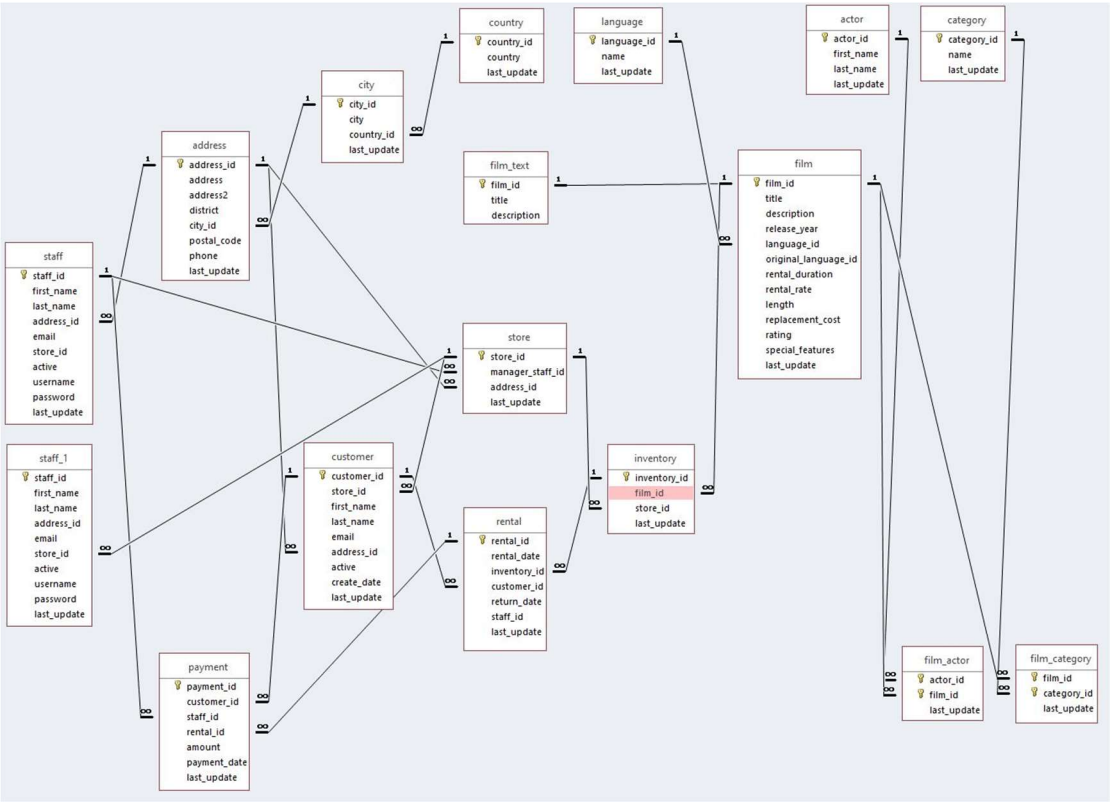

Figura 8: sakila Schema, (fonte própria)

Tabela 1: Tipo de tabelas no schema sakila

table name	type of	type of facts
country	lookup	
language	lookup	
actor	lookup	
category	lookup	
city	intermediate	
address	intermediate	
film	intermediate	
staff	intermediate	
store	intermediate	
customer	intermediate	
inventory	intermediate	
rental	intermediate	
film_text	fact	without facts
payment	fact	additive
film_actor	fact	without facts
film_category	fact	without facts

Tabela 2 : Tabela de factos do schema sakila

table name	payment	film_text	film_actor	film_category
country	X			
language	X	X	X	X
actor	X		X	
category	X			X
city	X			
address	X			
film	X	X	X	X
staff	X			
store	X			
customer	X			
inventory	X			
rental	X			
time	X	X	X	X

O departamento de recursos humanos, possui uma base de dados na qual se regista os funcionários, os departamentos a que estão afetos, os dias e os motivos de ausência bem como outras indicações pessoais, tal como contactos.

O schema da base de dados é representado pelo diagrama da Figura 9: Schema employees do departamento recursos humanos.

que se reproduz em anexo B com maior detalhe. Este schema é de elaboração própria, com influência no schema disponível no sítio da internet https://dev.mysql.com/doc/employee/en/.

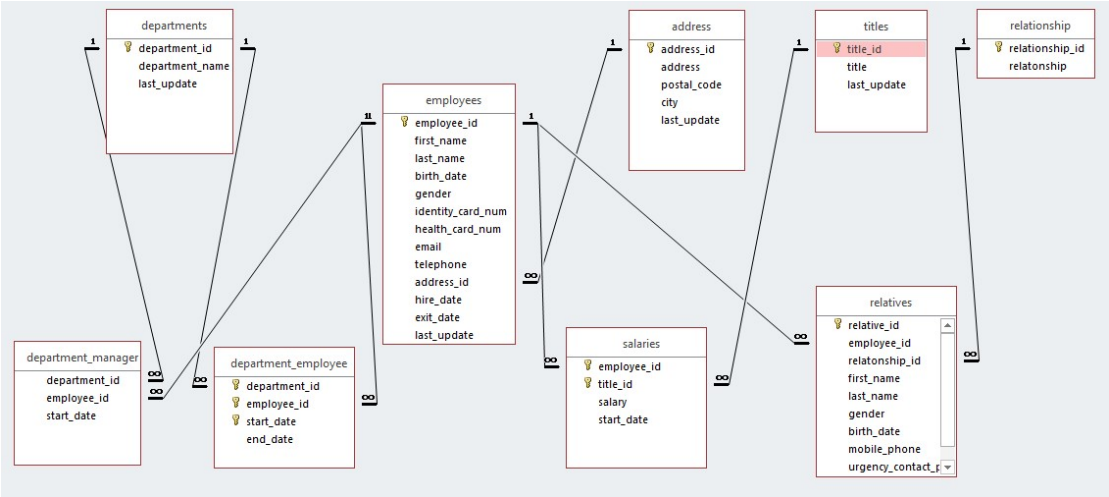

Figura 9: Schema employees do departamento recursos humanos.

Tabela 3: Tipo de tabelas no schema employee

table name	type of table	type of facts
departments	lookup	
address	lookup	
titles	lookup	
relatioship	lookup	
employees	intermediate	
department_manager	fact	without facts
department_employee	fact	without facts
salaries	fact	additive
relatives	fact	without facts

Tabela 4: Tabela de factos do schema employee

table name	department_manager	department_employees	salaries	relatives
departments	X			
address	X	X	X	X
titles	X		X	
relatioship	X			X
employees	X			
time	X	X	X	X

O departamento de recursos aquisições, possui uma base de dados de elaboração própria, onde se mantém um registo dos vídeos adquiridos, referenciando o responsável pela aquisição, assim como um registo de contactos efetuados com os fornecedores.

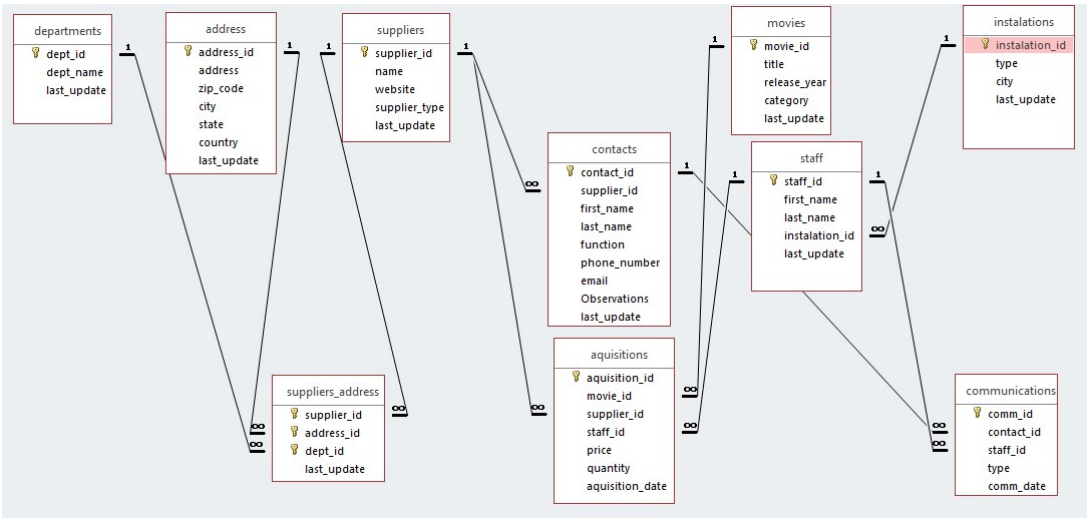

Figura 10: Schema suppliers do departamento recursos aquisições

Tabela 5: Tipo de tabelas no schema suppliers

table name	type of table	type of facts
departments	lookup	
address	lookup	
suppliers	lookup	
movies	lookup	
instalations	lookup	
contacts	intermediate	
staff	intermediate	
suppliers_address	fact	without facts
aquisitions	fact	additive
communications	fact	non-additive

Tabela 6: Tabela de factos do schema suppliers

table name	suppliers_address	aquisitions	communications
departments	X		
address	X		
suppliers	X	X	X
movies		X	
instalations		X	X
contacts			X
staff		X	X
time	X	X	X

Com esta informação, irá proceder-se ao estudo de caso de implementação da data warehouse da VideoX.

2.3- ETL

O processo de ETL (Extraction, Transform and Load) fundamentalmente em carregar em DW os objectos necessários de modo a satisfazer os objectivos propostos para a DW. Assim os objetos são retirados das fontes disponíveis, transformados no objetivo proposto e carregados em DW.

Para a extração das tabelas de factos, é necessário evitar os erros que podem ser originados por "multiple-acess path problems" e "fan-trap" . Luís Cavique (Cavique, Cavique, & Gonçalves, 2019) propõe duas formas de normalização para evitar a ocorrência desses erros.

Impõe-se uma análise do processo de extração de dados às tabelas de factos que foram consideradas como relevantes após aplicação do procedimento de extração de tabelas de factos conjugado com os requerimentos da matrix de oferta procura., de modo a definir se existe a necessidade de proceder à desnormalização das tabelas de factos resultantes.

Da base de dados sakila uma análise de extração de dados da tabela payment tem-se o seguinte schema relacionado:

