

Documentação do Projeto Integrador

Grupo H - Health Insurance Marketplace

Modelo de Dados para os Arquivos de Uso Público do Health Insurance Exchange (PUFs) da CMS

Baseado nas informações extraídas dos *data dictionaries* que você forneceu, podemos construir um modelo de dados para os arquivos PUFs do Health Insurance Exchange. Este modelo ainda requer refinamentos após uma análise mais completa, pois algumas relações entre tabelas precisam de maior clareza nos dicionários fornecidos. Em particular, a chave estrangeira entre as tabelas não está totalmente explícita em seus dicionários.

Entidades e Relacionamentos (Modelo Conceitual):

Este modelo assume que cada arquivo PUF representa uma tabela no banco de dados. As relações são sugeridas com base nas descrições dos dicionários, mas podem precisar de ajustes após análise mais detalhada.

- **tb_issuer:** (Chave Primária: IssuerId)
 - IssuerId (VARCHAR(5)): Código numérico de 5 dígitos que identifica a seguradora no Health Insurance Oversight System (HIOS).
 - FederalTIN (VARCHAR): Tax Identification Number da seguradora. (Presente em alguns PUFs, mas não em todos).
 - IssuerName (VARCHAR): Nome da seguradora (não presente nos dicionários, adicionar).
- **tb_plan:** (Chave Primária: PlanId)
 - PlanId (VARCHAR): Identificador único do plano de saúde (formato pode variar entre os anos).
 - PlanMarketingName (VARCHAR): Nome de marketing do plano.
 - HIOSProductId (VARCHAR): Identificador do produto HIOS.
 - HPID (VARCHAR): Identificador HPID (National Health Plan Identifier).
 - PlanType (VARCHAR): Tipo de plano (ex: Individual, Familiar).
 - MetalLevel (VARCHAR): Nível de metal do plano (ex: Bronze, Prata, Ouro, Platina).
 - BusinessYear (INT): Ano do plano.
 - IssuerId (VARCHAR(5)): Chave estrangeira para tb_issuer.
 - NetworkId (VARCHAR): Chave estrangeira para tb_network (se aplicável).
 - ServiceAreaId (VARCHAR): Chave estrangeira para tb_service_area.
 - FormularyId (VARCHAR): Identificador do formulário (se aplicável).
 - ... (Outras colunas dos PUFs Plan Attributes, Benefits and Cost Sharing, Business Rules após análise mais detalhada da redundância)
- **tb_service_area:** (Chave Primária: ServiceAreaId)
 - ServiceAreaId (VARCHAR): Identificador único da área de serviço.
 - ServiceAreaName (VARCHAR): Nome da área de serviço.
 - StateCode (VARCHAR(2)): Código do estado (abreviação de dois caracteres).
 - County (VARCHAR): Nome do condado.

