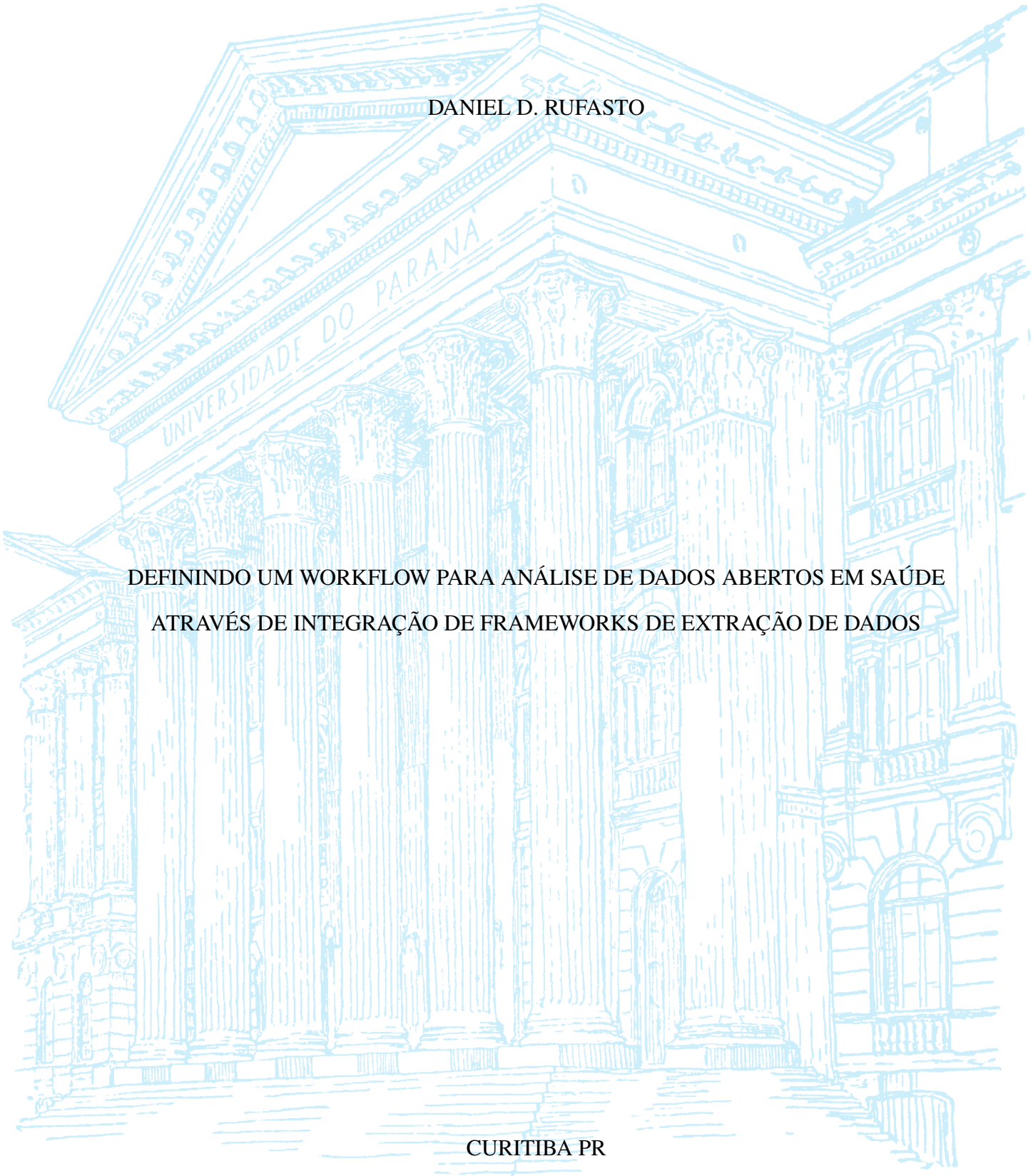
UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL D. RUFASTO



DEFININDO UM WORKFLOW PARA ANÁLISE DE DADOS ABERTOS EM SAÚDE ATRAVÉS DE INTEGRAÇÃO DE FRAMEWORKS DE EXTRAÇÃO DE DADOS

CURITIBA PR

2019

DANIEL D. RUFASTO

DEFININDO UM WORKFLOW PARA ANÁLISE DE DADOS ABERTOS EM SAÚDE ATRAVÉS DE INTEGRAÇÃO DE FRAMEWORKS DE EXTRAÇÃO DE DADOS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Informática Biomédica*.

Orientador: Marcos Didonet del Fabro.

CURITIBA PR

2019

Mi nur estas lernanto

AGRADECIMENTOS

Ao meu orientador Marcos Didonet del Fabro que aceitou orientar um aluno reprovado 3 vezes na disciplina de Trabalho de Graduação em Informática Biomédica, um professor exemplo ao ensinar.

A minha madrasta Ana Maria Alvarez Barros e minha mãe Dinara Souza Dutra, mulheres excepcionais que me nutriram com amor e me providenciaram um lar.

A minha orientadora de estágio Dora Yoko Nozaki Goto que me ensinou tudo o que eu sei sobre SUS

Ao Stack Overflow que democratiza conhecimento e tira dúvidas de alunos.

RESUMO

Dados abertos de saúde pública do paraná do SIM - Sistema de Informação de Mortalidade e SINASC - Sistema de Informação de Nascidos Vivos foram submetidos a um workflow de dados, desde sua extração, transformação, carregamento e agrupamento em um sistema gerenciador de banco de dados MonetDB através da ferramenta HOTMapper, quanto a realização de consultas OLAP através da ferramenta BlenDB, com o objetivo de facilitar o mapeamento e eventuais indicadores de saúde que necessitem de consultas OLAP.

Palavras-chave: Dados abertos em saúde. Workflow de dados. Ferramentas de Epidemiologia.

ABSTRACT

Open healthcare data of the state of Paraná from SIM - Sistema de Informação de Mortalidade (Death Surveillance System) and SINASC - Sistema de Informação de Nascidos Vivos (Born Alive Surveillance System) were submitted ta a data workflow, since its extraction, transformation, loading and grouping in a MonetDB Database Manager System via HOTMapper tool, until the capability of OLAP queries via BlenDB tool, aiming for easier mapping and generation of eventual health indicator metrics that may need OLAP queries.

Keywords: Open Health Data. Data workflow. Epidemiology tools.

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS

UFPR	Universidade Federal do Paraná
SUS	Sistema Único de Saúde
DO	Declaração de Óbito
DN	Declaração de Nascido Vivo
DOPR	Declarações de Óbito do Estado do Paraná
OLAP	Online Analytical Process
SGBD	Sistema Gerenciador de Banco de Dados
SIM	Sistema de Informação de Mortalidade
SINASC	Sistema de Informação de Nascidos Vivos
ETL	Extract Transfrom and Load

SUMÁRIO

1 INTRODUÇÃO

O SUS possui uma grande quantidade de dados de saúde pública. A análise de tais dados, feita principalmente pelos setores públicos de epidemiologia, serve de norteamento e justificativa para a tomada de decisão em saúde pública por parte dos gestores (por exemplo, a compra de vacinas e ações de promoção a saúde contra a dengue se justificam pela grande quantidade de casos em 2006).

Dados de mortalidade e nascidos vivos são base para muitos estudos e também são alicerce para o levantamento de muitos indicadores de saúde já estabelecidos e aceitos pela comunidade como métricas fundamentais para a análise de situação de saúde em determinado local.

Os dados de saúde pública, entretanto, são densos, e existe o potencial para várias análises além do que já é feito atualmente, porém, os dados estão em um formato pouco convencional, separados em arquivos anuais e seus campos estão codificados.

Neste trabalho pretende-se desenvolver um workflow de dados de saúde pública para agrupá-los em uma única base, com seus campos tratados e para facilitar a realização de consultas analíticas.

No capítulo 2 são apresentados conceitos relacionados à bancos de dados relacionais, integração de dados, sistemas OLAP e data warehouses e dados abertos. No capítulo 3 o problema de integrar dados abertos é discutido, sendo abordados dois trabalhos relacionados. Nos capítulos 4 e 5 o trabalho em detalhes é apresentado junto com os resultados obtidos e validações. Por fim o capítulo 6 contém a conclusão e discussão de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 BANCO DE DADOS RELACIONAIS

Um sistema gerenciador de banco de dados (SGBD) é um sistema computadorizado projetado para permitir que os usuários possam inserir, modificar, buscar e remover dados no computador (Date, 2000). Um banco de dados relacional é um banco de dados que implementa os conceitos de entidade e relacionamento.