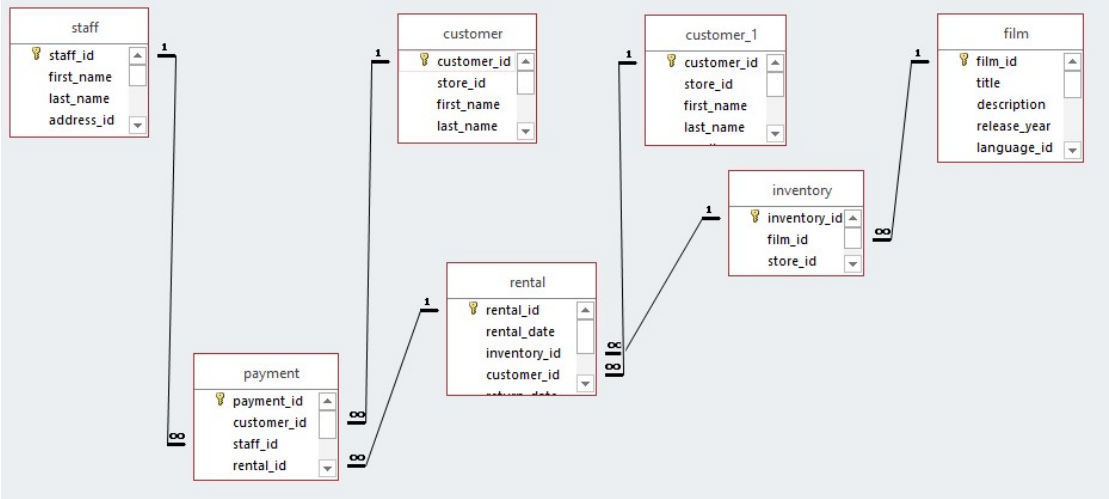

Figura 11 : Árvore de extração do facto payment

Do mesmo modo, da tabela film_actor

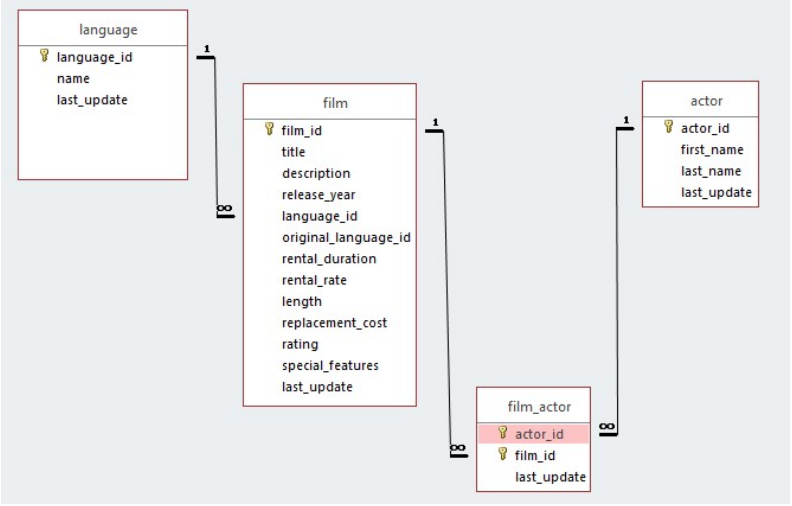

Figura 12 : Árvore de extração do facto film-actor

Assim como da tabela film_category

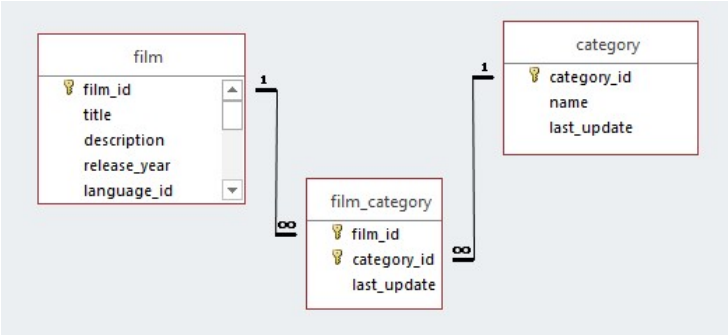

Figura 13 : Árvore de extração do facto film_category

Observa-se a que a possibilidade de ocorrer múltiplos caminhos de pesquisa na tabela *payment*, nomeadamente nas pesquisas que possam envolver o caminho "*payment-customer*" ou "*payment-rental-customer*", pelo que existe a necessidade de considerar o procedimento de desnormalização de forma a desacoplar a árvore acíclica resultante, através de uma cópia da tabela *customer*, obtendo a seguinte configuração com a configuração típica de uma árvore bem definida no seu percurso de pesquisa.

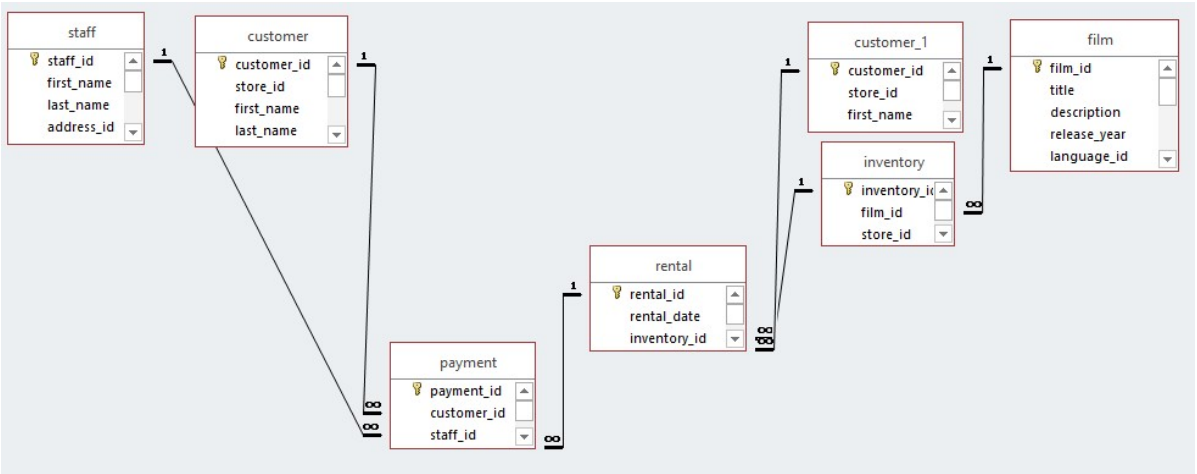

Figura 14 : Árvore de extração do facto payment, desnormalizada

Agregando a informação obtida das configurações de extração de dados das tabelas de factos, relativamente à base de dados sakila obtemos uma arvore de pesquisa totalmente desacoplado com a seguinte configuração:

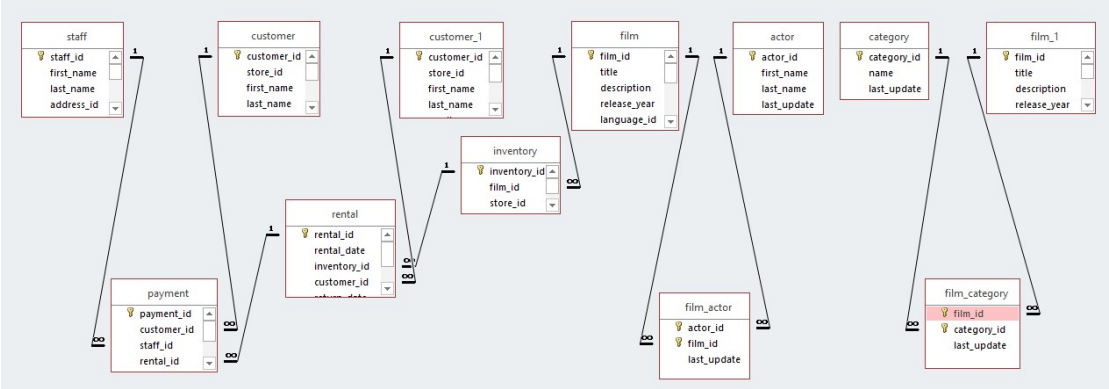

Figura 15 : Árvore de extração de factos da base de dados sakila

Esta análise, estendida à de dados *suppliers*, as configurações são representadas pelos seguintes esquemas:

Da tabela *aquisitions*

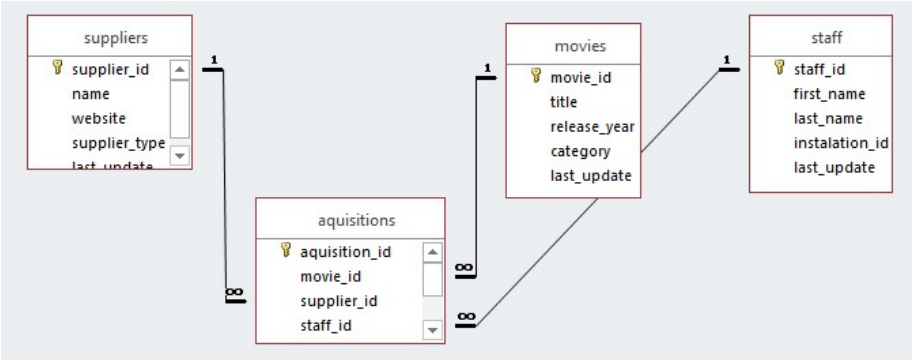

Figura 16 : Árvore de extração do facto aquisitions

Da tabela communications:

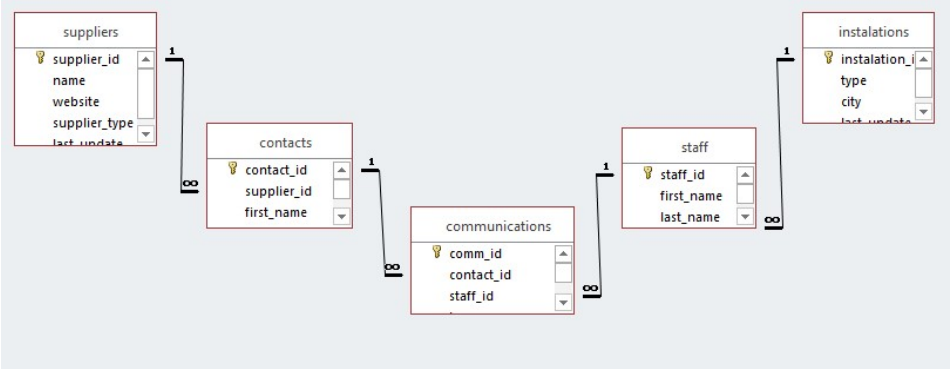

Figura 17: Árvore de extração do facto communications

Verifica-se nesta base de dados, que a extração de dados relativos às tabelas de factos apuradas que não existe a necessidade de se proceder a qualquer procedimento de desnormalização, uma vez que as configurações oferecidas se encontram bem definidas com forma de árvore de pesquisa sem caminhos duplos.

Resumindo obtemos o seguinte Schema:

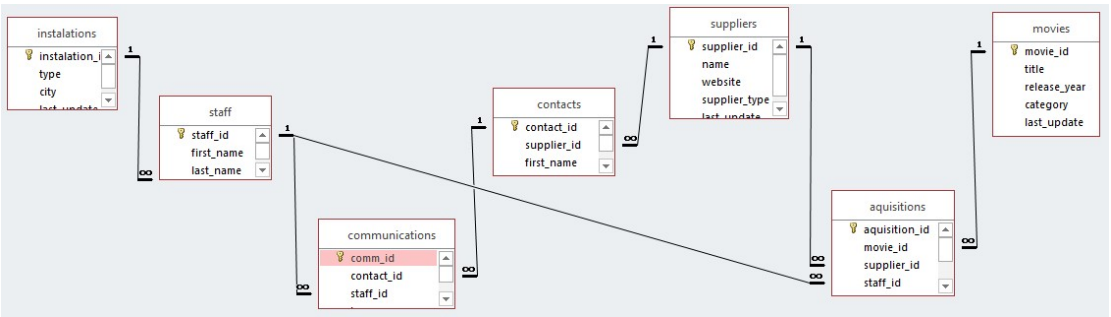

Figura 18: Árvore de extração dos factos da bd suppliers

No qual se verifica a possibilidade de poder ocorrer um caminho duplo de pesquisa, pelo que há que considerar a denormalização nas tabelas *staff* e *instalations*.erifica

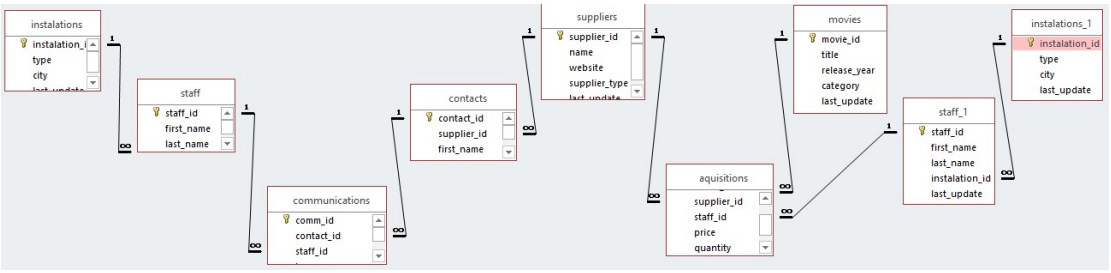

Figura 19 : Árvore de extração de factos da bd suppliers desnormalizada

Na base de dados employees, a mesma análise deve ser efetuada, sobre as tabelas de factos relative e salarie, cuja Schema de extração são os seguintes.