- ZipCodes (VARCHAR): Lista de códigos postais (considerar normalização em tabela separada se necessário).
 - BusinessYear (INT): Ano da informação.
 - IssuerId (VARCHAR(5)): Chave estrangeira para tb_issuer.
- **tb_network:** (Chave Primária: NetworkId)
 - NetworkId (VARCHAR): Identificador único da rede de provedores.
 - NetworkName (VARCHAR): Nome da rede de provedores.
 - NetworkURL (VARCHAR): URL da rede de provedores.
 - BusinessYear (INT): Ano da informação.
 - IssuerId (VARCHAR(5)): Chave estrangeira para tb_issuer.
- **tb_benefits_cost_sharing:** (Chave Primária: BenefitId - Sugestão: usar chave auto-incremental)
 - BenefitId (INT): Identificador único do benefício (usar auto-incremental).
 - PlanId (VARCHAR): Chave estrangeira para tb_plan.
 - StandardComponentId (VARCHAR): Identificador do componente padrão do benefício.
 - BenefitName (VARCHAR): Nome do benefício.
 - CopayInnTier1, CopayInnTier2, CopayOutofNet (DECIMAL): Valores de coparticipação.
 - CoinsInnTier1, CoinsInnTier2, CoinsOutofNet (DECIMAL): Valores de coseguro.
 - ... (Outras colunas do BenCS-PUF).
- **tb_rate:** (Chave Primária: RateId - Sugestão: usar chave auto-incremental)
 - RateId (INT): Identificador único da taxa (usar auto-incremental).
 - PlanId (VARCHAR): Chave estrangeira para tb_plan.
 - RatingAreaId (VARCHAR): Identificador da área de classificação.
 - Tobacco (VARCHAR): Indicador de fumante.
 - Age (INT): Faixa etária.
 - IndividualRate, IndividualTobaccoRate, ... (DECIMAL): Taxas para diferentes perfis.
 - BusinessYear (INT): Ano da informação.
- **tb_plan_crosswalk:** (Chave Primária composta: PlanID_2015, IssuerID_2015)
 - PlanID_2015 (VARCHAR): Identificador do plano no ano anterior.
 - IssuerID_2015 (VARCHAR(5)): Identificador da seguradora no ano anterior.
 - PlanID_2016 (VARCHAR): Identificador do plano no ano atual (pode ser nulo).
 - IssuerID_2016 (VARCHAR(5)): Identificador da seguradora no ano atual (pode ser nulo).
 - ... (Outras colunas do CW-PUF).
- **tb_business_rules:** (Chave Primária: BusinessRuleId - Sugestão: chave auto-incremental)
 - BusinessRuleId (INT): Identificador único da regra de negócio (auto-incremental).
 - IssuerId (VARCHAR(5)): Chave estrangeira para tb_issuer.
 - ProductId (VARCHAR): Identificador do produto.
 - ... (Outras colunas do BR-PUF).

Observações Importantes:

- **Chaves Estrangeiras:** As chaves estrangeiras precisam ser explicitamente definidas entre as tabelas. Os dicionários não fornecem essa informação de forma clara, necessitando de uma análise manual dos dados para confirmar as relações.
- **Normalização:** Este modelo pode exigir normalização adicional para remover redundância e melhorar a integridade dos dados. Por exemplo, a tabela `tb_plan` pode ser dividida em múltiplas tabelas para organizar melhor os atributos do plano.
- **Tipos de Dados:** Os tipos de dados foram definidos como `VARCHAR` e `INT` como uma aproximação, mas podem precisar de ajustes dependendo do conteúdo real dos dados (por exemplo, usar `DECIMAL` para valores monetários).
- **Tabelas Adicionais:** As tabelas `Quality PUF`, `Nationwide Quality Rating System PUF`, `Enrollee Survey PUF`, `Machine-readable URL PUF`, e `Transparency in Coverage PUF` precisam ser analisadas individualmente para determinar suas colunas e seus relacionamentos com as tabelas já definidas.
- **Limpeza e Transformação de Dados:** Antes de carregar os dados no banco de dados, é crucial realizar a limpeza e transformação de dados para lidar com valores nulos, valores inconsistentes e outros problemas de qualidade de dados.

Este modelo fornece uma estrutura inicial. A criação de um modelo de dados completo e robusto requer uma análise detalhada dos dados reais, utilizando os dicionários para determinar as chaves e restrições apropriadas para cada coluna e tabela. Recomenda-se o uso de ferramentas de modelagem de banco de dados para criar um diagrama ER completo e visualizar as relações entre as entidades.

Objetivos:

As questões de pesquisa a serem respondidas neste trabalho são:

1. Como as taxas e benefícios do plano variam entre os estados?
2. Como os benefícios do plano se relacionam com as taxas do plano?
3. Como as taxas do plano variam por idade?
4. Como os planos variam entre os provedores de rede de seguros?
5. Qual a progressão de preço dos planos por ano?

Para responder às questões de pesquisa, o modelo de dados proposto precisa de alguns ajustes e adições para garantir que as informações necessárias estejam disponíveis e acessíveis de forma eficiente. O modelo original era um ponto de partida, e agora o refinamos para atender aos objetivos específicos.

Ajustes e Adições ao Modelo de Dados:

1. **Tabela de Estados (`tb_state`):** Adicionar uma tabela para armazenar informações sobre os estados:

sql

```
CREATE TABLE tb_state (  
    StateCode VARCHAR(2) PRIMARY KEY,  
    StateName VARCHAR(255)  
);
```

As tabelas tb_plan, tb_service_area e tb_network devem conter uma chave estrangeira para tb_state.