2.1.1 Modelo Entidade Relacionamento

Date (2000) explica o conceito de entidade e relacionamento através da ilustração de uma fábrica, representada pelo diagrama na figura 2.1.

A "Faz Tudo Ltda."é uma empresa que normalmente deseja registrar informações sobre os projetos em andamento; as peças usadas nos projetos; os fornecedores que estão contratados para fornecer essas peças; os empregados que trabalham nesse projeto e assim por diante (Date, 2000).

Nesta ilustração, os itens citados, projetos, empregados etc. são considerados entidades. Em resumo, uma entidade é qualquer objeto que deva ser representado pelo banco de dados. Na ilustração também pode-se ver diversos interligamentos entre as entidades: eles são chamados de relacionamentos. Estes relacionamentos conectam as entidades, como o exemplo "FP"(ou "remessas") elucida as perguntas "Dado um fornecedor, obtenha as peças fornecidas para esse fornecedor", "Dada uma peça, obtenha os fornecedores que fornecem essa peça".

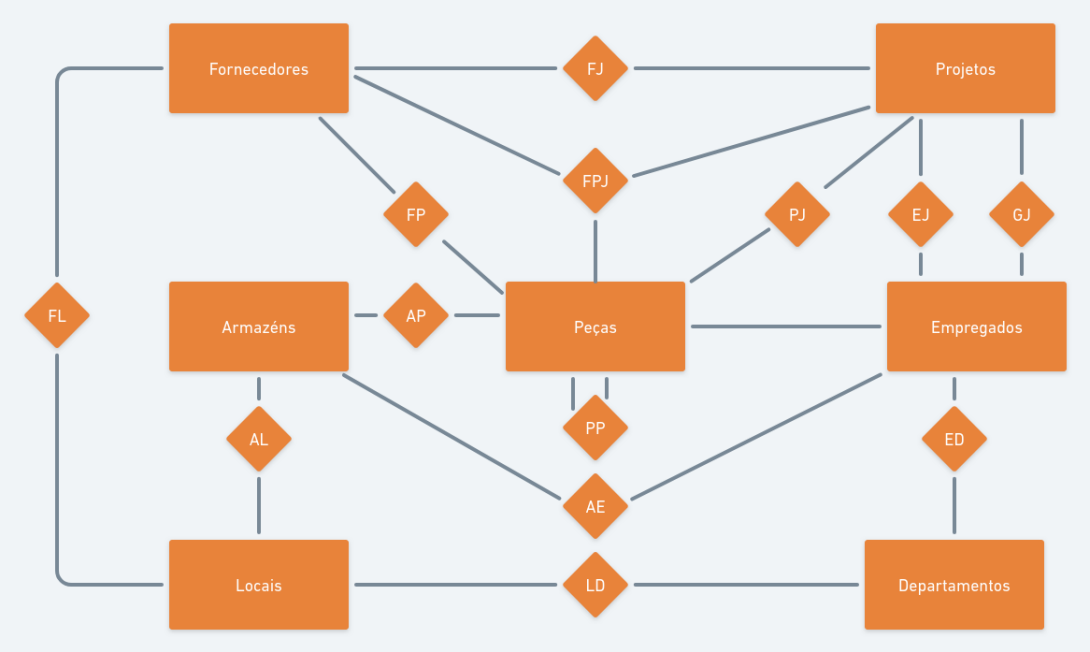


Figura 2.1: Ilustração do Modelo Entidade-Relacionamento

Tanto as entidades como os relacionamentos, podem ser representados no banco de dados relacional.

2.1.2 Modelo Relacional

Date (2000) descreve o modelo relacional através de três aspectos: aspecto estrutural, aspecto de integridade e aspecto manipulador (Date, 2000). O aspecto estrutural (figura 2.2) define que o usuário perceba os dados de um banco de dados como tabelas e apenas tabelas. O aspecto de integridade define que essas tabelas obedeçam um conjunto de regras de integridade. O aspecto manipulador (figura 2.3) define a maneira que o usuário interage com os dados; por meio de operadores de dados, por exemplo, ou operadores de restrição, projeção e junção.

DEPTO

DEPTO#	NOMEDEPTO	ORÇAMENTO
D1	Marketing	10M
D2	Desenvolvimento	12M
D3	Pesquisa	5M

EMP

EMP	NOMEEMP	DEPTO#	SALÁRIO
E1	Lopez	D1	40K
E2	Cheng	D2	42K
E3	Finzi	D3	30K
E4	Saito	D4	35K

Figura 2.2: Ilustração do Aspecto Estrutural, os dados são percebidos como tabelas pelo usuário

Restrição:
DEPTO's onde ORÇAMENTO > 8m

Resultado:

DEPTO#	NOMEDEPTO	ORÇAMENTO
D1	Marketing	10M
D2	Desenvolvimento	12M

Projeção:
DEPTO's sobre DEPTO#, ORÇAMENTO

Resultado:

DEPTO#	ORÇAMENTO
D1	10M
D2	12M
D3	5M

Junção:
DEPTO's e EMP's sobre DEPTO#

Resultado:

DEPTO#	NOMEDEPTO	ORÇAMENTO	EMP	NOMEEMP	SALÁRIO
D1	Marketing	10M	E1	Lopez	40K
D2	Desenvolviment	12M	E2	Cheng	42K
D3	Pesquisa	5M	E3	Finzi	30K

Figura 2.3: Ilustração de operadores de manipulação, Restrição, Projeção e Junção

Abraham Silberschatz define uma base de dados relacional como uma coleção de tabelas, cada uma com um nome único. Um exemplo é a figura 2.4, que guarda a informação sobre instrutores. A tabela é composta de quatro títulos de colunas: ID, nome, departamento e salário. Cada linha desta tabela guarda a informação de um instrutor constituído do ID do mesmo, o nome, o departamento onde trabalha e seu salário.

De modo geral, uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Já que uma tabela é considerada uma coleção de várias linhas, existe uma correlação próxima entre o conceito de tabela e o conceito matemático de relação, o que dá nome ao modelo relacional de dados. Em termos matemáticos, uma tupla é simplesmente uma sequência de valores, um relacionamento entre n valores é representado matematicamente por uma tupla com o mesmo numero n de valores, (também chamada de n-tupla), que corresponde a uma linha em uma tabela (Silberschatz, 2011).

2.2 INTEGRAÇÃO DE DADOS

Para a explicação de integração de dados, é necessário primeiro contextualizar um sistema multi base de dados. Em sistema multi base de dados, já existem diversas bases de

ID	Nome	Depto	Salario
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec Eng	80000

Figura 2.4: Tabela de instrutores

dados e o objetivo é fazer o design de todos em uma única base. Uma abordagem consiste em simplesmente integrar as várias bases com os *schemas* locais em uma base global com um *schema* conceitual global (e Valduriez, 2011).

A integração pode ser feita de maneira física ou lógica. Quando feita de maneira física, as bases de dados são integradas e a base integrada resultante é *materializada*, resultando no que é conhecido como *data warehouse*. A integração é auxiliada por ferramentas ETL (explicado posteriormente) que permitem a extração multi base, a transformação para que os dados finais correspondam ao *schema* final designado e por fim o carregamento (materialização) desses dados. A abordagem lógica não será abordada por fugir do escopo deste trabalho (e Valduriez, 2011). A figura 2.5 mostra uma ilustração do processo físico de integração de dados que funciona da seguinte maneira: para cada base, seus respectivos dados serão extraídos pela ferramenta, transformados e armazenados da base integrada. Esse processo pode ser independente e diferente dependendo das especificações de cada base de dados.