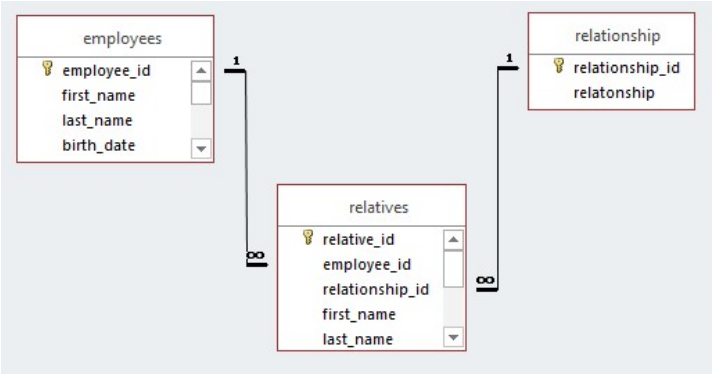

Figura 20 : Árvore de extração do facto relatives

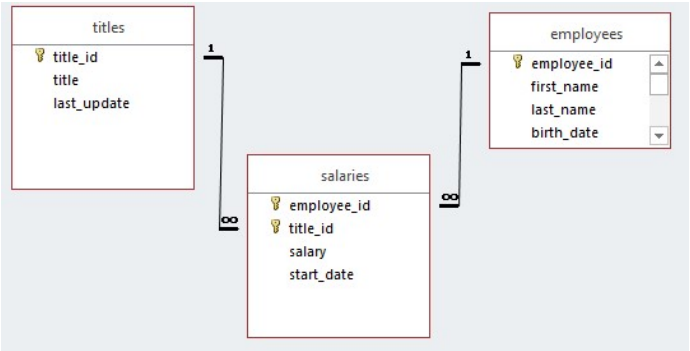

Figura 21 : Árvore de extração do facto salaries

Na extração de dados sobre ambos os factos relevantes desta base de dados, não se identifica estruturas que possibilitem caminhos cruzados nas pesquisas, observando-se o seguinte formato geral.

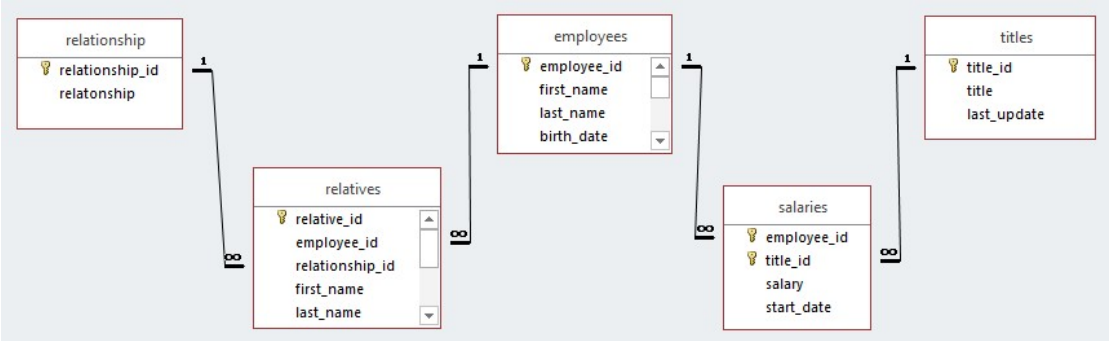

Figura 22: Árvore de extração dos factos da base de dados employees

Para determinar a forma como os objetos passam por todo este caminho de extração, transformação e carregamento em data warehouse, uma forma de os visualizar é por meio de gráficos de mapeamento. Quando se recorre a software proprietário e dedicado a estes processos os mapeamentos normalmente são disponibilizados de forma a tornar visualmente entendível este processo; existe também a necessidade de documentar todo este processo, e tornar acessível aos futuros intervenientes uma fácil compreensão e acessibilidade. Para este facto é importantíssimo a possibilidade de aceder de forma espontânea à informação disponível, pelo que documentar todo este processo do que em tabelas que coexistirão em paralelo com a datawarehouse, em que se registarão todo o processo de ETL. O Schema torna-se assim parte integrante da datawarehouse, possibilitando a consulta de todo o processo.

As consultas correspondentes estão descritas no manual de utilização em anexo a este relatório.

Apresenta-se o processo de ETL dos objetos da DW, de forma esquematizada, sendo de considerar em comum as seguintes anotações:

- Os campos ***last_update*** são extraídos para as tabelas correspondentes, e serão utilizados para comparações de diferentes extrações. Normalmente mantem-se as últimas extrações guardadas para referência e poderão ser utilizados nos processos de seleção de dados a selecionar para serem introduzidos em DW.

- Nas extrações é criado um campo ***extract_date***, tipo date/time, que será carregado em DW no campo ***load_date***, também do tipo date/time

- Os campos chave primária são campos numéricos automáticos.

Tabela actor (dimensão)

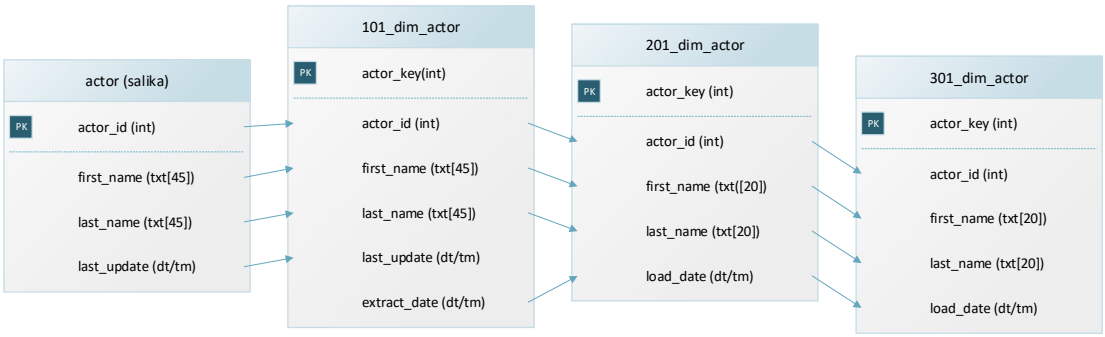

Figura 23 : ETL da tabela actor (dimensão)

Extract: Extração dos campos *actor_id*, *first_name*, *last_name* e *last_update* da tabela *actor* (sakila)

Transform: first_name e last_name para campos de 20 carateres.

Load: carregamento direto após transformação.

Tabela customers (dimensão)

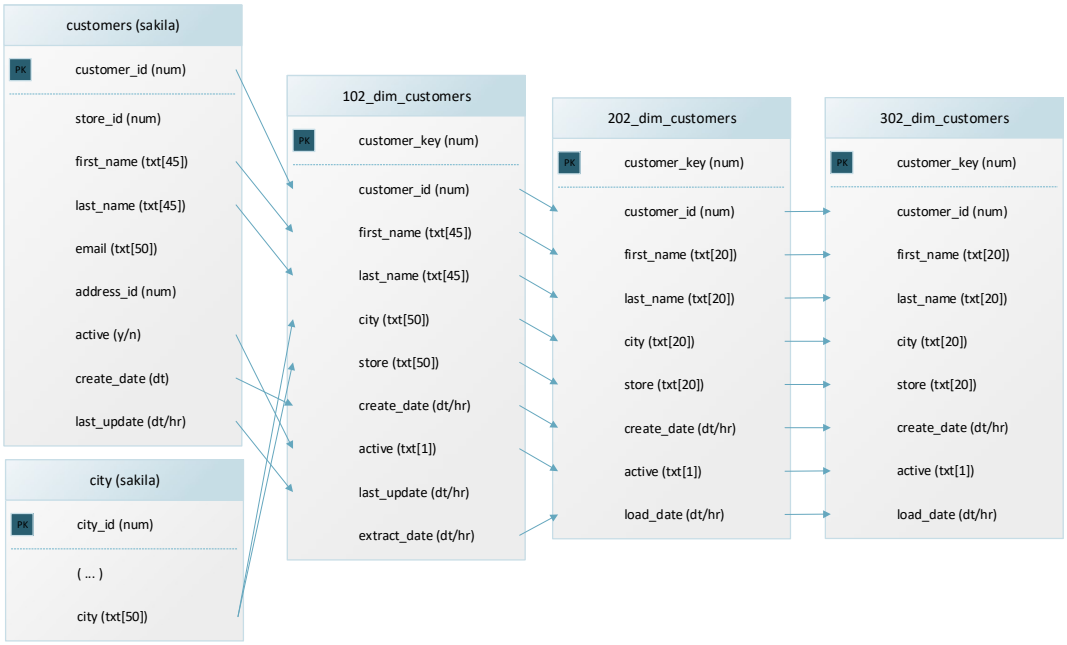

Figura 24 : ETL da tabela customers (dimensão)

Extract: Extração dos campos *customer_id*, *first_name*, *last_name*, *active*, *create_date* e *last_update* da tabela *customers* (sakila); e do campo *city* da tabela *city* (sakila).

Na base de dados sakila, não existe propriamente um nome para a loja. Para se obter uma designação optou-se pela designação da localidade onde a loja está implementada alojando essa informação no campo *store*; esse campo é obtido pela relação das tabelas *city-address-store-customer* na base de dados sakila. O campo city refere-se à localidade do cliente e é obtido pela relação das tabelas *city-address-customer*.

Transform: Os campos *first_name* e *last_name* e *city* são reduzidos para o tamanho de 20 caracteres;

O campo *active* é transformado de um campo *yes/no* para um campo de texto com 1 caractere.

Load: carregamento direto após transformação.

Tabela film (dimensão)

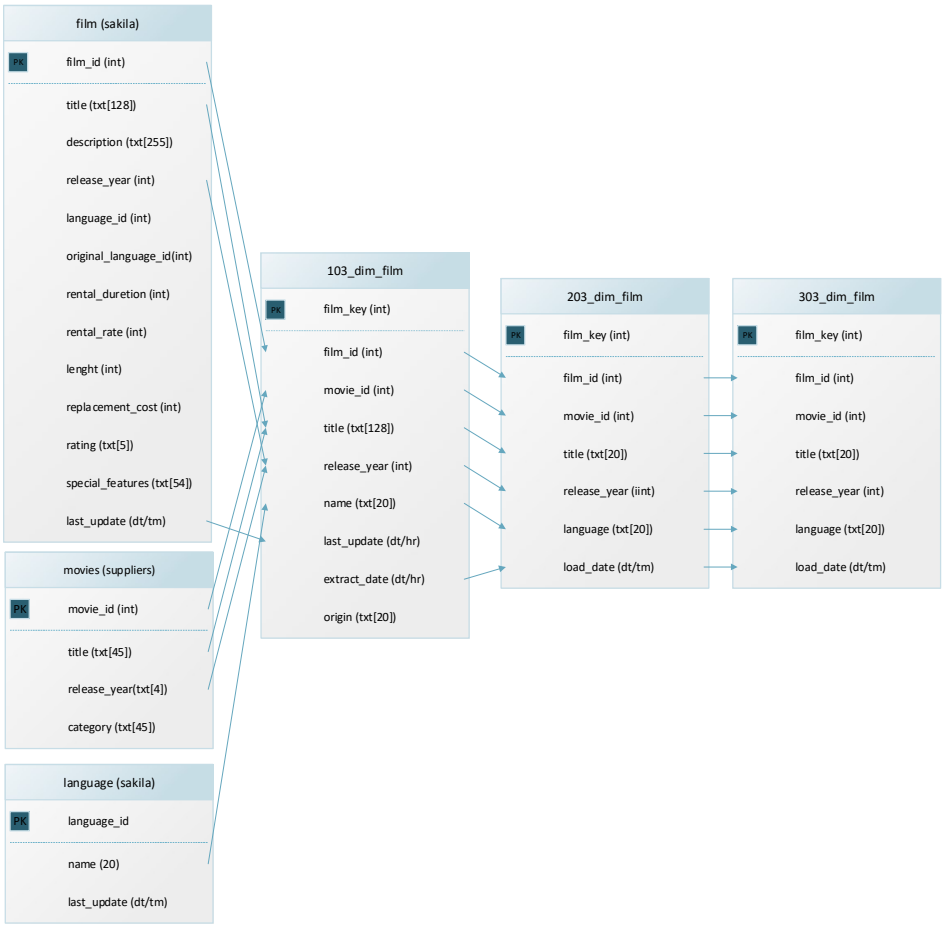

Figura 25 : ETL tabela film (dimensão)

Extract:

A extração de dados para a tabela film é efetuada de duas tabelas com origem em bases de dados diferentes: *film* (*sakila*) e *movie* (*suppliers*), para uma só tabela, onde são registados os campos de interesse de cada uma das tabelas, ficando por preencher em cada registo os campos que não tem origem na tabela de extração correspondente.

