

Projeto Final

Dados abertos do Bolsa Família

Dayanne Fernandes da Cunha, 13/0107191
Christian Costa Werner , 14/0134573

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
Banco de Dados - Turma A

dayannefernandesc@gmail.com, ccwerner96@gmail.com

Abstract. *This report corresponds to the step-by-step project of creating a database using open data from the federal government (transparency portal). We will answer some questions about the Bolsa Familia project and develop a CRUD system using this database.*

Resumo. *Este relatório corresponde ao passo a passo do projeto de criação de um banco de dados utilizando dados públicos do governo federal (portal da transparência). Iremos sanar algumas questões sobre o projeto Bolsa Família e implementar um sistema de CRUD utilizando este banco criado.*

Sumário

1	Introdução	3
1.1	Motivação	3
1.2	Programa Social - Bolsa Família	3
1.2.1	Sobre o Programa	3
1.2.2	Volume de recursos envolvidos	4
1.3	O Projeto	5
2	Diagrama de Entidade Relacionamento (DER)	5
2.1	Características dos atributos do DER	6
2.2	Entidades do DER	7
2.3	Relações do DER	8
3	Modelo Relacional (MR)	9
4	Formas Normais	9
4.1	1FN	10
4.2	2FN	10
4.3	3FN	10

5	Extract, Transform, Load (ETL)	11
5.1	Extract	11
5.2	Transform	12
5.3	Load	12
6	Camadas do Sistema para o CRUD	12
6.1	Persistência	13
6.2	Apresentação	13
7	Consultas	13
7.1	Queries	13
7.2	Views	13
7.3	Procedures	14
7.4	Triggers	14
8	Análise dos Resultados e Considerações Finais	14

1. Introdução

1.1. Motivação

Este relatório se deve ao Projeto de Banco de Dados da disciplina de Banco de Dados da Universidade de Brasília (UnB), referente ao segundo semestre de 2016, turma A. Esse projeto consiste em montar um Banco de Dados com os dados do governo do programa Bolsa Família - setembro de 2015, segundo disponível no [Portal da Transparência 2004].

Este relatório abrange o que é o programa social Bolsa Família e todo um passo-a-passo de criação de um banco de dados, partindo de um arquivo maciço CSV (*Comma-Separated Values*) usando as mais variadas ferramentas (para verificar as ferramentas, visite o item 1.3 e confira o *README* do projeto).

1.2. Programa Social - Bolsa Família

1.2.1. Sobre o Programa

O Programa Bolsa Família - PBF é um programa de transferência de renda do Governo Federal, sob condicionalidades, instituído no Governo Lula pela Medida Provisória 132, de 20 de outubro de 2003, convertida em lei em 9 de janeiro de 2004, pela Lei Federal n. 10.836.

Tendo sido apelidado de *Programa 1335 – Transferência de Renda com Condicionalidades – Bolsa Família*, foram geradas ações de controle com o fito de avaliar a efetiva aplicação dos recursos destinados ao cumprimento da finalidade constante da *Ação 8442 – Transferência de Renda Diretamente às Famílias em Condições de Pobreza e Extrema Pobreza*, pertencente ao *Programa Bolsa Família - PBF*.

O objetivo do programa *1335 – Transferência de Renda com Condicionalidades – Bolsa Família* é Combater a fome, a pobreza e outras formas de privação das famílias e promover a segurança alimentar e nutricional e o acesso à rede de serviços públicos de saúde, educação e assistência social, criando possibilidades de emancipação sustentada dos grupos familiares e de desenvolvimento local dos territórios.

O objetivo da ação *8442 – Transferência de Renda Diretamente às Famílias em Condições de Pobreza e Extrema Pobreza* é Melhorar as condições socioeconômicas das famílias pobres e extremamente pobres por meio de transferência direta de renda.

A gestão do PBF é descentralizada e compartilhada por União, estados, Distrito Federal e municípios. Estes entes federados trabalham em conjunto para aperfeiçoar, ampliar e fiscalizar a execução do Programa, instituído pela Lei 10.836/04 e regulamentado pelo Decreto nº 5.209/04.

As famílias beneficiárias são selecionadas com base nas informações inseridas pelo município no Cadastro Único para Programas Sociais (Cadastro Único). O Cadastro é um instrumento de coleta de dados que tem como objetivo identificar todas as famílias de baixa renda existentes no País. A seleção dos beneficiários é realizada de forma automatizada, utilizando-se como principal critério a renda per capita da família.