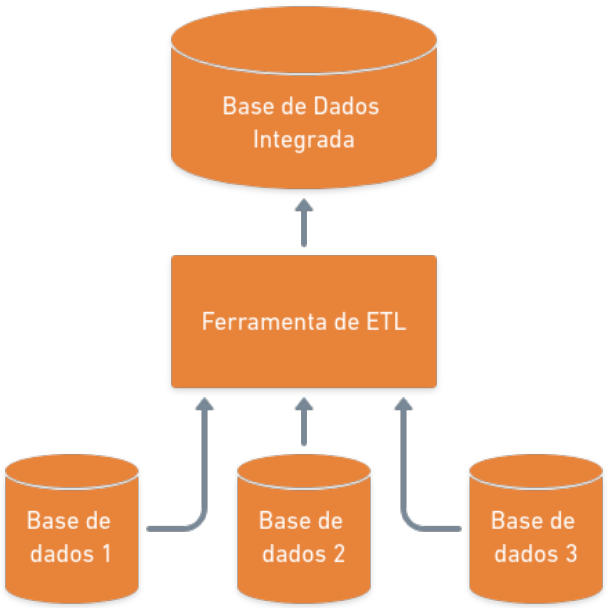


Figura 2.5: Esquematização da Integração de dados (e Valduriez, 2011)

Ferramentas ETL (do inglês *Extract, Transform, Load*), operam em três níveis: Extração, Transformação e Carregamento.

A Extração consiste em saber a origem das diferentes fontes de dados que serão ingeridos pelo processo ETL e saber como acessá-las. É comum o uso de mapeamentos para cada origem diferente, tais mapeamentos contem as informações de cada campo da(s) base(s) de origem e a base destino do seu ETL. Também é necessário atentar ao fato de que diferentes bases origem de dados podem ter formatos diferentes.(Kimball, 2004)

A Transformação consiste no processo de aplicar em uma série de operações sobre os dados previamente extraídos, visando dentre vários propósitos, a melhor legibilidade e padronização dos dados. Também é necessario atentar ao fato de que campos diferentes podem necessitar de mais operações ou menos operações para que se alcance um resultado satisfatório.(Kimball, 2004) Os seguintes exemplos de transformações são provenientes do SIM - Sistema de Informação de Mortalidade que é mantido pelo DATASUS:

- Transformar o campo SEXO de 1 para "M"; 2 para "F"e 9 para "I"
- Transformar o campo C.I.D. de X70. para "Lesão autoprovocada intencionalmente por enforcamento, estrangulamento e sufocação"
- Construir o campo "Quantidade de óbitos"através de uma contagem aplicada no banco de dados.
- Separação de uma coluna em diversas outras (por exemplo utilizando o caractere ";"como separador)

O Carregamento consiste no processo de inserção dos dados previamente transformados no Data Warehouse. É necessário atentar ao fato de que diferentes Warehouses possuirão diferentes maneiras de inserção, por exemplo, alguns necessitarão que a inserção ocorra em determinados intervalos de tempo, outros terão regras específicas de inserção.(Kimball, 2004)

2.3 PROCESSAMENTO ANALÍTICO ONLINE (OLAP) E DATA WAREHOUSES

O termo OLAP vem do inglês *Online Analytical Processing*, é entendido como um conceito que envolve a criação, o gerenciamento, a análise e a geração de relatórios de dados.(Date, 2000) O termo Data Warehouse é entendido como uma coleção de dados que contém algumas particularidades: a coleção é voltada a um ou mais propósitos específicos, é integrada, não volátil, variável no tempo para o suporte à tomada de decisão. Os data warehouses servem para prover acesso a dados para diversas análises de dados, que por sua vez auxiliam organizações na tomada de decisão. (Navathe, 2011) A figura 2.6 mostra o funcionamento de um data warehouse em um projeto.

O melhor modelo de dados para os data warehouses e OLAP's é o modelo de armazenamento de dados multidimensionais. Este modelo consiste em dois conceitos expressos em tipos de tabelas, o primeiro é o conceito de dimensão, uma tabela de dimensão é composta de tuplas de atributos da dimensão. O segundo conceito é o conceito de fatos, uma tabela de fatos pode ser explicada como uma série de tuplas, uma para cada fato registrado, cada fato contém uma ou mais variáveis observadas e a(s) atribui(em) a tabelas de dimensão através de ponteiros. Em resumo, a tabela de fatos possui os dados e a tabela de dimensões é responsável por identificar cada tupla desses dados. (Navathe, 2011) Uma ilustração desse modelo pode ser vista na figura 2.7

Por se tratar de relatórios, invariavelmente existirão agregações de dados. A questão é que cada agregação exige uma consulta diferente ao sistema gerenciador de banco de dados.

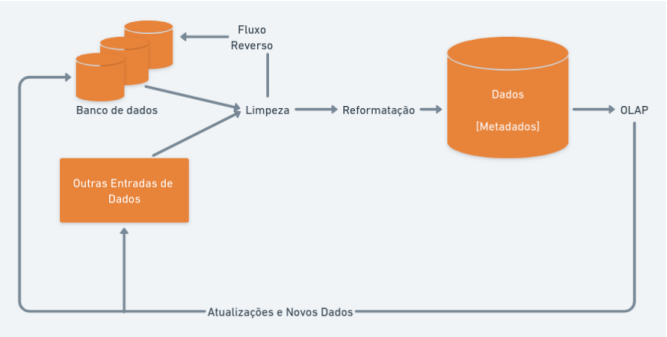


Figura 2.6: Esquematização de data warehouse (Navathe, 2011)

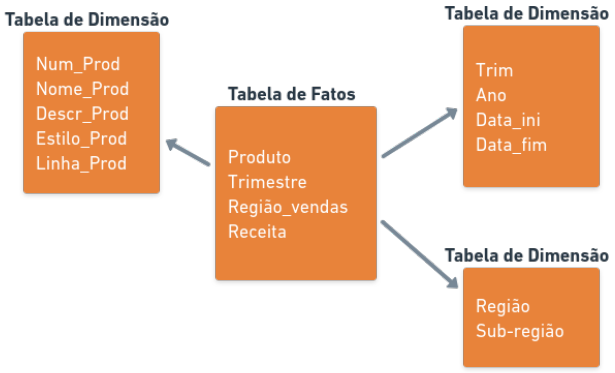


Figura 2.7: Modelo multidimensional (Navathe, 2011)

Existem desvantagens ao utilizar essa abordagem: é tedioso para o usuário ter que formular muitas consultas semelhantes mas diferentes, muitas vezes com diferenças minuciosas, a execução dessas consultas também é dispendiosa em tempo de execução (em particular, pesquisando repetidamente os mesmos dados), em suma, produzir n agregações exige que n consultas diferentes sejam executadas no banco de dados(Date, 2000). Para exemplificar, suponha um banco de dados com apenas 2 peças (P1 e P2) com apenas 4 fornecedores (F1, F2, F3 e F4) e a tabela de remessas seja a tabela da figura 2.7. A consulta representada pela figura 2.8 mostra a contagem da quantidade total de remessas, sem qualquer distinção de fornecedores ou peças. A consulta representada pela figura 2.9 mostra a quantidade total de remessas agrupadas pela dimensão de fornecedores.

F#	P#	Qtde
F1	P1	300
F1	P2	200
F2	P1	300
F2	P2	400
F3	P2	200
F4	P2	200

Figura 2.8: Tabela de remessas (Date, 2000)

Consideram-se as seguintes consultas:

- Quantidade total de remessas.

- Quantidades totais de remessas por fornecedor.

  As ilustrações das consultas de um modelo OLAP são:

```
SELECT SUM(QTDE) AS QTDETOTAL
FROM   FP
GROUP  BY ( );
```

QdeTotal
1600

Figura 2.9: Primeira Consulta (Date, 2000)

```
SELECT   F#
         SUM(QTDE) AS QTDETOTAL
FROM   FP
GROUP  BY (F# );
```

F#	QtdeTotal
F1	500
F2	700
F3	200
F4	200

Figura 2.10: Segunda Consulta
(Date, 2000)

2.4 DADOS ABERTOS

Dados abertos são dados que seguem os seguintes critérios:

- **Acessibilidade**: deve ser possível adquirir os dados pagando apenas o custo de reprodução, digital, sem restrições quanto ao usuário que solicita os dados.
- **Digitalizado**: os dados devem estar num formato interpretável por máquina, por fins de interoperabilidade.
- **Livre**: os dados não devem ser livres para uso e redistribuição em seu termo de licenciamento.