Extração do campos *film_id, title, release_year, last_update* da tabela *film* (sakila), e *name* da tabela (*language*) através da relação *film-language*;

Extração do campos *movie_id, title, release_year, last_update* da tabela *movies* (suppliers);

Em cada extração é preenchido o campo ***origin*** com o nome da tabela de origem

Transform:

Como os dados tiveram origem em duas tabelas diferentes, a principal transformação é efetuar um agrupamento por nome do filme, e obter o preenchimento dos campos não comuns. Por sua vez origina a necessidade de efetuar confirmações nomeadamente nos

campos comuns das duas tabelas de origem, uma vez que pode existir erros de ortografia no nome do filme, o que iria originar dois filmes diferentes.

O campo *name* é renomeado em *language* para atribuição do significado da tabela de origem;

O campo *title* é reduzido para o tamanho de 20 caracteres;

Load: carregamento direto após transformação.

Tabela suppliers (dimensão)

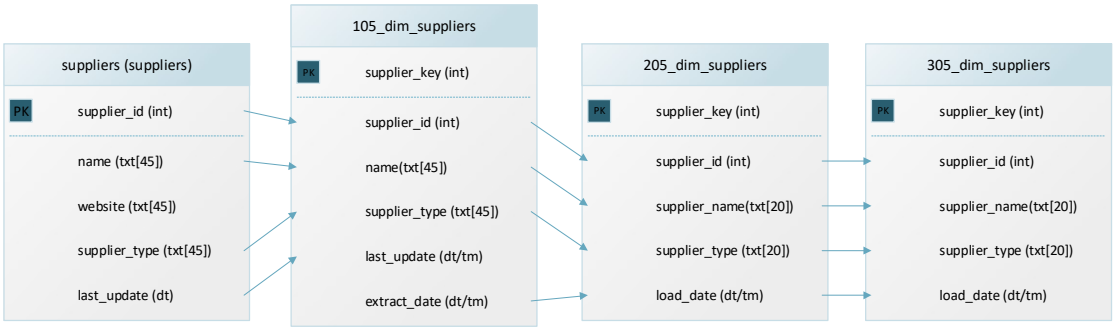

Figura 26: ETL da tabela suppliers (dimensão)

Extract : Extração dos campos *supplier_id, name, suplier_type e last_update* da tabela *suppliers* (suppliers);

Transform: Transformação do nome do campo *name* para *supplier_name*.

Redução do tamanho dos campos *suplier_name, supplier_type* para uma dimensão máxima de 20 caracteres.

Load: Carregamento direto após transformação.

Tabela staff (dimensão)

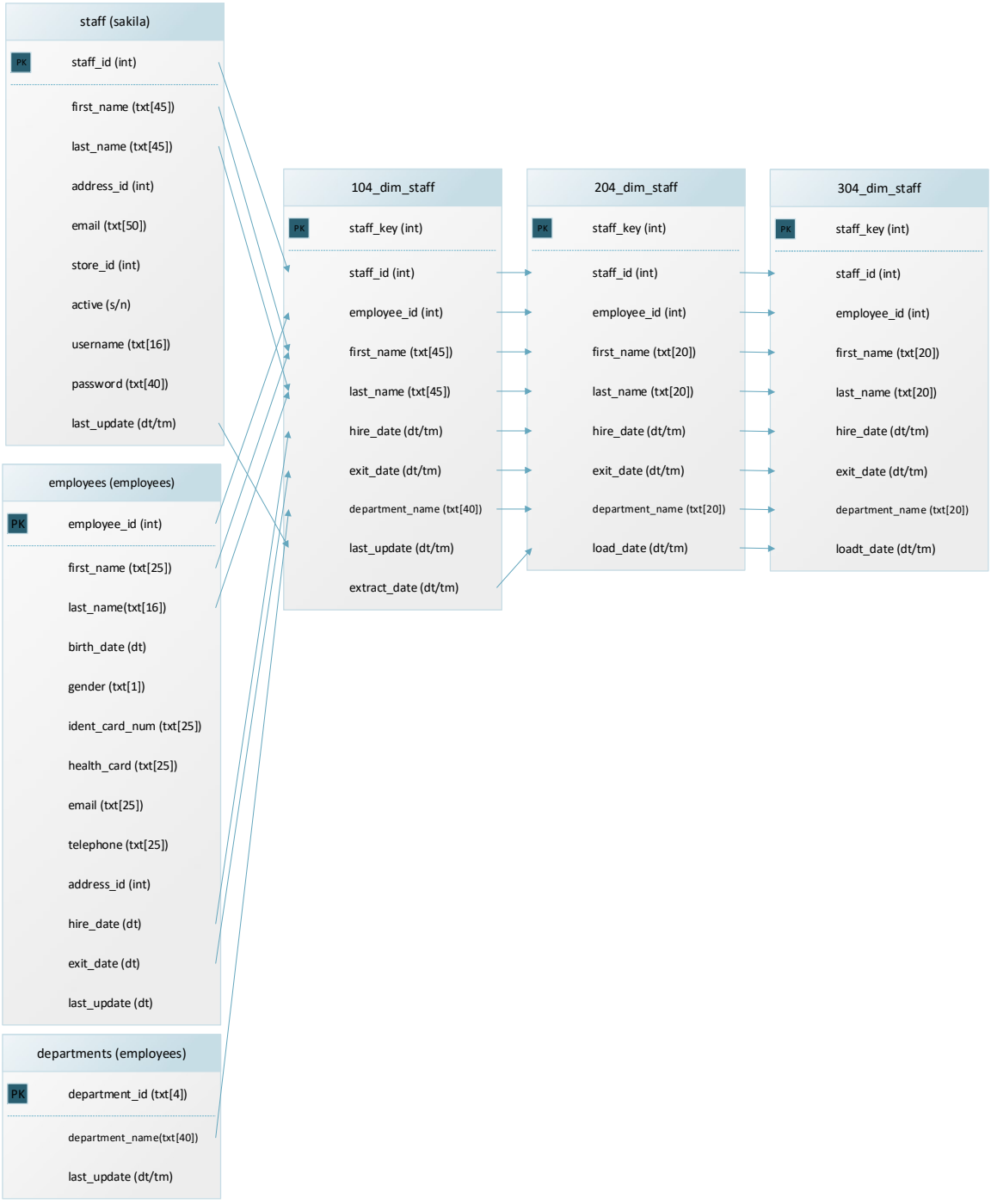

Figura 27 : ETL da tabela staff (dimensão)

Extract : Extração dos campos *employee_id*, *first_name*, *last_name*, *hire_date*, *exit_date*, *last_update* da tabela *employees* (*employees*) e *department_name* da tabela *departments(employees)* pela relação *employees-department_employee-departments*;

Extração dos campos staff_id, first_name, last_name, last_update da tabela staff (sakila)

Transform: Os campos first_name, last_name e department_name são reduzidos no seu tamanho para um máximo de 20 caracteres.

Os campos *first_name* e *last_name* são agrupados. Retirado o maior valor nos restantes campos.

Requer verificação de conformidade nos campos que tem origem em tabelas diferentes, nomeadamente *first_name* e *last_name*.

Load: carregamento direto após transformação.

Tabela rental (dimensão)

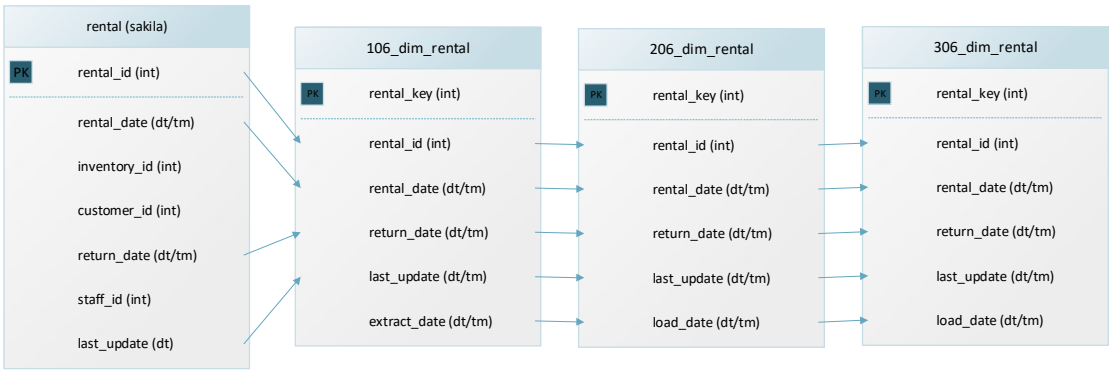

Figura 28 : ETL da tabela rental (dimensão)

Extract : Extração dos campos *rental_id, rental_date, return_date e last_update* da tabela rental (sakila);

Transform: Nada é efetuado.

Load: Carregamento direto após transformação.

Tabela category (dimensão)

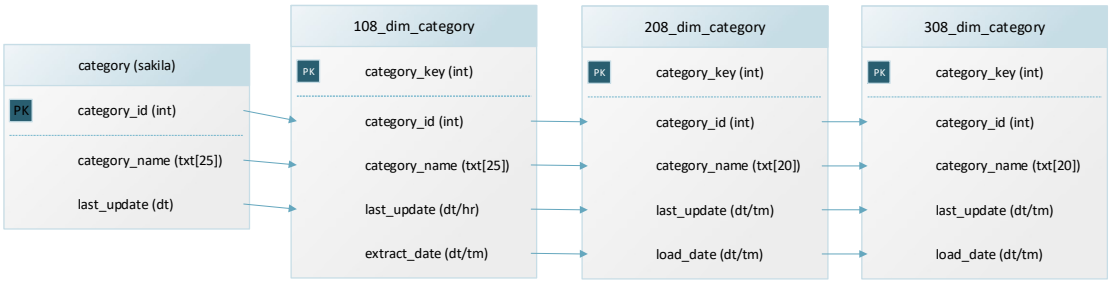

Figura 29: ETL da tabela category (dimensão)

Extract : Extração dos campos *category_id, category_name e last_update* da tabela *category* (sakila);

Transform: Redução do tamanho do campo *category_name* para uma dimensão máxima de 20 caracteres.

Load: Carregamento direto após transformação.

Tabela calendar (dimensão)

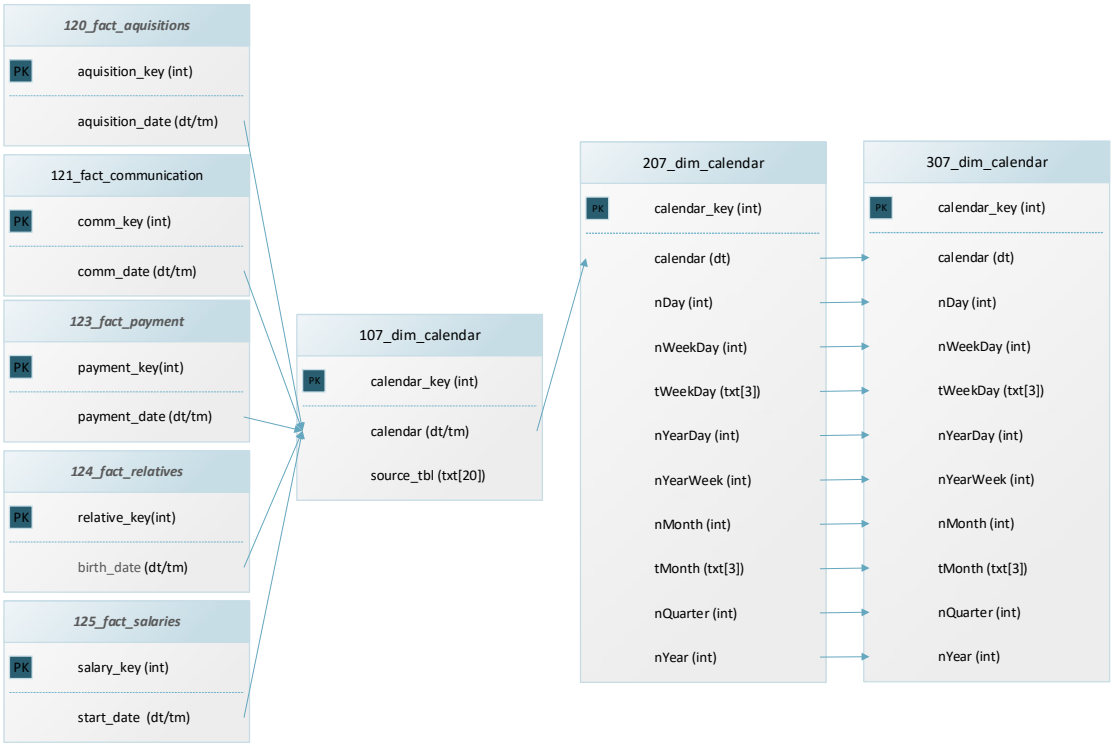

Figura 30 : ETL da tabela calendar (dimensão)

Extract: Para o campo *calendar* é efetuada a extração dos seguintes campos, e o nome da tabela de origem:

- *aquisition_date* da tabela *aquisitions*(*suppliers*);

- *payment_date* da tabela *payment*(*sakila*);

- *comm_date* da tabela *communications*(*suppliers*);

- *start_date* da tabela *salaries*(*employees*);

- *birth_date* da tabela *relatives (employees);*

Transform: O campo calendar é definido pelo grão mínimo igual ao "dia" como o pormenor que se pretende para a DW.