Os Órgãos Federais envolvidos na execução do Programa são: a SECRETARIA NACIONAL DE RENDA DE CIDADANIA DO MINISTÉRIO DO DESENVOLVIMENTO SOCIAL E COMBATE À FOME - SENARC/MDS (UG: 550007), responsável

pela coordenação, gestão e operacionalização do Programa Bolsa Família; a CAIXA ECONÔMICA FEDERAL (CAIXA), com a função de Agente Operador do PBF, mediante remuneração e condições pactuadas no contrato com o MDS; o MINISTÉRIO DA SAÚDE – MS, por meio da Secretaria de Atenção a Saúde (UG 250010), responsável pela gestão federal do acompanhamento do cumprimento das condicionalidades de saúde das famílias beneficiárias do PBF; o MINISTÉRIO DA EDUCAÇÃO - MEC, por meio da Secretaria de Educação Continuada, Alfabetização e Diversidade (UG: 150028), responsável pela gestão federal do sistema de frequência escolar dos alunos; e a SECRETARIA NACIONAL DE ASSISTÊNCIA SOCIAL DO MINISTÉRIO DO DESENVOLVIMENTO SOCIAL E COMBATE À FOME – SNAS/MDS (UG 550011), responsável em realizar a coleta e o registro periódico das informações referentes à condicionalidade da assistência social.

Os demais órgãos responsáveis, governos estaduais com a função de apoiar os municípios na implementação do Programa; governos municipais e Distrito Federal, responsáveis pela gestão local do Programa Bolsa Família e do Cadastro Único; e Órgãos de Controle Social, responsáveis por garantir aos cidadãos espaço para o acompanhamento do Programa, atuam visando assegurar os interesses da sociedade, bem como permitir que suas demandas e necessidades sejam apresentadas ao poder público.

1.2.2. Volume de recursos envolvidos

Os volumes de recursos orçamentários destinados à Ação 8442 - Transferência de Renda Diretamente às Famílias em Condição de Pobreza e Extrema Pobreza (Lei nº 10.836/2004) para os exercícios entre 2007 a 2012 estão a seguir representados:

Ação 8442- Transferência de Renda Diretamente às Famílias em Condição de Pobreza e Extrema Pobreza	
Ano	Valor dotado na LOA - atualizado
2007	R\$ 8.755.556.796,00
2008	R\$ 10.548.462.345,00
2009	R\$ 11.844.280.000,00
2010	R\$ 13.057.000.000,00
2011	R\$ 16.699.040.000,00
2012	R\$ 19.964.030.000,00
Fonte: SIGplan de 2007 a 2011, LOA/2012 e Siafi Gerencial 2012.	

Figura 1. Quantidade de recursos envolvidos investido no PBF nos anos de 2007 a 2012.

Podemos ver também, com dados mais atualizados, a correção monetária sobre os investimentos do PBF, e que, embora crescente, vem tendo um ritmo desacelerado, devido à crise financeira que vem assolando o Brasil nos últimos tempos. Segue então um infográfico simples que revela o investimento no PBF e as variações nominal e real nos anos de 2010 a 2016:

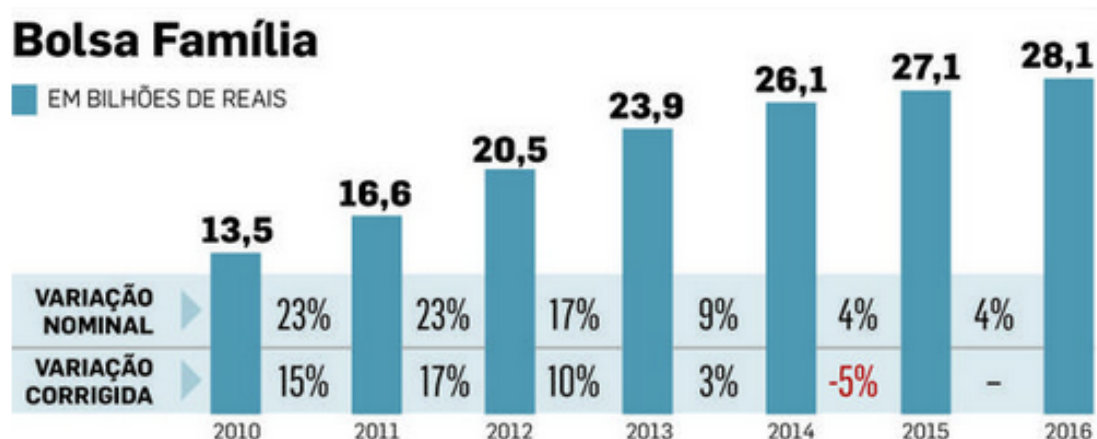


Figura 2. Infográfico representando o investimento em bilhões de reais e as variações nominal e corrigida sobre o PBF nos anos de 2010 a 2016.