(Parnia, 2000)

2.4.1 Dados Abertos Governamentais

O conceito de dados abertos governamentais segue o conceito de dados abertos, porém com a ressalva de que os dados são provenientes de setores públicos governamentais. Nesse caso, o governo serve de fonte de dados, seja através de sistemas de informação governamentais, iniciativas sociais, políticas públicas de informação etc. (Parnia, 2000)

2.4.2 DataSus

O DataSus é o órgão dentro do SUS responsável pelo armazenamento e distribuição de dados sobre saúde pública no Brasil. O DataSus é responsável pela manutenção de vários sistemas de informação em saúde brasileiros, dentre eles, o SIM - Sistema de Informação de Mortalidade e o SINASC - Sistema de Informação de Nascidos Vivos são dois sistemas cujos dados abertos são utilizados neste trabalho. A figura 2.11 esquematiza o fluxo de dados do sistema SIM, estes dados são coletados através do preenchimento manual de papéis de fichas de ocorrências em diversas pontos físicos do SUS (sejam eles unidades básicas de saúde, hospitais ou maternidades, em suma locais do SUS onde ocorra um óbito ou um nascimento), que são digitalizadas pelas respectivas secretarias municipais e enviadas através de lotes para o Ministério da Saúde, para após este processo serem distribuídas para as secretarias de saúde e abertas para a população por meio dos sistemas e site do DataSus. O sistema SINASC funciona de maneira análoga, porém ao invés dos dados serem fichas de declaração de óbito, eles são declarações de nascidos vivos. (da Saúde Brasileiro, 2019)

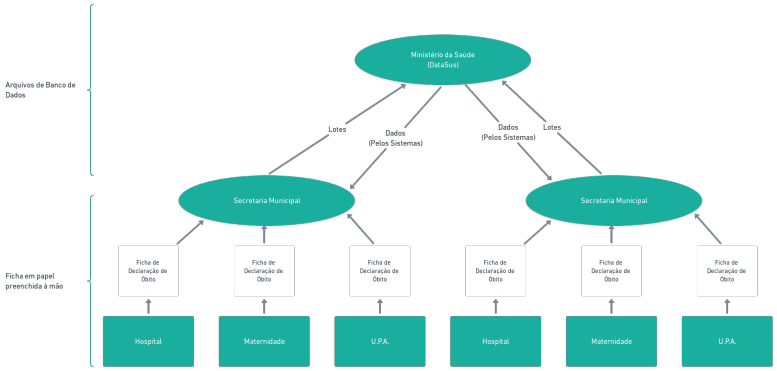


Figura 2.11: Fluxograma de extração de dados do SIM (da Saúde Brasileiro, 2019)

3 TRABALHOS RELACIONADOS

3.1 BIOD - BLENDED INTEGRATED OPEN DATA

3.1.1 Introdução

O trabalho realizado no BIOD é um exemplo de workflow de dados abertos, porém, ao invés do escopo de dados de saúde pública, os dados utilizados são dados de educação pública (Ehrenfried et al., 2019a). O objetivo final do projeto é a utilização da base de dados para consultas do tipo OLAP (analíticas) através de uma API (Application Programming Interface) RESTful (Representional State Transfer). É necessário ter em mente que os dados públicos estão disponíveis na web, porém não num formato que facilite consultas analíticas, levando em consideração que a maioria dos dados públicos são fontes de indicadores para tomada de decisão em gestão pública, estes indicadores por sua vez, são obtidos através de agregações, essa demanda justifica a arquitetura estar preparada e otimizada para consultas OLAP.(Ehrenfried et al., 2019a) A figura 3.1 mostra as tabelas integradas do BIOD.

	Tabelas	Colunas	Registros		Tabelas	Colunas	Registros
LDE Ensino Superior	aluno_ens_superior	128	81.813.362	LDE Geral	cidade	6	5570
	curso_ens_superior	132	262.691		estado	4	27
	docente_ens_superior	52	3.095.302		familias_cadunico	32	30.009.293
	fies	55	85.268.278		ibge_pib	8	27.835
	ies_ens_superior	51	19.137		pessoas_cadunico	36	90.419.338
	localoferta_ens_superior	17	1.396.990		pnad	56	6.015.874
	ocde_ens_superior	9	1.162		regiao	2	5
	prouni	15	1.069.600				
LDE Ensino Básico	aluno	106	272.597.280	SIMMC	fnu	9	2.046.110.236
	formação_superior	6	255		localizacao_ponto	3	6732
	escola	185	1.375.322		ponto	8	18424
	professor	146	58.190.497				
	turma	95	12.494.116				
	instituição_superior	8	6002				

Figura 3.1: Tabelas Integradas do BIOD (Ehrenfried et al., 2019a)

3.1.2 Armazenamento

O armazenamento dos dados do BIOD foi feito pelo SGBD MonetDB(mon, 2019). O MonetDB é um banco de dados relacional com armazenamento colunar. O motivo também são as consultas do tipo OLAP. O armazenamento colunar facilita agregações.

3.1.3 Disponibilização

A disponibilização dos dados é feita pelo BlenDB.(ble, 2019) O BlenDB é a ferramenta responsável por permitir que o banco de dados do MonetDB seja acessado pela API RESTFUL, utilizando uma linguagem de consulta simplificada. As vantagens desse método são que as consultas podem ser realizadas no próprio repositório, não precisando fazer uma cópia local, fazer a consulta no repositório também permite que eles sejam pré-agregados, ou seja, os dados perdem precisão mas ficam menores e mais fáceis de serem transportados.

3.2 HOTMAPPER

3.2.1 Introdução

HOTMapper é o acrônimo para *Historical Open Data Table Mapper* (Ehrenfried et al., 2019b). O Hotmapper é uma proposta para solucionar o problema de dados abertos que são disponibilizados em um certo período de tempo. O problema em questão é que os dados abertos podem estar em formatos diferentes em anos diferentes, portanto, é papel do Hotmapper providenciar o mapeamento de cada base de dados de *schemas* e formatos diferentes para uma única base de dados. Sua demonstração foi realizada utilizando dados públicos de educação.

3.2.2 Funcionamento e Arquitetura

A figura 3.2 ilustra a arquitetura do Hotmapper

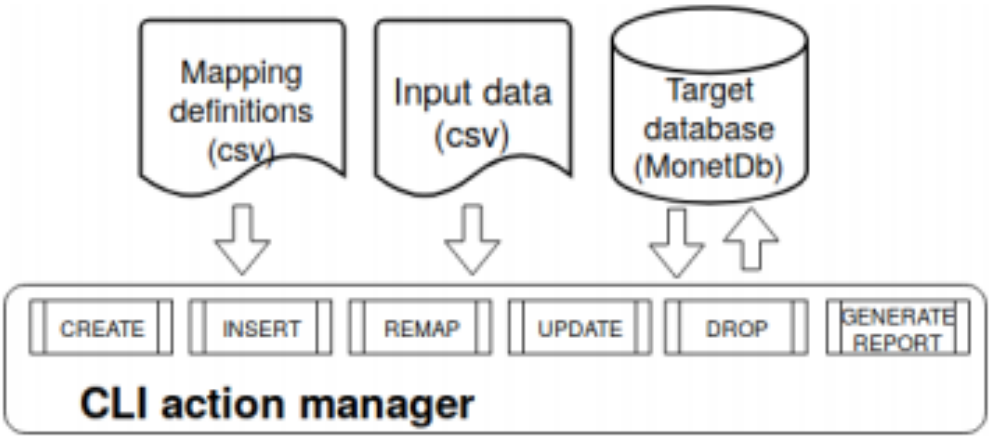


Figura 3.2: Arquitetura do Hotmapper (Ehrenfried et al., 2019b)

Os arquivos de **definição de mapeamento** do hotmapper são arquivos *.csv* um por cada base de dados a ser importada pelo hotmapper. Eles contém a informação de como cada coluna deve ser mapeada, ou seja, como o banco integrado irá registrar a informação de cada coluna, para cada ano especificado. Os arquivos de **input** são os dados abertos que serão importados pela ferramenta. Eles obrigatoriamente devem estar no formato *.csv*, que é o formato padrão da maioria dos arquivos públicos de educação utilizados para demonstração do Hotmapper. A **base de dados** alvo é(são) a(s) tabela(s) no MonetDB que registrarão a informação armazenada. O objetivo final do hotmapper é que o usuário ao precisar agrupar várias versões de uma mesma base de dados, tenha apenas uma vez, o trabalho de fazer o mapeamento, para em seguida acionar o hotmapper e ter uma tabela final agrupando todos os dados de todos os anos, facilitando futuras análises de dados e consultas OLAP.