Por funções disponíveis no Microsoft Access são criados a partir do campo calendar os seguintes campos:

- *nDay*, para albergar o número do dia;

- *nWeekDay*, para registo do dia da semana em formato numérico;

- *tWeekDay*, para registo do dia da semana em formato de texto;

- *nYearDay* para registo do dia do ano em formato numérico;

- *nWeekYear* para registo da semana relativamente ao ano em formato numérico;

- *nMonth* para registo do mês no formato numérico;

- *tMonth* para registo do mês no formato de texto;

- *nQuarter* para registo do trimestre no formato numérico;

- *nYear* para registo do ano no formato numérico.

Load: Carregamento direto após transformação.

Tabela film_actor (facto)

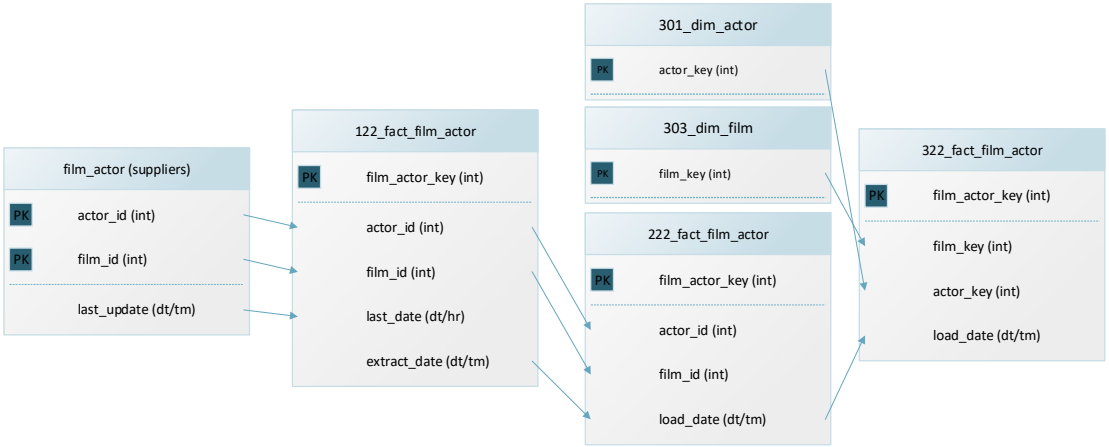

Figura 31 : ETL da tabela film_actor (facto)

Extract : Extração dos campos *actor_id, film,* e *last_update* da tabela *film_actor* (*suppliers*);

Transform: Nada a salientar.

Load: O campo *actor_id* é substituído em DW pelo campo *actor_key* correspondente retirado da relação com a tabela de dimensão *dim_actor* da DW.

O campo *film_id* é substituído em DW pelo campo *film_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

Tabela aquisitions (facto)

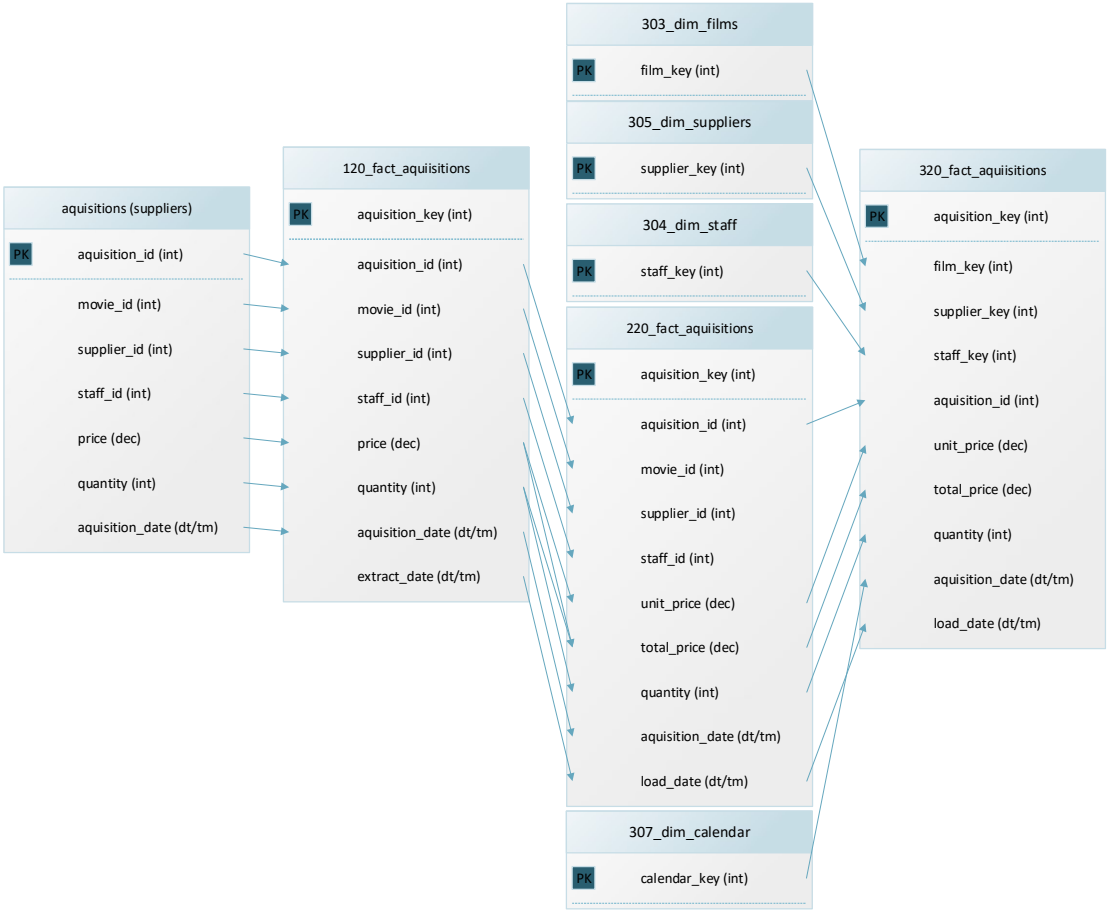

Figura 32 : ETL da tabela aquisitions (facto)

Extract: Extração dos campos *aquisition_id, movie_id, supplier_id, staff_if, price, quantity, e aquisition_date* da tabela aquisitions (suppliers);

Transform: Nada a considerar.

Load: O campo *movie_id* é substituído em DW com o campo *film_key* correspondente retirado da relação com a tabela de dimensão *dim_films* da DW.

O campo *supplier_id* é substituído em DW pelo campo *supplier_key* correspondente retirado da relação com a tabela de dimensão *dim_suppliers* da DW.

O campo *staff_id* é substituído em DW pelo campo *staff_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

O campo *aquisition_date* é substituído em DW pelo campo *aquisition_date_key* correspondente retirado da relação com a tabela *dim_calendar* da DW.

Tabela communications (facto)

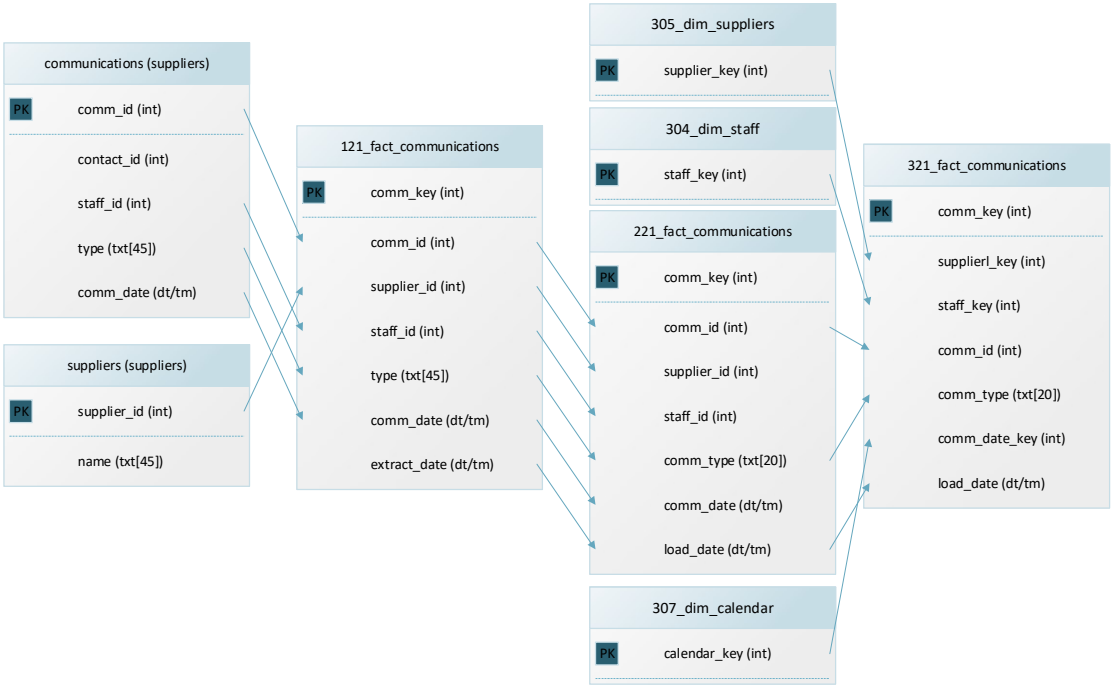

Figura 33 : ETL da tabela communications (facto)

Extract: Extração dos campos *comm_id, staff_id, type,* e *comm_date* da tabela *communicatiosn* (*suppliers*);

Extração do campo *supplier_id* da tabela *suppliers*(*suppliers*) pela relação *communications-contacts-suppliers.*

Transform: O campo *type* é renomeado em *comm_type* para atribuição de significado da tabela de origem e passa a ter um tamanho máximo de 20 caracteres.

Load: O campo *supplier_id* é substituído em DW pelo campo *supplier_key* correspondente retirado da relação com a tabela de dimensão *dim_suppliers* da DW.

O campo *staff_id* é substituído em DW pelo campo *staff_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

O campo *comm_date* é substituído em DW pelo campo comm_*date_key* correspondente retirado da relação com a tabela *dim_calendar* da DW.