Os dados referentes à setembro de 2015 serão abordados posteriormente junto ao desenvolvimento do banco.

1.3. O Projeto

Este projeto foi desenvolvido com o fim de poder analisar os dados abertos do Programa Bolsa Família – PBF e também realizar CRUD (Create, Read, Update e Delete) no sistema. Todos os algoritmos, scripts e imagens comentadas neste relatório podem ser encontrados no repositório aberto do GitHub criado para o desenvolvimento deste projeto: https://github.com/Dayof/OpenData_BolsaFamilia.

2. Diagrama de Entidade Relacionamento (DER)

O Diagrama de Entidade Relacionamento do sistema, apresentado na Figura 3 foi feito pensando já no MR, e depois alterado conforme a normalização no MR foi sendo realizada. A ferramenta utilizada para desenhar o diagrama foi a [LucidChart 2016]. Temos no DER 7 entidades, das quais duas são entidades fracas.

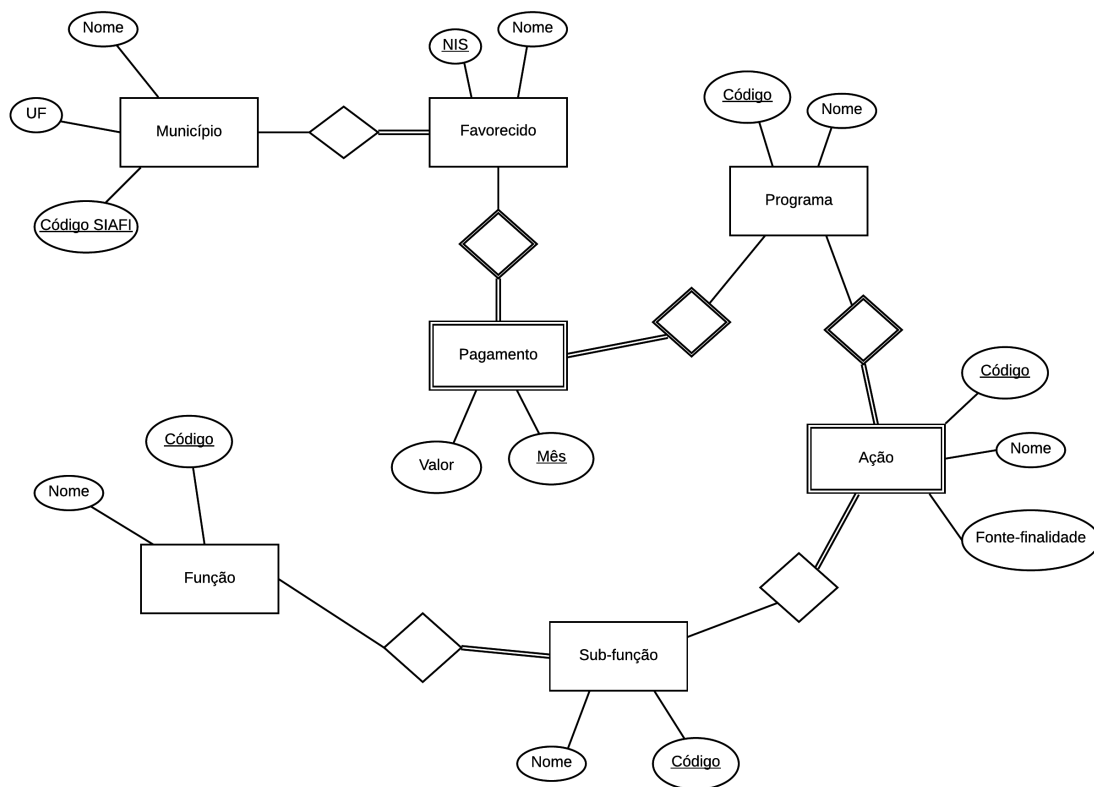


Figura 3. Diagrama de entidade relacionamento do banco de dados dos pagamentos do Programa Bolsa Família (PBF).

A entidade **PAGAMENTO** é na verdade uma relação de multiplicidade N:N entre as entidades **FAVORECIDO** e **PROGRAMA**, porém, devido o programa utilizado para desenhar o DER, o [LucidChart 2016] apresentar problemas para inserir as multiplicidades então foi criado outra entidade para representar esta relação.