3.2.3 Descrição das Tarefas

O Hotmapper é utilizado por meio de uma CLI *command line interface*, assim como ilustrado na figura 3.2, os comandos possíveis são:

- CREATE: utiliza um arquivo de definição de mapeamento como parâmetro, cria uma tabela e executa os comandos de manipulação de dados do arquivo.
- DROP: deleta a tabela passada como parâmetro e quaisquer dados relacionados.

- INSERT: insere um arquivo *.csv* em uma tabela temporária na base, então, é lido o mapeamento para transferir os dados na tabela passada por parâmetro já existente.
- REMAP: modifica a definição de mapeamento
- UPDATE: recarrega os dados após um remapeamento das definições
- GENERATE REPORT: gera um relatório com as equivalências entre a tabela de entrada e a base de dados alvo. Isto facilita a criação de um arquivo de definição.

4 WORKFLOW DE EXTRAÇÃO E ANÁLISE DE DADOS ABERTOS EM SAÚDE

4.1 VISÃO GERAL DO WORKFLOW DE EXTRAÇÃO E ANÁLISE DE DADOS ABERTOS EM SAÚDE

O objetivo desta seção é apresentar a arquitetura do workflow, para que durante cada seção individual de cada componente, se tenha mais clareza do seu papel no sistema como um todo. O workflow apresentado neste trabalho pode ser resumido nos seguintes passos:

1. Download dos dados do DataSus
2. Conversão para o formato *.csv* através do script conversor
3. Inserção dos dados *.csv* no SGBD MongoDB através da linha de comando do HOTMapper
4. Realização das consultas OLAP através de requisições via API RESTful do BlenDB

A figura 4.1 ilustra o processo do workflow de dados de saúde pública realizado neste trabalho.

4.1.1 Requisitos e configurações

Este trabalho foi desenvolvido em um computador com a especificação: 4Gb de memória RAM, processador Intel I3. Para realizar o trabalho foi necessário

- Sistema Gerenciador de Banco de Dados MonetDB (para o HOTMapper, será onde serão guardados os dados)
- Python 3.6 ou acima (para o Conversor, o HOTMapper e o BlenDB)
- Java versão 8.1 (Para o BlenDB)

Para a facilidade de instalação dos requisitos, foram utilizados containers Docker. Tanto o MonetDB quanto o BlenDB foram hospedados na máquina local.

4.2 DATASUS E DADOS ABERTOS DE SAÚDE

O DataSus é o órgão do Sistema Único de Saúde (SUS) brasileiro responsável pela manutenção e administração de vários sistemas de informação em saúde e portanto dos dados de saúde governamentais. Dentre os vários sistemas abrangidos pelo DataSus, por fins de recorte, ou seja, para evitar que o grande número de sistemas do DataSus interferisse no andamento do trabalho, foram utilizados apenas os dados de mortalidade e nascidos vivos oriundos, respectivamente, do Sistema de Informação de Mortalidade (SIM) e Sistema de Informação de Nascidos Vivos (SINASC). Os dados dos 2 sistemas foram obtidos nesse link `http://datasus.saude.gov.br/informacoes-de-saude/servicos2/transferencia-de-arquivos`.

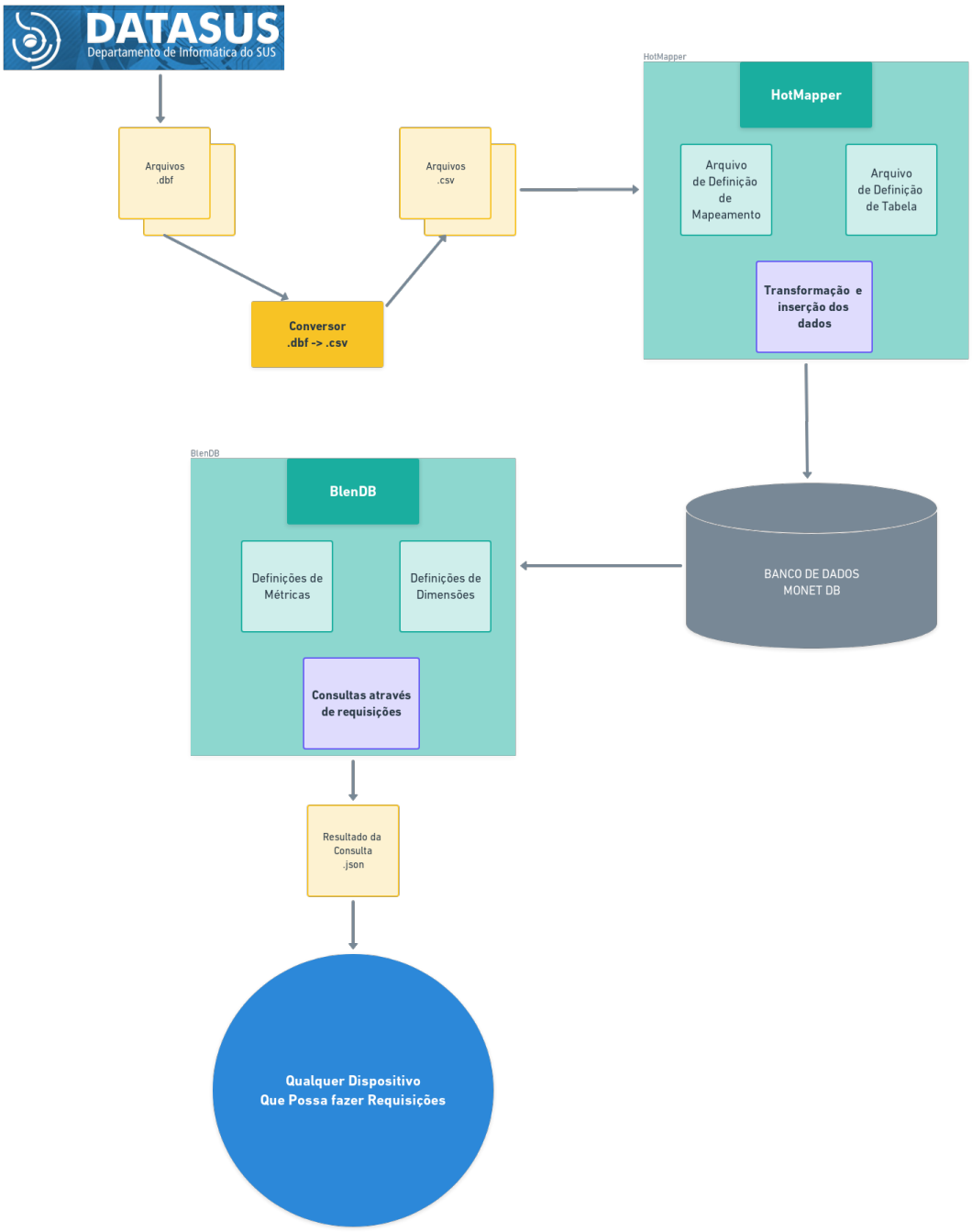


Figura 4.1: Esquematização do WorkFlow. O autor

4.2.1 Dados de mortalidade - SIM

Conforme explicado na fundamentação teórica, o SIM possui uma estrutura capilar para captação de dados, ou seja, ele é presente e abrangente nos municípios brasileiros. Foram utilizados os dados de óbito de cidadãos residentes em cidades paranaenses de 2015 a 2017. OBS: Existem campos que permitem a identificação pessoal dos indivíduos das fichas, o DataSus os deixa intencionalmente em branco por questões éticas. Portanto, não há a necessidade de um comitê de ética em pesquisa, mesmo se tratando de dados de seres humanos.