Tabela payment (facto)

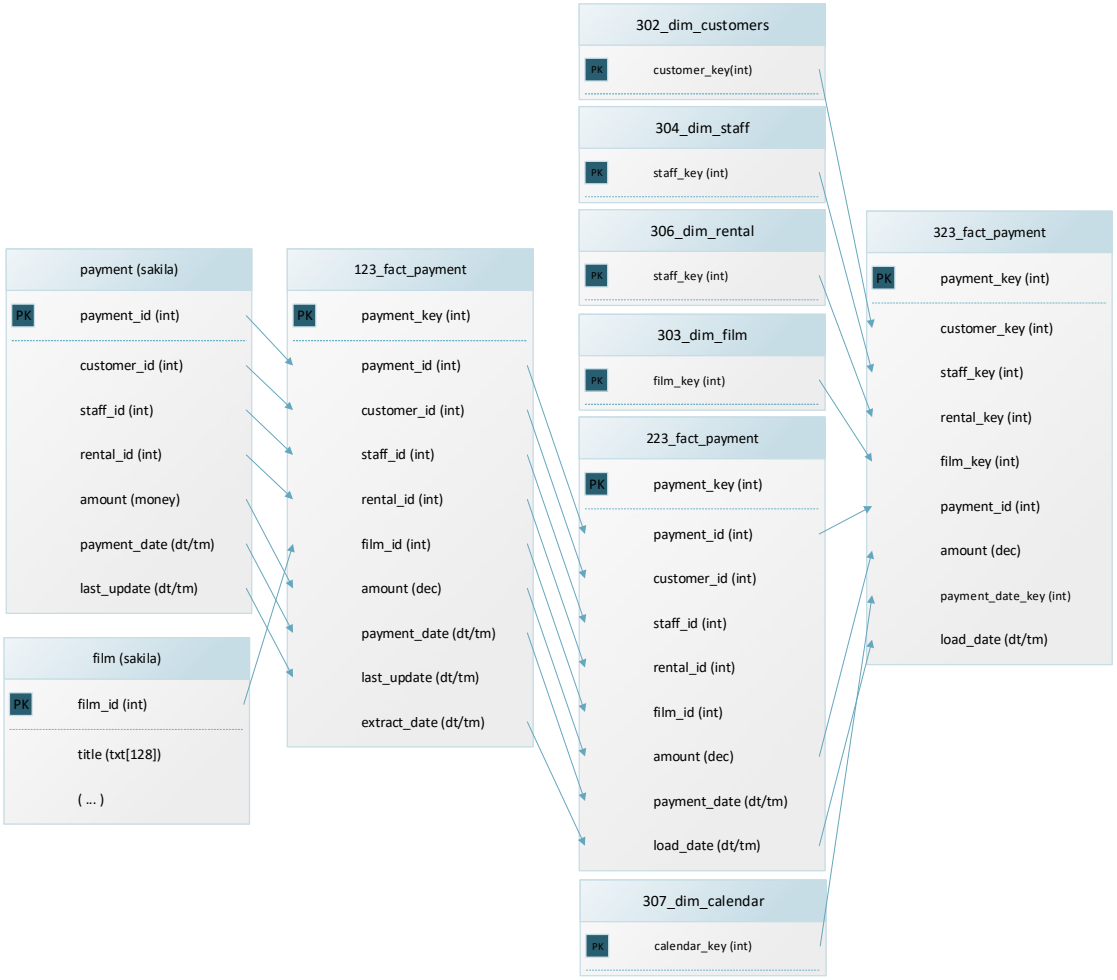

Figura 34 : ETL da tabela payment (facto)

Extract: Extração dos campos *payment_id, customer_id, staff_id, rental_id, amount, payment_date e last_update* da tabela *payment* (*sakila*);

Extração do campo *film_id* da tabela *inventory*(*sakila*) pela relação *payment-rental-inventory*.

Transform: Nada a registar.

Load: O campo *customer_id* é substituído em DW pelo campo *customer_key* correspondente retirado da relação com a tabela de dimensão *dim_customer* da DW.

O campo *staff_id* é substituído em DW pelo campo *staff_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

O campo *rental_id* é substituído em DW pelo campo *rental_key* correspondente retirado da relação com a tabela de dimensão *dim_rental* da DW.

O campo *film_id* é substituído em DW pelo campo *film_key* correspondente retirado da relação com a tabela de dimensão *dim_film* da DW.

O campo *payment_date* é substituído em DW pelo campo *payment_date_key* correspondente retirado da relação com a tabela *dim_calendar* da DW.

Tabela relatives (facto)

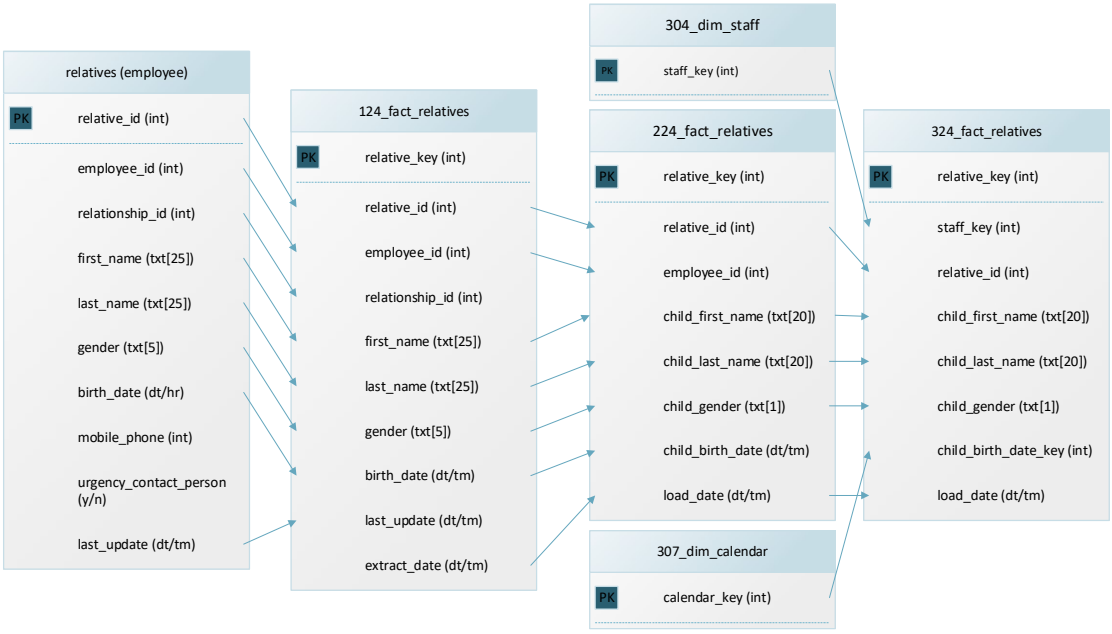

Figura 35 : ETL da tabela relatives (facto)

Extract: Extração dos campos *relative_id, employee_id, relationship_id, first_name, last_name, gender, birth_date e last_update* da tabela *relatives* (*employee*);

Transform: Os campos first_name, last_name e birth_date são renomeados como child_first_name, child_last_name e child_birth_date com tamanho máximo de 20, 20 e 1 caratere respetivamente.

Só são considerados os filhos dos funcionários, isto é, aqueles registos onde o *relationship_id* é 3 ou 4.

O Campo relationship_id deixa de fazer sentido para carregar em DW, uma vez que só vai interessar os filhos dos funcionários.

Load: O campo *staff_id* é substituído em DW pelo campo *staff_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

O campo *child_birth_date* é substituído em DW pelo campo *child_birth_date_key* correspondente retirado da relação com a tabela *dim_calendar* da DW.

Tabela salaries (facto)

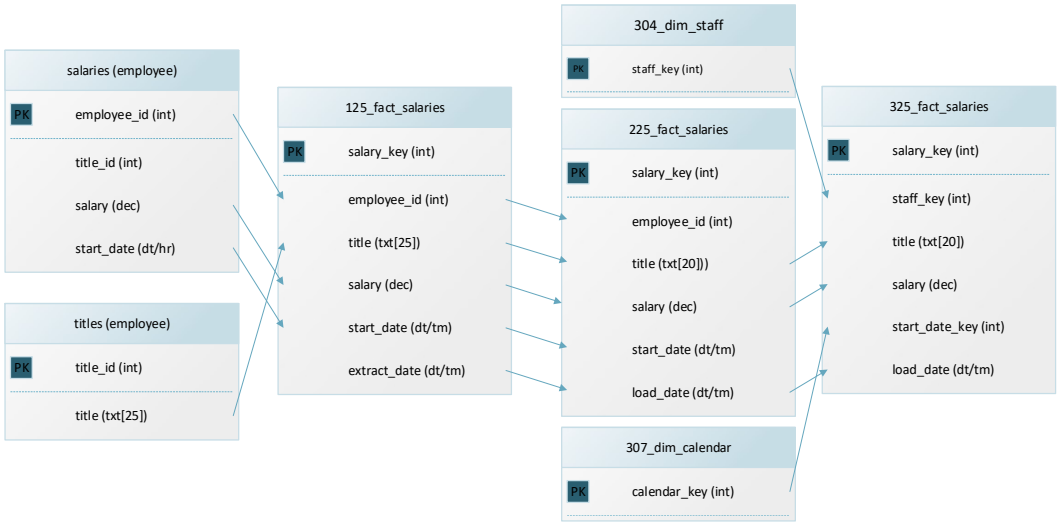

Figura 36: ETL da tabela salaries (facto)

Extract: Extração dos campos *employee_id, salary, start_date* da tabela *salaries* (employee);

Extração do campo *title* da tabela *titles*(*employee*) pela relação *salaries-titles.*

Transform: Redução do tamanho do campo *title* para uma dimensão máxima de 20 caracteres.

Load: O campo *staff_id* é substituído em DW pelo campo *staff_key* correspondente retirado da relação com a tabela *dim_staff* da DW.

O campo *start_date* é substituído em DW pelo campo *start_date_key* correspondente retirado da relação com a tabela *dim_calendar* da DW.

Tabela film_category (facto)

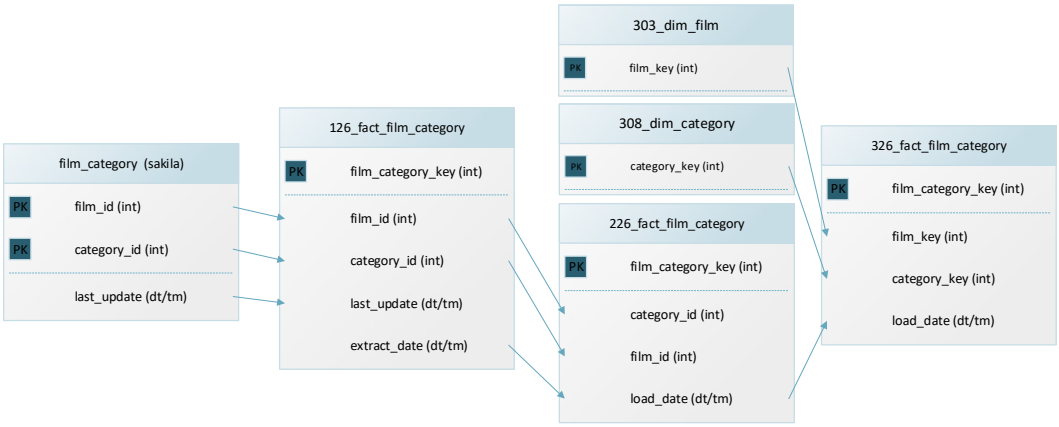

Figura 37 : ETL da tabela film_category (facto)

Extract: Extração dos campos *film_id, category_id e last_update* da tabela *film_category* (*sakila*);

Transform: Nada a registar.

Load: O campo *film_id* é substituído em DW pelo campo *film_key* correspondente retirado da relação com a tabela de dimensão *dim_film* da DW.

Dos tipos de factos apurados constata-se que a tabela *payment*(*sakila*) disponibiliza o campo *amount como* passível de ser somado e por essa razão o consideramos um facto aditivo; pela mesma razão a tabela salaries(*employees*) oferece-nos o campo salaries como um facto aditivo; bem como a tabela *aquisitions*(*suppliers*) onde os campos *price* e *quantity* também nos disponibiliza valores aditivos, sendo assim também factos aditivos. A tabela *communications* considera-se um facto não aditivo, uma vez que não possui valores passíveis de serem somados, no entanto é possível efetuar contagens sobre a chave da tabela, agrupando os restantes campos.

Capítulo 3 – Matriz de Oferta-procura da Informação

3.1 – Narrativa do SI

O diretor deseja enriquecer a tomada de decisão da organização tornando acessível para os decisores dispor de dados históricos que lhes permitam basear as suas decisões. Foi para isso pedido uma solução aos SI, que lhes apresentaram a necessidade de possuir uma data warehouse, por existir dificuldades em responder de forma imediata às diversas questões que lhes tem sido colocado com a informação disponível nos sistemas de bases de dados operacionais.

As questões que fazem parte dos pedidos dos responsáveis são as relacionadas com a quantidade de filmes alugados, com uma periodicidade semanal, interessando também neste tema quais os filmes com melhor saída assim como as preferências dos clientes em relação aos atores e a categoria de filmes alugados. O diretor de recursos humanos habitualmente costuma pedir também uma relação dos funcionários e o seu vencimento, assunto que lhe interessa para planear as necessidades de admitir mais funcionários. Relativamente às relações com os fornecedores, têm sido recorrente pedido de informação de quais temos tido uma relação mais efetiva, quer em termos de aquisições, quer de comunicações efetuadas.