2.1. Características dos atributos do DER

- **UF [MUNICIPIO]**
Siglas dos estados. Valores repetem na tabela original do Bolsa Família.
- **Nome [MUNICIPIO]**
Nome de cada município de origem dos favorecidos, são relacionados ao Código SIAFI do município. Valores repetem na tabela original do Bolsa Família.
- **Código SIAFI [MUNICIPIO]**
É o Sistema Integrado de Administração Financeira do Governo Federal que consiste no principal instrumento utilizado para registro, acompanhamento e controle da execução orçamentária, financeira e patrimonial do Governo Federal. Valor é único para cada nome de município distinto. Valores repetem na tabela original do Bolsa Família.
- **Nome [FAVORECIDO]**
Nome de cada favorecido que recebeu pagamento do programa do bolsa família no ano de 2015 e mês de Setembro. Às vezes as folhas de pagamento contém

retroativos de pagamento para algum favorecido, portanto, os valores de nome dos favorecidos repetem na tabela original do Bolsa Família.

- **NIS [FAVORECIDO]**

É uma solução que permite a identificação do trabalhador nos diversos cadastros, bem como do cidadão brasileiro beneficiário de Programas Sociais e/ou que se enquadre nas condições estabelecidas pelas Políticas Públicas de Governo Federal, Estadual ou Municipal. O NIS - Número de Identificação Social é um número de cadastro e devem ser cadastrados: o trabalhador, vinculado à empresa privada, cooperativa ou empregador pessoa física; os beneficiários de Programas Sociais (cadastrados pelo agente definido pelo Gestor do Programa); o diretor não-empregado quando optante pelo FGTS e os beneficiários de Políticas Públicas (cadastrados pela SRTE, MS e MEC). Como foi citado acima as folhas de pagamento contém retroativos de pagamento para algum favorecido, portanto, os valores de NIS dos favorecidos também repetem na tabela original do Bolsa Família.

- **Código [PROGRAMA]**

O código do programa bolsa família. Além disso, as famílias que atendem aos critérios do Programa Bolsa Família e estão inscritas em outros programas federais também têm direito ao benefício ([CaixaGov]). Os valores do código de programa podem variar em outras planilhas, porém, como está em estudo somente o programa do bolsa família o valor do código é único.

- **Nome [PROGRAMA]**

Nome oficial do programa do bolsa família oferecido pelo Governo Federal.

- **Código [ACAO]**

Código da ação do bolsa família. Análogo ao código de programa.

- **Nome [ACAO]**

Nome oficial da ação do bolsa família oferecido pelo Governo Federal.

- **Fonte-Finalidade [ACAO]**

Possui valor igual para todos os dados da planilha coletada do portal da transparência.

- **Nome [SUBFUNCAO]**

Nome da classificação sub-funcional da despesa, exemplo: Ação Legislativa, Controle externo, etc.

- **Código [SUBFUNCAO]**

Código da classificação sub-funcional da despesa.

- **Nome [FUNCAO]**

Nome da classificação funcional da despesa, exemplo: Legislativa, Judiciária, etc.

- **Código [FUNCAO]**

Código da classificação funcional da despesa.

2.2. Entidades do DER

- **MUNICIPIO**

Temos que a entidade **MUNICIPIO** possui 3 atributos: Nome, Código SIAFI (PK) e UF. Decidimos colocar UF como atributo pois não haveria utilidade de uma entidade para UF em si. O Código SIAFI foi escolhido como PK pela característica de ser único para cada valor de nome de município como foi descrito na seção anterior.

- **FAVORECIDO**

Tem 2 atributos: Nome e NIS (PK). O NIS foi escolhido como PK pela característica de ser único para cada valor de nome de favorecido como foi descrito na seção anterior.

- **PAGAMENTO**

Consiste de 2 atributos próprios: Valor e Mês, o Mês é uma das chaves PK. A chave também é composta pelo NIS do favorecido relacionado e Código do programa que o favorecido. A decisão da chave ser composta por estes 3 atributos é devido à repetição dos valores de Código do programa e NIS do favorecido por causa dos retroativos da folha, porém, dois favorecidos de um mesmo programa não podem receber dois pagamentos do mesmo mês e ano, portanto, se tornando crucial a composição destes 3 atributos.

- **PROGRAMA**

Possui dois atributos: Código (PK) e Nome. O Código do programa foi escolhido como PK pela característica de ser único para cada valor de nome do programa como foi descrito na seção anterior.

- **ACAO**

Possui 3 atributos próprios: Código (PK), Nome e Fonte-finalidade. O código da ação é PK da entidade pois ele é único para cada valor de nome da ação e fonte-finalidade como foi descrito na seção anterior.

- **SUBFUNCAO**

Contém 2 atributos: Nome e Código (PK). O código da subfunção é PK da entidade pois ele é único para cada valor de nome da subfunção como foi descrito na seção anterior.

- **FUNCAO**

Contém 2 atributos: Nome e Código (PK). O código da função é PK da entidade pois ele é único para cada valor de nome da função como foi descrito na seção anterior.