4.2.2 Estrutura dos dados

A figura 4.2 mostra a ficha de declaração de óbito (DO) e os campos que ela possui. Dentre os vários campos da ficha, os campos mais importantes para este trabalho foram:

- **SEXO** Um código para o sexo do indivíduo: 1 - Masculino; 2 - Feminino; 9 - Indefinido.
- **CODMUNRES** O código do IBGE do município de residência do indivíduo (ex: 41090 - Curitiba)
- **CODESTAB** O código do Cadastro Nacional de Estabelecimento de Saúde (CNES) do estabelecimento onde ocorreu o óbito (ex:6388671 - Hospital do Idoso Zilda Arns)
- **RACACOR** Um código para a etnia do indivíduo: 1 - Branca; 2 - Preta; 3 - Amarela; 4 - Parda
- **CAUSABAS** A causa básica do óbito representada pelo Código Internacional de Doenças (CID) (ex: X70. - "Lesão auto-provocada intencionalmente por enforcamento, estrangulamento e sufocação"
- **IDADE** A idade do óbito codificada por um número de três dígitos. O primeiro indica em que unidade de data estão os outros dois, ele pode ser: 0 - Minutos, 1 - Horas, 2 - Dias, 3 - Meses, 4 - Anos. Então por exemplo "121"Significa 21 Horas de vida, "458"Significa que era um(a) adulto(a) de 58 anos

4.2.3 Dados de natalidade - SINASC

O SINASC é muito semelhante ao SIM, tanto na sua distribuição, captação de dados e disponibilização, a única diferença entre eles é que um sistema trata de dados de mortalidade o outro de dados de nascidos vivos. A figura 4.3 mostra a ficha de declaração de nascido vivo (DN) e os campos que ela possui, muitos campos são iguais aos campos das declarações de óbito. Dentre os vários campos da ficha de, os campos mais importantes para este trabalho foram:

- **SEXO** Um código para o sexo do indivíduo: 1 - Masculino; 2 - Feminino; 9 - Indefinido.
- **CODMUNRES** O código do IBGE do município de residência do indivíduo (ex: 41090 - Curitiba)
- **CODESTAB** O código do Cadastro Nacional de Estabelecimento de Saúde (CNES) do estabelecimento onde ocorreu o nascimento (ex:6388671 - Hospital do Idoso Zilda Arns)
- **RACACOR** Um código para a etnia do indivíduo: 1 - Branca; 2 - Preta; 3 - Amarela; 4 - Parda
- **PESO** O Peso do indivíduo representado em gramas

4.2.4 Dados Complementares

As informações dos municípios nos dados do SIM e do SINASC estão codificadas, isto torna difícil a análise posterior, para melhorar a legibilidade, foi adicionada a tabela municípios, que contém o código do IBGE, o nome do município e sua respectiva regional de saúde. A regional de saúde é uma divisão organizacional de municípios entre o nível municipal e estadual.

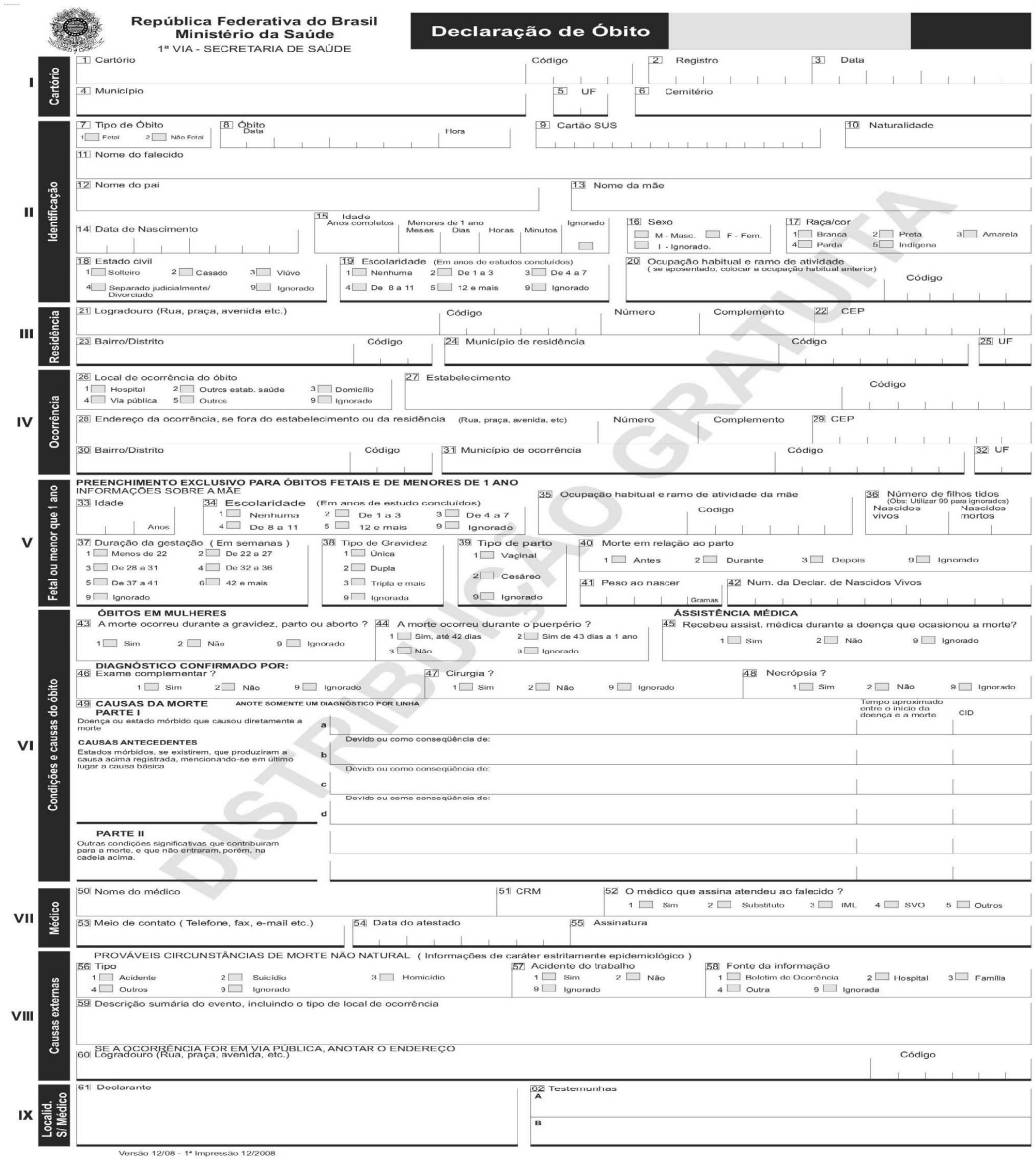


Figura 4.2: Modelo Aberto de Declaração de Óbito (da Saúde Brasileiro, 2019)

4.2.5 O Formato *.DBF e .DBC*

Apesar dos dados estarem públicos, o formato que o Datasus permite o Download não é um formato comum. O formato *.DBC* significa *Data Base Compressed* é a forma compactada do mesmo arquivo em formato *.DBF Data Base File*. O problema é que o algoritmo de descompressão deste formato exclusivo é próprio do Datasus. Os dados públicos são disponibilizados neste formato *.DBC*, foi necessário a utilização de uma biblioteca da comunidade python chamada PYsus[1], para a construção de um *script* conversor de *.DBC* para *.csv* que é o formato aceito pelo Hotmapper, o próximo passo do workflow. O script simplesmente encontra todos os arquivos com a extensão *.DBC*, descompacta os dados de cada arquivo, armazena cada um em uma variável do tipo Pandas Dataframe e exporta esses Dataframes em arquivos *.csv*. Ao final desta etapa temos os arquivos de óbitos e nascidos vivos residentes do paraná dos anos 2015, 2016 e 2017 em *.csv*.