A empresa não dispensa a realização da festa de natal para os familiares dos funcionários como forma de fomentação do espirito coletivo. Há a necessidade de contemplar os filhos dos funcionários com idade até aos 14 anos de uma lembrança. Importa anualmente saber que tipo de prendas é necessário adquirir.

3.2- Matriz Oferta-Procura

A análise dos requisitos de uma data warehouse deve ter como objetivo analisar e compreender o contexto das necessidades da organização relacionando-os com os objetivos quer do negócio, como entidade única, quer dos diversos decisores, cada qual com as suas expectativas próprias, nomeadamente nas expectativas em relação à exploração da DW. É expectável também que estes requisitos sejam alterados ao longo do tempo, pelo que deverá estar sujeito a alterações devido ao aumento de requisitos.

Uma forma de se ter uma visão geral dos requisitos da data warehouse, é elaborar uma matriz, que registe de forma visual o que se oferece versus a procura de informação.

Luís Cavique (Cavique, Cavique, & Santos, 2020) apresenta um procedimento para se obter a matriz de oferta-procura, que s transcreve numa tradução livre para português:

Procedimento 1 – Geração da matriz oferta-procura

Input: ficheiros, bases de dado

Output: matriz de oferta-procura

1. Iterar
   1.1. Obter uma nova tabela de factos
   1.2. Fazer correspondência com dimensões
   1.3. Integrar com os requisitos
2. até que o equilíbrio entre a oferta e procura seja estabelecido

Aplicando este procedimento, obtemos uma matriz referente à oferta (Tabela 7: Matriz constelação das bases de dados Sakila, employees e suppliers) que resultará na proposta de constelação par a data warehouse, a que se associa a matriz de procura, resultando no conjunto matriz oferta procura (Tabela 8: Matrix oferta-procura) por aplicação do procedimento descrito.

O procedimento aplicado permite-nos obter uma visão geral do que a DW pode-nos oferecer.

A matriz de oferta-procura documenta a arquitetura da DW, dando uma perspetiva geral da oferta de factos que nos são disponibilizados e a sua relação com as dimensões. Isto permite-nos uma visão dos requisitos que podem ser extraídos da constelação resultante do procedimento aplicado, e conjuga-los com a procura de informação de interesse para a tomada de decisão. Assim, os requisitos, ou seja, a procura, devem ser suportados pelos dados disponíveis, nomeadamente pelas tabelas de factos, podendo ser suportados por uma ou mais tabelas de factos. A complexidade deve ser inicialmente reduzida, podendo, no entanto, ser aumentada durante o crescimento da própria DW.

Também é possível pela matrix de oferta-procura identificar factos cujo conteúdo é irrelevante no que concerne ao objetivo da DW, por os seus dados não apresentarem acréscimo qualitativo na tomada de decisão em termos estratégicos da empresa, logo não fazem sentido serem modelado; encaixam-se nesta observação as tabelas de factos *film_text*, *suppliers_address department_employee* e *department_manager*.

Tabela 7: Matriz constelação das bases de dados Sakila, employees e suppliers

Business Database	fact table		inventory	customer	staff	rental	actor	film	departments	suppliers		
Sakila	film_text	wf	X					X				
	payment	a		X	X	X		X				
	film_actor	wf					X	X				
	film_category	wf			X			X				
Employees	department_manager	wf			X				X			
	department_employee	wf			X				X			
	Salaries	a			X							
	relatives	wf			X							
Suppliers	suppliers_address	wf								X		
	aquisitions	a			X			X		X		
	communications	a			X					X		

Tabela 8: Matrix oferta-procura

fact table		inventory	customer	staff	rental	actor	film	departments	suppliers	Requeriments	1	2	3	4	5	6	7	8	9	10	11
film_text	wf	X					X														
payment	a		X	X	X		X			1. Filmes com melhor saída		X									
										2. Rel. Semanal de filmes alugados		X									
film_actor	wf					X	X			3. Preferência clientes por atores		X	X								
film_category	wf			X			X			4. Preferência clientes por categorias		X		X							
department_manager	wf			X				X													
department_employee	wf			X				X													
Salaries	a			X						5. Relação de funcionarios e vencimentos							X				
relatives	wf			X						6. Festa natal crianças idade até 14 anos								X			
suppliers_address	wf								X												
aquisitions	a			X			X		XX	7. Relação aquisições por funcionário									X		
communications	na			X					x	8. Relação de comunicações por funcionário											X

3.3 - Perguntas OLAP (8 perguntas)

A DW tem que oferecer respostas a métricas que são importantes para a tomada de decisão. Essas respostas têm que estar disponíveis para consulta dos decisores. Para este objetivo há a necessidade de manter relatórios padronizados que ofereçam essa resposta atualizada com os dados existentes.

De seguida mostra-se 5 exemplos de perguntas para as quais se desenvolveu em consultas do Microsoft Access 2016 e/ou no *Microsoft Power Pivot For Excel 2016* os relatórios e gráficos correspondentes.

As querys correspondentes estão descritas no manual de utilização em anexo a este relatório.

Pergunta 1: Filmes com melhor saída

Um top 10 de filmes alugados é um bom indicador. Poderá até ser utilizado para afixar de forma a ajudar utilizadores indecisos. Mostra-se o exemplo do mês 4

```
SELECT TOP 10
    FIL.title AS Titulo,
    Count(REN.rental_key) AS Alugueres
FROM 307_dim_calendar  AS CAL
    INNER JOIN (306_dim_rental AS REN
        INNER JOIN (323_fact_payment AS PAY
            INNER JOIN 303_dim_film AS FIL
            ON PAY.film_key = FIL.film_key)
        ON REN.rental_key = PAY.rental_key)
    ON CAL.calendar_key = PAY.payment_date_key
GROUP BY FIL.title, CAL.nMonth
HAVING (CAL.nMonth=4)
ORDER BY Count(REN.rental_key) DESC;
```

Filmes mais Alugados (top)	
Titulo	**Alugueres**
DINOSAUR SECRETARY	13
DORADO NOTTING	13
HOBBIT ALIEN	12
CARRIE BUNCH	12
SPY MILE	12
STEEL SANTA	12
KISS GLORY	12
ROCKETEER MOTHER	12

Filmes mais Alugados (top)	
Titulo	**Alugueres**
STORM HAPPINESS	12
MOCKINGBIRD HOLLYWOOD	12
ARIZONA BANG	12
DYNAMITE TARZAN	12
CAT CONEHEADS	12

Tabela 9 : Filmes mais alugados (Top mensal)

Pergunta 2: Relação semanal dos filmes alugados

A evolução diária de filmes alugados, e a sua relação semanal é uma forma de acompanhar a vitalidade do negócio, nomeadamente alguma oscilação que possa ocorrer.

```
TRANSFORM Count(REN.rental_key) AS Contar
SELECT
    CAL.nYearWeek AS Semana,
    Count(REN.rental_key) AS Contar1
FROM 307_dim_calendar AS CAL
INNER JOIN (306_dim_rental AS REN
    INNER JOIN 323_fact_payment AS PAY
    ON REN.rental_key = PAY.rental_key)
ON CAL.calendar_key = PAY.payment_date_key
GROUP BY CAL.nYearWeek
PIVOT CAL.nWeekday;
```

Dias da semana								
Semana	**Total**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
4	145						8	137
5	1011	174	166	196	154	158	163	
7	364						16	348
8	1947	324	325	344	348	331	275	
10	992					27	504	461
11	2461	512	513	480	461	495		
13	1302					33	649	620
14	3268	641	634	679	671	643		
16	1237					23	593	621
17	3135	628	624	659	626	598		
42	182				182			
Totais	16044	2279	2262	2358	2442	2308	2208	2187

Tabela 10 : Relação semanal de filmes alugados

Pergunta 3:

As preferências dos clientes têm uma relação com os atores intervenientes. É importante acompanhar essa preferência, pois pode ser um fator importante nos momentos de decisão na quantidade de cópias a adquirir de determinado filme, consoante o seu elenco. Pretende-se manter atualizado um relatório onde conste os atores cujos filmes são objeto de maior número de alugueres.

```
TRANSFORM Count(REN.rental_id) AS ContarDerental
SELECT
    ACT.first_name & ' ' & ACT.last_name AS Actor,
    Count(REN.rental_id) AS QtyAlugueres
FROM (((303_dim_film AS FIL
INNER JOIN (301_dim_actor AS ACT
INNER JOIN 322_fact_film_actor AS FAC
    ON ACT.actor_key = FAC.actor_key)
    ON FIL.film_key = FAC.film_key)
INNER JOIN 323_fact_payment AS PAY
    ON FIL.film_key = PAY.film_key)
INNER JOIN 306_dim_rental AS REN
    ON PAY.rental_key = REN.rental_key)
INNER JOIN 307_dim_calendar AS CAL
    ON REN.rental_date = CAL.calendar
GROUP BY
    ACT.first_name & ' ' & ACT.last_name
PIVOT CAL.nMonth;
```

Em power pivot, selecionou-se o top 5.

Top 5 atores com mais alugueres						
Alugueres	**Semanas**					
Atores	**1**	**2**	**3**	**4**	**10**	**Total**
SUSAN DAVIS	60	120	333	303	9	825
GINA DEGENERES	50	110	320	262	11	753
MATTHEW CARREY	46	93	293	232	14	678
MARY KEITEL	57	98	276	233	10	674
ANGELA WITHERSPOON	53	89	272	231	9	654
Total	**266**	**510**	**1494**	**1261**	**53**	**3584**

Tabela 11: Top 5 atores com mais alugueres (Power Pivot, Excel 2016)

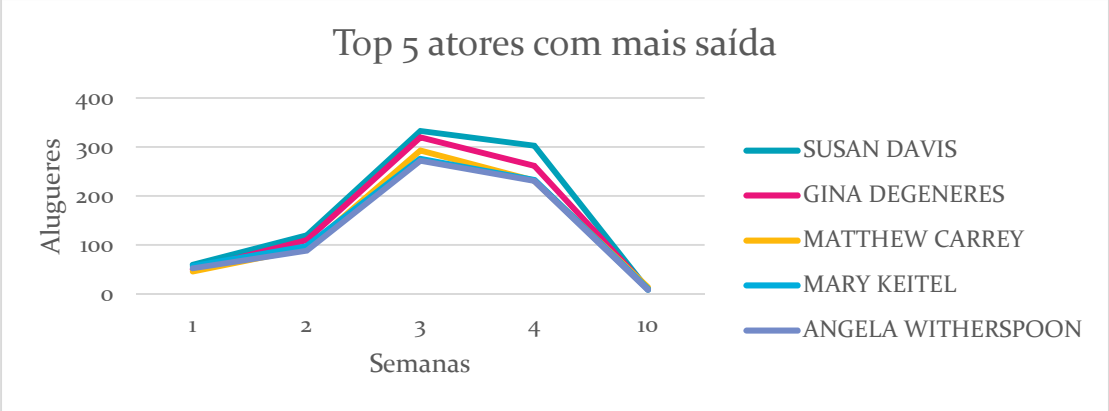

Figura 38: Top 5 atores com mais alugueres (Power Pivot – Excel 2016)

Pergunta 4:

Para análise do negócio, é importante existir uma visão geral da preferência dos utilizadores quanto às categorias de filmes preferidas, e a sua evolução temporal.

```
TRANSFORM Count(CAT.category_key) AS Soma
SELECT CAT.category_name
FROM ((326_fact_film_category AS FCA
INNER JOIN ((303_dim_film AS FIL
INNER JOIN 323_fact_payment AS PAY ON FIL.film_key = PAY.film_key)
INNER JOIN 306_dim_rental AS REN ON PAY.rental_key = REN.rental_key) ON
FCA.film_key = FIL.film_key)
INNER JOIN 308_dim_category AS CAT ON FCA.category_key =
CAT.category_key)
INNER JOIN 307_dim_calendar AS CAL ON REN.rental_date = CAL.calendar
GROUP BY CAT.category_name
PIVOT CAL.nYearWeek;
```