2.3. Relações do DER

- **MUNICIPIO R FAVORECIDO - 1:N**

Relação de participação parcial x Relação de participação total

Um município pode conter N favorecidos e 1 favorecido só pode vir de um município.

- **FAVORECIDO R PAGAMENTO - 1:N**

Relação de participação parcial x Relação de participação total e de identificação

Um favorecido pode receber N pagamentos e 1 pagamento de determinado mês/ano pode ir somente para um favorecido.

- **PAGAMENTO R PROGRAMA - N:1**

Relação de participação total x Relação de participação parcial

Um pagamento é direcionado a um determinado programa do governo e um programa pode emitir N pagamentos para os favorecidos.

- **PROGRAMA R ACAO - 1:N**

Relação de participação parcial x Relação de participação total e de identificação

Um programa pode conter N ações e uma ação pertence a um programa.

- **ACAO R SUBFUNCAO - N:1**

Relação de participação total x Relação de participação parcial

Uma ação advém de uma subfunção e uma subfunção pode realizar N ações.

- **SUBFUNCAO R FUNCAO - N:1**

Relação de participação total x Relação de participação parcial

Uma subfunção pertence a uma determinada função geral e uma função possui N subfunções.

3. Modelo Relacional (MR)

O Modelo Relacional do sistema, apresentado na Figura 4 foi feito utilizando o software [MySQLWorkbench 2016]. Foi utilizado o modelo de diagrama do tipo EER (Enhanced Entity–Relationship), que difere pelo ER (Entity–Relationship) por ter possibilidade de adicionar especializações das entidades, agregação, associação, etc. Este tipo de modelo também é utilizado para apresentar um modelo relacional de tabelas.

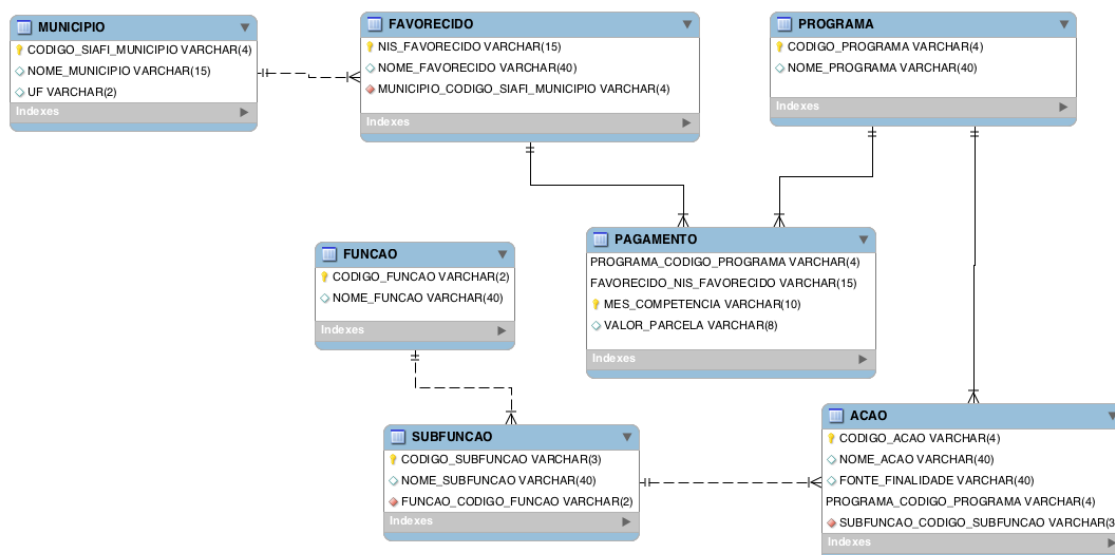


Figura 4. Modelo relacional do banco de dados dos pagamentos do Programa Bolsa Família (PBF).

As entidades, atributos e relações do MR foram descritas na Seção 2.

A partir deste MR o software disponibiliza gerar um *script* de criação das tabelas apresentadas no diagrama, portanto, este foi gerado e utilizado. Na Seção 5 iremos explicar melhor como foi realizada a parte de ETL (Extract, Transform, Load) das tabelas normalizadas apresentadas na Figura 4.

4. Formas Normais

Como foi possível observar na Seção 2 e Seção 3 já foram apresentadas soluções normalizadas pois a tabela original que foi obtida através de um arquivo do tipo CSV possui 12 atributos na mesma tabela. Na Seção 5 será exposto mais informações a respeito da extração, transformação e população das tabelas cruciais ao sistema aqui apresentado.