[1]https://pypi.org/project/PySUS/

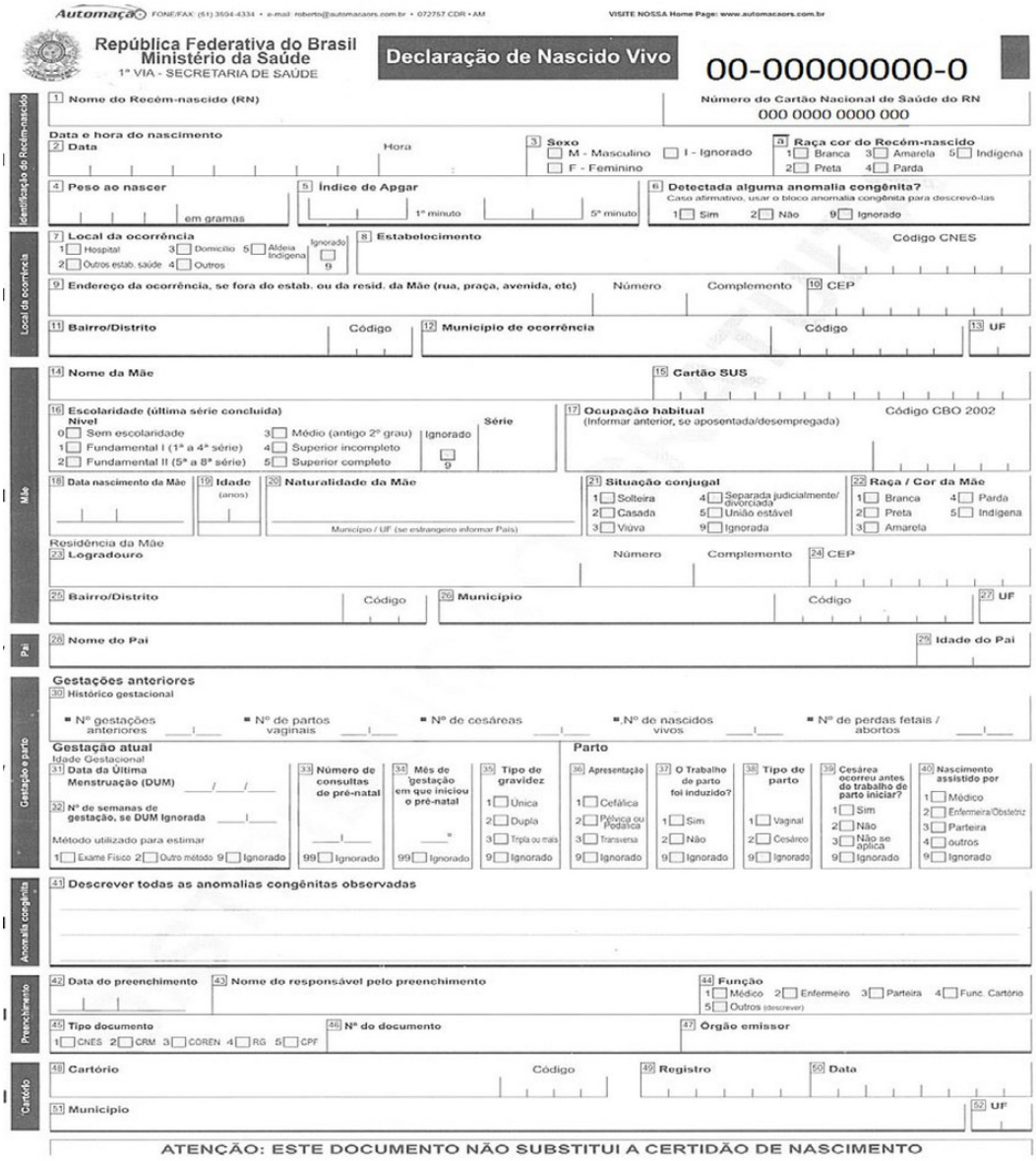


Figura 4.3: Modelo Aberto de Declaração de Nascido Vivo (da Saúde Brasileiro, 2019)

4.3 UTILIZANDO HOTMAPPER PARA SALVAR NO BANCO DE DADOS

4.3.1 O que é o HOTMapper

Conforme explicado na seção 3.2 nos trabalhos relacionados, o HOTmapper é o acrônimo para *Historical Open Data Table Mapper*, ou seja, ele nada mais é que um mapeador de dados abertos. O HOTMapper propõe que, o problema de agrupar uma base de dados que possua muitos registros ao longo do tempo, seja realizado manualmente apenas uma vez, o usuário deve ciar uma definição de tabela, que será o *schema* final no MonetDB onde serão agrupados os dados, e definições de mapeamento da base para todos os anos que ela possui.

4.3.2 Arquivos e Utilização do HOTMapper

Como o HOTMapper é, no fim de tudo, uma ferramenta para agrupar dados em um SGDB MonetDB, deve-se ter uma instância do MonetDB rodando no computador. O primeiro passo é criar uma tabela no MonetDB, utilizando o comando CREATE no terminal, exemplo:

```
python manage.py create <table_name>
```

Aplicando:

```
python manage.py create dopr
```

Neste exemplo o parâmetro *dopr* é o nome da tabela que será criada no MonetDB, no anexo B, existe o arquivo *dopr.json*, que é o arquivo de definição da tabela. Ao rodar o comando, o HOTMapper verifica se existe algum arquivo de definição com o mesmo nome da tabela que foi passada por parâmetro, lê esse arquivo e por fim constrói a tabela no SGBD de acordo com as especificações. O mesmo processo foi feito com a tabela auxiliar de municípios e com a tabela de nascidos vivos.
OBS: no caso das declarções de óbito e declarações de nascido vivo, esperava-se que apenas os campos *numerodo* e *numerodn* fossem suficientes como chaves primárias, porém existe o problema da duplicidade, que é quando duas fichas físicas são cadastradas com o mesmo número, por causa disso a chave primária foi a combinação dos respectivos campos com o *codinst*.
Após a criação da tabela o próximo passo é a inserção dos dados dos arquivos *.csv* para o MonetDB, utilizando o seguinte comando no terminal:

```
python manage.py insert <full/path/for/the/file>
<table_name> <year> [--sep separator] [--null null_value]
```

Aplicando:

```
python manage.py insert ../sus_data/DOPR2015.csv dopr 2015 --sep=';'
```

Ao rodar o comando, o HOTMapper irá procurar um arquivo de definição de mapeamento com o nome *dopr.csv*, irá ler o arquivo e procurar uma coluna com o mapeamento de cada campo para o ano de 2015, para cada linha, o HOTMapper fará o mapeamento especificado naquela linha. O mapeamento aceita incluir transformações, o que é útil nos casos em que se tem um código simples (ex: LOCALOCOR 1-Hospital, 2-Outro Estabelecimento de saúde, 3-Domicílio, 4-Via Pública).
Apos realizar o mapeamento, o HOTMapper realizará a inserção dos dados mapeados no SGBD. O mesmo processo foi feito com os dados de nascidos vivo e os dados auxiliares dos municípios.

4.4 UTILIZANDO BLENDB PARA FAZER CONSULTAS OLAP

De acordo com o que foi explicado na fundamentação teórica, o modelo multidimensional possui dimensões e fatos (também chamados de métricas). A maioria dos indicadores de saúde importantes sempre estão relacionados a contagem de óbitos de algum lugar. Portanto o fato (também chamado de métrica) definido foi a contagem total de registros pelo campo *NUMERODO*, a chave primária da tabela. As dimensões (como se pode dividir essas contagens) foram variadas em local e em categoria. Sendo elas:

- **SEXO** O sexo do indivíduo, pode ser masculino, feminino ou indefinido. Como foi "traduzido"na etapa do HOTMapper, ao invés de "1","2","9", aparecerá "M","F","I".
- **RACACOR** A Etnia do indivíduo, pode ser branca, preta, parda, amarela ou indefinida. Como foi "traduzido"na etapa do HOTMapper, ao invés de "1","2","3","4","9"aparecerá "branca","preta", "parda", "amarela", "indefinida".

- **REGIONAL** A regional de saúde, ao contrário das outras dimensões que eram puramente categóricas, esta é uma dimensão geográfica, visto que a regional de saúde é o nível de organização entre o estado e o município. Como a declaração de óbito contém apenas o código do município, esta dimensão é gerada com o uso do arquivo auxiliar que relaciona municípios e regionais de saúde.

- **NOME_MUN_OCOR** O Nome do município de ocorrência do óbito. Esta também é uma dimensão geográfica. Muito utilizada na hora de avaliar a qualidade de saúde a nível municipal. Como a declaração de óbito contém apenas o código do município, esta dimensão foi gerada com o uso do arquivo auxiliar que relaciona código do município com o seu respectivo nome.

O apêndice C contém o arquivo *dopr.yaml* este foi o arquivo construído que especifica as métricas e dimensões.
Com o arquivo pronto, e o ambiente do BlenDB ativo, pode-se fazer as requisições e observar os resultados.