2. **Tabela de Faixas Etárias (tb_age_range):** Para facilitar a análise da variação de preços por idade:

sql

```
CREATE TABLE tb_age_range (  
    AgeRangeId INT PRIMARY KEY,  
    AgeRangeDescription VARCHAR(255)  
);
```

A tabela tb_rate deve conter uma chave estrangeira para tb_age_range.

3. **Relacionamento mais explícito entre tb_plan e tb_benefits_cost_sharing:** A chave estrangeira PlanId em tb_benefits_cost_sharing precisa ser explicitamente definida e talvez renomeada para PlanVariantId para refletir a natureza variante dos benefícios:

sql

```
ALTER TABLE tb_benefits_cost_sharing  
    ADD CONSTRAINT fk_plan_variant FOREIGN KEY (PlanVariantId) REFERENCES tb_plan(PlanId);
```

4. **Relacionamento entre tb_plan e tb_rate:** A chave estrangeira PlanId em tb_rate precisa ser explicitamente definida:

sql

```
ALTER TABLE tb_rate  
    ADD CONSTRAINT fk_plan_rate FOREIGN KEY (PlanId) REFERENCES tb_plan(PlanId);
```

5. **Adição de Coluna de Preço em tb_rate:** A tabela tb_rate precisa de colunas para representar os valores das taxas para cada tipo de plano familiar (Individual, Casal, Família com 1 dependente, etc.) . Ajuste os nomes das colunas de acordo com as informações do seu dicionário de dados.

sql

```
ALTER TABLE tb_rate  
    ADD COLUMN IndividualRate DECIMAL(10, 2),  
    ADD COLUMN CoupleRate DECIMAL(10, 2),  
    ADD COLUMN Family1DependentRate DECIMAL(10, 2),  
    ADD COLUMN Family2DependentRate DECIMAL(10, 2),  
    ADD COLUMN Family3MoreDependentRate DECIMAL(10, 2);
```

Respondendo as Perguntas de Pesquisa:

Com este modelo de dados ajustado, podemos responder às questões de pesquisa através de consultas SQL:

1. **Variação de taxas e benefícios entre estados:**

sql

```
SELECT s.StateName, AVG(r.IndividualRate) AS AverageRate, COUNT(*) AS NumPlans  
FROM tb_plan p  
JOIN tb_benefits_cost_sharing bcs ON p.PlanId = bcs.PlanVariantId
```

```
JOIN tb_state s ON p.StateCode = s.StateCode
JOIN tb_rate r ON p.PlanId = r.PlanId
GROUP BY s.StateName
ORDER BY AverageRate DESC;
```

-- Consultas similares podem ser feitas para outros tipos de plano e benefícios.

2. **Relação entre benefícios e taxas:** Isso requer uma análise correlacional. Você precisará usar uma ferramenta de análise de dados ou uma linguagem de programação (ex: Python com pandas) para calcular a correlação entre as colunas de custo (em tb_benefits_cost_sharing) e taxas (em tb_rate).

3. **Variação de taxas por idade:**

sql

```
SELECT ar.AgeRangeDescription, AVG(r.IndividualRate) AS AverageRate
FROM tb_rate r
JOIN tb_age_range ar ON r.AgeRangeld = ar.AgeRangeld
GROUP BY ar.AgeRangeDescription
ORDER BY ar.AgeRangeld;
```

4. **Variação de planos entre provedores:**

sql

```
SELECT n.NetworkName, COUNT(DISTINCT p.PlanId) AS NumPlans
FROM tb_plan p
JOIN tb_network n ON p.NetworkId = n.NetworkId
GROUP BY n.NetworkName
ORDER BY NumPlans DESC;
```

5. **Progressão de preços por ano:**

sql

```
SELECT BusinessYear, AVG(IndividualRate) AS AverageIndividualRate
FROM tb_rate
GROUP BY BusinessYear
ORDER BY BusinessYear;
```

Lembre-se que estas são apenas consultas exemplares. Você precisará adaptá-las para refletir a nomenclatura exata das colunas em suas tabelas e agregar outras colunas relevantes para análise. A complexidade das consultas dependerá da granularidade dos dados e das especificações adicionais necessárias para a sua pesquisa.