4.1. 1FN

A primeira forma normal (1FN) apresenta uma solução onde toda tabela é minimamente normalizada, sendo, o valor de cada coluna indivisível:

(UF, CODIGO-SIAFI-MUNICIPIO, NOME-MUNICIPIO, CODIGO-FUNCAO,
CODIGO-SUBFUNCAO, CODIGO-PROGRAMA, CODIGO-ACAO,
NIS-FAVORECIDO, NOME-FAVORECIDO, FONTE-FINALIDADE,
VALOR-PARCELA, MES-COMPETENCIA)

Nesta forma os resultados apresentam muita redundância de dados, principalmente ao lidar com um banco de dados extenso como este. Também possui problemas de inserção, remoção e atualização. Logo na próxima sub-seção vamos apresentar uma forma normal mais enxuta.

4.2. 2FN

Avaliando às dependências funcionais dos atributos é possível gerar a segunda forma normal (2FN). Se a tabela está em 1FN e todo atributo do complemento de uma chave candidata é **totalmente funcionalmente dependente** daquela chave então a tabela estará em 2FN:

1ª Tabela
(CODIGO-SIAFI-MUNICIPIO) → NOME-MUNICIPIO, UF
2ª Tabela
(NIS-FAVORECIDO) → NOME-FAVORECIDO
3ª Tabela
(CODIGO-PROGRAMA, CODIGO-ACAO) → CODIGO-FUNCAO,
CODIGO-SUBFUNCAO, FONTE-FINALIDADE
4ª Tabela
(NIS-FAVORECIDO, MES-COMPETENCIA, CODIGO-PROGRAMA) →
VALOR-PARCELA

Mesmo na forma 2FN ainda possuem atributos transitivamente dependentes, logo, na próxima sub-seção vamos estudar estes casos para uma maior melhora das tabelas.

4.3. 3FN

A transitividade de dependência nos permite melhorar as tabelas encontradas da sub-seção anterior, portanto, se uma tabela está na forma 2FN e todos os atributos não-chave forem **dependentes não-transitivos** de chave primária teremos uma relação de normalização 3FN:

1ª Tabela
(CODIGO-SIAFI-MUNICIPIO) → NOME-MUNICIPIO, UF
2ª Tabela
(NIS-FAVORECIDO) → NOME-FAVORECIDO
3ª Tabela
(CODIGO-PROGRAMA, CODIGO-ACAO) → CODIGO-SUBFUNCAO,
FONTE-FINALIDADE

4ª Tabela
(NIS-FAVORECIDO, MES-COMPETENCIA, CODIGO-PROGRAMA) →
VALOR-PARCELA

5ª Tabela
(CODIGO-SUBFUNCAO) → NOME-SUBFUNCAO

6ª Tabela
(CODIGO-FUNCAO) → NOME-FUNCAO

A 3ª tabela foi alterada pois o CODIGO-FUNCAO é dependente transitivo da chave composta (CODIGO-PROGRAMA, CODIGO-ACAO). A tabela 6 serviu para separar esta dependência transitiva da tabela 3.

A tabela 5 surgiu somente para fins de clareza do significado do CODIGO-SUBFUNCAO do programa e ação. O atributo de NOME-FUNCAO foi adicionado na tabela 6 pela mesma justificativa.

5. Extract, Transform, Load (ETL)

A extração, transformação e carregamento dos dados foi realizada de acordo com os seguintes passos:

1. Fazer o download dos arquivos CSV disponíveis no [PortaldaTransparência 2004];
2. Extrair os arquivos baixados que vieram em formato ZIP de compreensão na pasta de desenvolvimento do projeto;
3. Estudar o conteúdo dos arquivos extraídos;
4. Pré-processar os arquivos extraídos para importar para um formato de banco de dados;
5. Importar os arquivos CSV pra um arquivo único de DB (Database);
6. Criar tabelas e popular as tabelas normalizadas utilizando as tabelas originais extraídas.

As etapas 1 e 2 pertencem à parte de extração dos dados, já as etapas 3, 4 e 5 fazem parte do processo de transformação dos dados. A etapa 6 é elemento do processo de carga de dados.

5.1. Extract

A extração dos dados foi obtida através do [PortaldaTransparência 2004]. Foi possível obter dados aberto sobre o Programa Bolsa Família (PBF), utilizando como objeto de estudo um determinado ano e mês. Também foram extraídos dados aberto sobre a Função e Subfunção de recursos e gastos diretos do governo federal.

A primeira extração foi sobre os pagamentos da Bolsa Família para cada favorecido na data de Setembro/2015. A segunda extração foi sobre a Classificação Funcional da Despesa (MTO), onde, possui informações em formato aberto da classificação funcional da Despesa, publicada no Manual Técnico de Orçamento, pelo Ministério do Planejamento, do ano de 2015.