Q1 - Evolução semanal de alugueres por categoria											
category_name	**4**	**5**	**7**	**8**	**10**	**11**	**13**	**14**	**16**	**17**	**42**
Action	10	77	28	132	79	165	74	238	81	211	17
Animation	14	60	34	140	80	172	100	235	79	231	21
Children	12	59	15	115	52	154	75	194	72	191	6
Classics	9	53	18	118	48	157	63	207	75	182	9
Comedy	8	64	11	124	48	148	73	196	84	176	9
Documentary	10	76	21	139	70	150	83	209	96	190	6
Drama	8	77	22	130	65	163	91	207	81	209	7
Family	8	77	17	137	65	173	90	219	76	221	13
Foreign	10	61	22	120	58	165	91	208	75	212	11
Games	6	63	25	117	65	141	84	192	80	182	14
Horror	5	48	20	96	63	122	72	174	57	177	12

Q1 - Evolução semanal de alugueres por categoria											
category_name	4	5	7	8	10	11	13	14	16	17	42
Music	9	52	26	107	50	125	75	156	66	153	11
New	2	58	23	109	56	144	79	195	63	198	13
Sci-Fi	16	68	29	133	62	173	95	216	78	223	8
Sports	6	70	25	134	80	177	95	249	100	228	15
Travel	12	48	28	96	51	132	62	173	74	151	10

Tabela 12: Evolução semanal de alugueres por categoria (Access 2016)

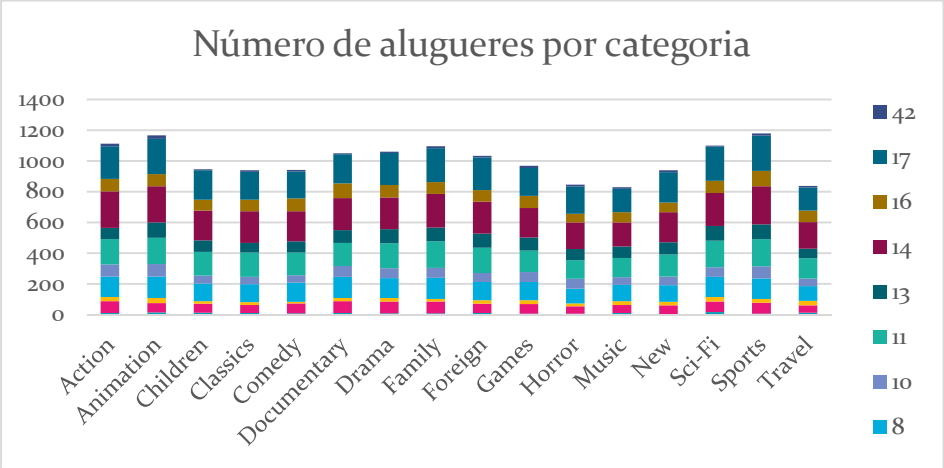

Figura 39 : Acumulado Evolução semanal de alugueres por categoria (Power Pivot, Excel 2016)

Pergunta 5: Relação dos funcionários e vencimentos.

```
SELECT
    STA.first_name & ' ' & STA.last_name AS Nome,
    SAL.title AS Categoria,
    SAL.salary AS Vencimento
FROM 304_dim_staff AS STA
    INNER JOIN 325_fact_salaries AS SAL
    ON STA.staff_key = SAL.staff_key;
```

Funcionários		
Nome	**Categoria**	**Vencimento**
Albert Payne	Counter 2nd	1 000,00 €
Alex Landry	Counter 2nd	1 000,00 €
Ally Williams	Counter newbie	820,00 €
Amelie Vargas	Counter newbie	800,00 €
Anna Moss	Counter 2nd	1 000,00 €
Cory Cob	Counter 2nd	1 000,00 €
Elliott McConnell	Administrator	2 500,00 €
Hazel Haney	Manager	1 550,00 €
Jon Stephens	Counter manager	1 200,00 €
Leila Bautista	Manager	1 500,00 €
Mike Hillyer	Counter manager	1 200,00 €

Pergunta 6: Relação de prendas a distribuir pelos filhos dos funcionários na festa de natal.

```
TRANSFORM Count(REL.relative_key) AS numero
SELECT
    REL.child_gender
FROM 324_fact_relatives AS REL
    INNER JOIN 307_dim_calendar AS CAL
    ON REL.child_birth_date_key = CAL.calendar_key
WHERE
    (((Year((Date()-[calendar]))-1900)<=14))
GROUP BY
    REL.child_gender
PIVOT Year((Date()-[calendar]))-1900;
```

Idades						
Género	0	1	2	7	10	14
F			1		1	1
M	1	1		1		

Pergunta 7:

Interessa saber o nível de atualização do catalogo de filmes disponíveis. Para este índice convém manter um relatório atualizado das aquisições que se tem efetuado em cada semana, e quem tem sido os funcionários mais dinâmicos nesta atividade.

```
TRANSFORM Count(AQU.aquisition_id) AS Aquisições
SELECT
    CAL.nYearWeek AS Semana,
    Count(AQU.aquisition_id) AS [Total Aquisições]
FROM ((320_fact_aquisition AS AQU
INNER JOIN 304_dim_staff AS STA
    ON AQU.staff_key = STA.staff_key)
INNER JOIN 303_dim_film AS FIL
    ON AQU.film_key = FIL.film_key)
INNER JOIN 307_dim_calendar AS CAL
    ON AQU.aquisition_date_key = CAL.calendar_key
GROUP BY
    CAL.nYearWeek
PIVOT STA.first_name & ' ' & STA.last_name;
```

Aquisições semanais por funcionário				
Semana	Total Aquisições	Albert Payne	Alex Landry	Amelie Vargas
4	235	3	118	114
5	903	14	447	442
7	81	2	40	39
8	234	6	113	115
10	30	1	14	15
11	33	2	15	16
13	3		1	2
14	2		1	1

Tabela 13: Aquisições semanais por funcionário (Access 2016)

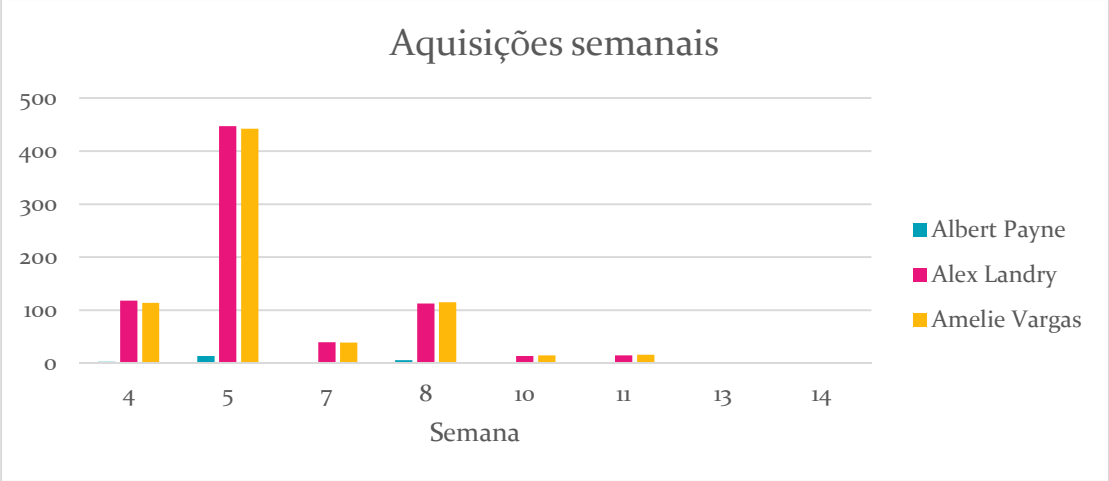

Figura 40 : Aquisições semanais por funcionário (Power Pivot, Excel 2016)

Pergunta 8:

Ainda relacionado com as ações desenvolvidas pelos funcionários, deseja-se manter um relatório das interações efetuadas com os fornecedores.

```
TRANSFORM
    Count(COM.comm_key) AS ContarDecomm_key
SELECT
    STA.first_name & ' ' & STA.last_name AS Nome,
    SUP.supplier_name, Count(COM.comm_key) AS Total
FROM ((321_fact_communication AS COM
INNER JOIN 304_dim_staff AS STA
    ON COM.staff_key = STA.staff_key)
INNER JOIN 305_dim_suppliers AS SUP
    ON COM.supplier_key = SUP.supplier_key)
INNER JOIN 307_dim_calendar AS CAL
ON COM.comm_date_key = CAL.calendar_key
GROUP
    BY STA.first_name & ' ' & STA.last_name,
    SUP.supplier_name
PIVOT CAL.nMonth;
```

Contactos com fornecedores			
	Mês		
Funcionários	**1**	**2**	**Total**
Albert Payne	**15**	**8**	**23**
Castelo Lopes Movies		1	1

Disney Movies	1		1
DreamWorks Movies	5	3	8
Lusomundo Movies	2	1	3
Marvel Comics	1	1	2
Rocky movies	4	1	5
Sony Movies	2	1	3
Alex Landry	**3**	**2**	**5**
Castelo Lopes Movies		2	2
Marvel Comics	3		3
Amelie Vargas	**9**	**4**	**13**
DreamWorks Movies	3	1	4
Lusomundo Movies	2	1	3
Marvel Comics		1	1
Rocky movies	2		2
Sony Movies	2	1	3
Anna Moss	**2**	**2**	**4**
Castelo Lopes Movies		1	1
Lusomundo Movies		1	1
Sony Movies	2		2
Cory Cob	**2**	**3**	**5**
DreamWorks Movies	1	1	2
Marvel Comics	1		1
Sony Movies		2	2
Total Geral	**31**	**19**	**50**

Tabela 14: : Contactos a fornecedores por funcionário (Power Pivot, Excel 2016)

Capítulo 4 - Conclusão

Implementação de uma data warehouse tem sido um tema amplamente debatido e estudado desde que Ralph Kimball e Bill Inmon, considerados os pioneiros, desenvolveram bases sólidas através de duas interpretações distintas de implementação, mas com o objetivo único de construir bases de dados orientadas e otimizadas para a exploração de dados. Hoje existe uma vasta gama de estudos e processos de implementação, bem como publicações que de forma mais ou menos didática abordam o assunto. Destaca-se neste campo os artigos de Luís Cavique et al. Mencionados que abordam questões práticas, em linguagem acessível e compreensível, o que vem contrastar com artigos mais técnicos e de linguagem mais difícil de compreensão para quem não está muito familiarizado com a temática.

A segunda parte reflete a parte prática do estudado, aplicando os métodos teóricos e traduz-se pela elaboração da matriz constelação, e a correspondente matriz de oferta e procura; e a implementação pratica da DW através do processo de ETL, finalizando com a elaboração de relatórios com vista à resposta de questões a que a DW deva ser capaz de fornecer métricas para suporte à decisão. Para o efeito foram extraídas métricas interessantes disponibilizados através de relatórios.

A motivação para a realização deste trabalho está relacionada pelo desempenho de funções atuais e talvez futuras uma vez que o tema possui um interesse grande no contexto da missão de entidade patronal.

De referir ainda que não existiram dificuldades acrescidas na realização do projeto que no entanto exigiu bastante consulta bibliográfica sobre o tema.

Anexos

Manual de utilização da implementação da Data Warehouse

Bibliografia

Adamson, C. (2010). *Star Schema, the complete reference.* McGraw-Hill.

Cavique, L., Cavique, M., & Gonçalves, A. (2019). Extraction of Fact Tables from a relacional Database: An Effort to Establish Rules in Denormalization. *7th World Conference on Information Systems and Technologies.* La Toja Island, Galiza, Spain.

Cavique, L., Cavique, M., & Santos, J. M. (2020). Supply-demand matrix: a process-oriented approach for data warehouses with constellation schemas. *8th World Conference on Information Systems and Technologies* .

Group, K. (s.d.). *https://www.kimballgroup.com/*. Obtido de Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/additive-semi-additive-non-additive-fact/

Inmon, W. H. (2002). *Building the Data Warehouse, Third Edition.* John Wiley & Sons, Inc.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit second edition.* New York: John Wiley & Sons, Inc.

Kimball, Ralph; Caserta, Joe. (2004). *The Data Warehouse ETL Toolkit.* Indianapolis: Wiley Publishing, Inc.

Oracle. (1 de Setembro de 2020). *Sakila Sample Database*. Obtido de mysql: https://dev.mysql.com/doc/sakila/en/

Ponniah, P. (2001). *Data Warehousing Fundamentals: A comprehensive Guide for IT professionals.* John Wiley & Sons, Inc.

Silvers, F. (2007). *Building and Maintaning a Data Warehouse.* Taylor & Francis Group, LLC.