5 ESTUDO DE CASO

5.1 REQUISIÇÕES E RESULTADOS

A seguir serão mostrados algumas requisições pela API do BlenDB e o respectivo resultado nele retornado. Para fins de legibilidade resultados muito grandes serão recortados. A seguir, a quantidade total de óbitos por município de ocorrência do Paraná acumulada nos anos de 2015,2016 e 2017.

`http://localhost:3000/v1/data?metrics=met:numerodo&dimensions=dim:sexo,dim:nome_mun_ocor&sort=dim:nome_mun_ocor&filters=dim:nome_mun_ocor==ABATIA`

```
dim:nome_mun_ocor: "MORRETES"
met:numerodo: 107

dim:nome_mun_ocor: "CORNELIO PROCOPIO"
met:numerodo: 830

dim:nome_mun_ocor: "AMAPORA"
met:numerodo: 24

dim:nome_mun_ocor: "TELEMACO BORBA"
met:numerodo: 600

dim:nome_mun_ocor: "CAMPO MOURAO"
met:numerodo: 1397

*
*
*
```

A seguir, a quantidade total de óbitos do Paraná por sexo, acumulada nos anos de 2015,2016 e 2017 com o filtro do município de Abatiá. `http://localhost:3000/v1/data?metrics=met:numerodo&dimensions=dim:sexo&sort=dim:nome_mun_ocor&filters=dim:nome_mun_ocor==ABATIA`

```
"dim:sexo":"M"
"dim:nome_mun_ocor":"ABATIA"
"met:numerodo":82

"dim:sexo":"F"
"dim:nome_mun_ocor":"ABATIA"
"met:numerodo":35
```

Para comparar, os resultados foi comparado com a consulta em linguagem SQL realizada no arquivo original antes do workflow. Não houve nenhuma discrepância.

6 CONCLUSÃO

Neste trabalho foi desenvolvido um workflow de dados abertos em saúde pública vindos do DataSus utilizando a ferramenta HOTMapper como ferramenta de ETL, o MonetDB como banco de dados final e o BlenDB como provedor de uma API para consultas OLAP seguindo algumas métricas e dimensões de dados de mortalidade e nascidos vivos do Paraná dos anos de 2015, 2016 e 2017.

Este trabalho visa unificar as bases separadas por ano do SIM e SINASC, visa traduzir alguns campos que estão codificados, para melhorar a legibilidade e visa facilitar a geração de informações facilitando a linguagem de consulta através de dimensões e métricas.

Os resultados foram comparados com a consulta em SQL no próprio arquivo antes do workflow.

Futuramente este trabalho pode ser expandido utilizando o mesmo workflow para a agregação de mais anos, também de outras bases como por exemplo o Sistema de Internação Hospitalar.

REFERÊNCIAS

(2019). Blendb. `https://gitlab.c3sl.ufpr.br/c3sl/blendb`. Acessado em 26/11/2019.

(2019). Monetdb. `http://monetdb.org/`. Acessado em 26/11/2019.

da Saúde Brasileiro, M. (2019). Datasus. `http://datasus.saude.gov.br/`. Acessado em 26/11/2019.

Date, C. J. (2000). *Introdução a Sistemas de Bancos de Dados*. Elsevier.

e Valduriez, O. (2011). *Principles of Distributed Database Systems*. Springer.

Ehrenfried, H. V., Eckelberg, R., Iboshi, H., Todt, E., Weingaertner, D., and Fabro, M. D. D. (2019a). Hotmapper: Historical open data table mapper. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, pages 550–553.

Ehrenfried, H. V., Eckelberg, R., Iboshi, H., Todt, E., Weingaertner, D., and Fabro, M. D. D. (2019b). Hotmapper: Historical open data table mapper. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, pages 550–553.

Kimball, R. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley.

Navathe, E. . (2011). *Sistemas de bancos de dados*. Pearson Education do brasil.

Parnia, A. (2000). *User Interface design for open data platforms*. PhD thesis, Faculty EEMCS, Delft University of Technology, Delft, The Netherlands.

Silberschatz, A. (2011). *Database System Concepts*. McGraw-Hill.

APÊNDICE A – CONVERSOR DE .DBC PARA .CSV

Como explicado no capítulo da proposta, os dados públicos do Datasus vêm no formato próprio .DBC. A seguir é mostrado o Script que converte estes arquivos para o formato .csv

Por fins de comodismo, ele foi programado para converter todos os arquivos .DBC na pasta.

```
import pandas as pd
from pysus.utilities import readdbc
from pathlib import Path

input_dir='/home/daniel/tcc/sus_data/original_dbc/'
output_dir='/home/daniel/tcc/sus_data/formatted_csv/'

dbc_files=list(Path(input_dir).glob('*.dbc'))
print('Buscando em {}'.format(input_dir))
print('Foram encontrados {} arquivos .dbc'.format(str(len(dbc_files))))
for dbc_file in list(dbc_files):
   file_name=dbc_file.name
   print('Converting {}'.format(str(file_name)))
   data = readdbc.read_dbc(filename=str(dbc_file))
   data.to_csv(path_or_buf=Path(output_dir).with_name(file_name).\
      with_suffix('.csv'),sep=';',index_label=False)
```

APÊNDICE B – ARQUIVOS DE DEFINIÇÃO DE TABELAS DO HOTMAPPER

A definições das tabelas de dados de mortalidades e de dados de município. No caso da tabela de mortalidade, a chave primária foi uma combinação do número da DO e do dígito identificador. A tabela de municípios apenas precisou do código do IBGE como chave primária

```
{
   "pairing_description": "Tabela de Declarações de Óbito - Paraná",
   "data_source": "Dados abertos DATASUS",
   "pk": ["numerodo","codinst"],
   "foreign_keys": [ ]
}
```

```
{
   "pairing_description": "Tabela de Declarações de Nascido Vivo - Paraná",
   "data_source": "Dados abertos DATASUS",
   "pk": ["numerodn","codinst"],
   "foreign_keys": ["numerodo"]
}
```

```
{
   "pairing_description": "Tabela de municípios - Paraná",
   "data_source": "Dados abertos DATASUS",
   "pk": ["cod_mun"],
   "foreign_keys": [ ]
}
```

APÊNDICE C – ARQUIVOS DO BLENDB

O BlenDB precisa apensa que se diga quais são as dimensões e as métricas para cada campo

```
tags:
   links: []
   obj:
      -
         name: "noDescription"
         description: "Related with seller"
views:
   links: []
   obj:
      -
         alias: "view:dopr"
         origin: true
         aliasAsName: true
         dimensions:
            - "teste"
            - "dim:sexo"
            - "dim:codmunocor"
            - "dim:cod_mun"
            - "dim:nome_mun_ocor"
         metrics:
            - "met:numerodo"
            - "met:sexo"
      -
         alias: "municipio"
         origin: true
         aliasAsName: true
         dimensions:
            - "nome_mun"
            - "regional"
         metrics:
            - "cod_mun"
metrics:
   links: []
   obj:
      -
         name: "met:numerodo"
         dataType: "integer"
         aggregation: "count"
         description: ""
         tags:
            - "noDescription"

      -
         name: "met:sexo"
         dataType: "integer"
         aggregation: "count"
         description: ""
         tags:
            - "noDescription"

      -  name: "cod_mun"
```

```
52          dataType: "integer"
53          aggregation: "count"
54          description: ""
55          tags:
56             - "noDescription"
57
58
59 dimensions:
60    links: []
61    obj:
62       -
63          name: "dim:sexo"
64          dataType: "integer"
65          description: "Colocar descrição"
66       -
67          name: "teste"
68          dataType: "integer"
69          description: "Colocar descrição"
70       -
71          name: "dim:cod_mun"
72          dataType: "string"
73          description: "Colocar descrição"
74
75       -
76          name: "dim:codmunocor"
77          dataType: "string"
78          description: "Colocar descrição"
79       -
80          name: "dim:nome_mun_ocor"
81          dataType: "string"
82          description: "Colocar descrição"
83       -
84          name: "nome_mun"
85          dataType: "integer"
86          description: "Colocar descrição"
87       -
88          name: "regional"
89          dataType: "integer"
90          description: "Colocar descrição"
91
92 enumTypes:
93    links: []
94    obj:
95       - []
96 sources:
97    links: []
98    obj: []
```