O [PortaldaTransparência 2004] disponibiliza arquivos de formato CSV (Comma-separated values) sobre todos os dados lá contidos. Os arquivos CSV foram baixados e extraídos na pasta de desenvolvimento do projeto.

5.2. Transform

Como foi dito na etapa 3, os arquivos foram estudados e analisados antes da importação para um arquivo de banco de dados. Na análise foram descobertos dois problemas:

PROBLEMA 1: Os arquivos não eram de fato do formato CSV e sim TSV (Tab-separated values).

PROBLEMA 2: Os nomes dos atributos de cada "coluna" possuíam codificação não padrão, ou seja, acento nas palavras.

Sendo assim, diante destes 2 problemas encontrados foram pesquisadas soluções para tratar os dados extraídos.

SOLUÇÃO PARA O PROBLEMA 1: Utilizar SQLite 3 e Python 3 para tratar as importações dos arquivos TSV para arquivos DB, pois, eles dão suporte para importar este tipo de formato.

SOLUÇÃO PARA O PROBLEMA 2: Implementar um script para substituir a primeira linha do arquivo extraído com nomes válidos para tratar.

Dada as soluções acima elas foram executadas e os resultados foram obtidos com sucesso. Estes problemas foram resolvidos nas etapas 4 e 5 do processo de ETL.

5.3. Load

Na etapa 6 ficou estipulado que as tabelas obtidas do MR apresentado na Seção 3 seriam criadas e populadas a partir das tabelas originais extraídas das fontes externas de dados abertos.

Após a extração e tratamento destes dados foi alcançada a parte de carga de dados nas tabelas normalizadas 3FN.

- Primeiramente o script SQL de criação das tabelas gerado pelo software [MySQLWorkbench 2016] foi modificado para se adequar à linguagem aceita pelo SQLite 3;
- Foram estudadas "QUERIES" para extrair os dados das tabelas originais e inserir nas tabelas normalizadas de forma adequada;
- Implementação das "QUERIES";
- Execução das "QUERIES" utilizando Python 3 e o SQLite 3 (processo bastante demorado).

Entretanto, pela demora de carregamento e processamento dos dados, foi decidido que **apenas 1000 linhas** das 13 milhões do banco seriam utilizadas, devido a tabela de pagamentos. Desse modo, as perguntas gerais, situadas na seção 8 não poderão ser respondidas nesse sistema simplificado.

Por fim, ao final do processo de ETL os dados estão disponíveis para realizar o CRUD (Create, Read, Update e Delete).

6. Camadas do Sistema para o CRUD

Com os dados prontos para realizar o CRUD (Create, Read, Update e Delete) chega o momento de implementar um sistema para ler estes dados e apresentar a algum

usuário do sistema. O *Design Pattern* de Persistência e Apresentação foi sugerido para aplicação. Como já estava sendo utilizado a linguagem de programação Python 3 no processo de importação dos dados abertos, foi então considerado em dar continuidade à utilização da linguagem para desenvolvimento do sistema.

Por experiências prévias com Web Service com Python 3 e o Framework Web Flask de um membro da equipe que produziu este sistema foi então decidido a utilização destas ferramentas para implementar o sistema que irá executar o CRUD.

6.1. Persistência

A camada de persistência foi implementada utilizando o SQLite 3 e Python 3 para ler e escrever os dados no arquivo de banco de dados. Tanto a leitura quanto a escrita de algum dado seria passada através de alguma requisição do usuário na camada de Apresentação.

6.2. Apresentação

Foram utilizadas ferramentas como HTML e CSS para desenhar a camada de apresentação ao usuário. A ponte entre os *templates* HTML e a camada de persistência é feita pelo Framework Web Flask.

7. Consultas

Foram elaborados scripts de consultas para interagir com o banco de dados obtido além do CRUD básico. As consultas elaboradas servem para facilitar a obtenção e leitura de dados do banco de dados. Abordaremos as *queries*, *procedures*, *views* e *triggers* utilizados. Como o SQLite 3 não tem suporte para as *procedures* padrões de outros SGBDs, não foi possível implementá-los como esperado neste projeto, porém, foi realizada outra solução de "*procedure*" condicional que funciona quase como uma *procedure* padrão.

7.1. Queries

As *queries* realizadas tem como objetivo responder às seguintes perguntas:

- Quem foi a pessoa que mais recebeu recurso do governo em um determinado ano/mês?
- Qual o estado que mais recebeu recurso?
- Quem são todos os favorecidos do programa Bolsa Família de Setembro de 2015 organizados em ordem alfabética?
- Quantos favorecidos existem por estado?
- Qual a média das parcelas?
- Qual a média de pagamentos por estado?

7.2. Views

A *view* que utilizamos tem como objetivo mostrar os favorecidos que mais foram beneficiados nesse mês de setembro de 2015. Essa *view* nos mostra o NIS e o nome do favorecido, o total recebido e o mês em que foi recebido, ordenados pelo total recebido por cada pessoa, do mais beneficiado ao menos beneficiado.

7.3. Procedures

Como dito anteriormente, a biblioteca que usamos, SQLite 3, não tem suporte para *procedures* padrões de outros SGBDs, logo, foi realizada outra solução de "*procedure*" condicional que funciona quase como uma *procedure* padrão. A solução utilizada abrange a utilização do SQLite 3 junto da linguagem de programação usado para interagir com a persistência, que no caso seria o Python 3.

Foi implementado uma função em Python e esta foi mapeada para interagir com as *queries* do SQLite. O método utilizado nesta solução pertence à biblioteca do SQLite 3 e tem como assinatura a seguinte descrição: `create_function("name_function_sql", params, name_function_python)`.

A função utilizada no projeto tem como objetivo buscar todos os favorecidos que tenham recebido ordem de pagamento de valores maiores ou iguais a R\$600,00.

7.4. Triggers

O *trigger* utilizado tem o fim de gerar um *drop* para cada pagamento de um favorecido que foi removido da tabela. Ou seja, sempre que um favorecido for removido da tabela de favorecidos, todos os pagamentos referentes a esse favorecido também serão removidos. Caso não haja pagamento ao favorecido, nenhum pagamento será removido.

8. Análise dos Resultados e Considerações Finais

Foi implementado um sistema capaz de ler e escrever dados abertos de Setembro de 2015 sobre o programa Bolsa Família, com o objetivo responder algumas questões levantadas sobre esta base de dados como:

- Quem foi a pessoa que mais recebeu recurso do governo em um determinado ano/mês?
- Qual o estado que mais recebeu recurso?

Foi descoberto no processo de ETL do banco de dados que existe mais de um pagamento direcionado a um mesmo favorecido, do mesmo programa no mesmo mês/ano, logo, por este motivo, a chave composta não foi suficiente para identificar unicamente a ordem de pagamento, por isso, foi então decidido que a tabela teria um campo extra para ser utilizado como chave unitária, denominado por *cod_pag*, sendo ele de *auto_increment*. Essa alteração, entretanto, acarretou em mudanças no BD, já que a entidade *Pagamento* deixa de ser entidade fraca, e as relações entre *Favorecido* e *Programa* deixam de ser de identificação.

Além disso, nesse período de setembro de 2015, foram registradas 13 912 767 entradas de pagamento, deixando o processamento realmente lento. E, para dificultar o processo, para cada FK o SGBD conferia se existia tal chave nas outras tabelas, o que deixou o processo de população das tabelas exponencialmente lento. Foram gastos 4 dias de ETL sem sucesso para importar completamente os dados para o banco do sistema.

Por fim, por causa da lentidão no processamento de dados, não foi possível analisar o banco de dados inteiro. Desse modo, foi utilizado **apenas as primeiras 1000 tuplas** do banco, limitando as tabelas a um máximo de 1000 linhas. Somente assim foi possível de extrair alguns dados do banco e testar as consultas. Por causa dessa redução de dados, não foi possível responder às perguntas gerais citadas acima.

Todo o sistema e scripts de ETL estão no repositório aberto do GitHub criado para o desenvolvimento deste projeto: https://github.com/Dayof/OpenData_BolsaFamilia. As bases de dados de formato CSV e DB não estão neste repositório remoto devido à extensão destes.

Referências

- [CaixaGov] CaixaGov. Caixa - programas sociais - bolsa família. <http://www.caixa.gov.br/programas-sociais/bolsa-familia/Paginas/default.aspx>. [Online; accessed 23-November-2016].
- [CNM] CNM. Cnm - confederação nacional de municípios. www.cnm.org.br. [Online; accessed 23-November-2016].
- [LucidChart 2016] LucidChart (2016). Lucid chart. <https://www.lucidchart.com/demo>. Software.
- [MySQLWorkbench 2016] MySQLWorkbench (2016). Mysql workbench, software de modelagem de banco de dados. <http://www.mysql.com/products/workbench/>. Software.
- [PortaldaTransparência 2004] PortaldaTransparência (2004). Portal da transparência - governo federal - ministério da transparência, fiscalização e controladoria-geral da união. <http://www.portaltransparencia.gov.br>. [Online; accessed 23-November-2016].