# Estratégia

Após uma breve observação nas tabelas do banco de dados legado, podemos perceber que apesar da utilização de colunas com o nome “ID” havia uma grande redundância nos dados como visto na figura abaixo:b
tabela agendamentos:

| # cod_convenio | convenio |
|---|---|
| 5 | Hospital |
| 3 | Migração |
| 2 | Particular |

tabela pacientes:

| # id_conv | convenio |
|---|---|
| 5 | Hospital |
| 2 | Particular |

Entao mesmo havendo um relacionamento entre as duas tabelas isso poderia ser ignorado para a criação das tabelas do banco de dados legado;

Para a criaçao das tabelas foi utilizado o script “0temp_schema.sql” localizado na mesma pasta que os demais arquivos do projeto;

A importação dos dados originais para o banco de dados legado se deu por meio de pegar todas as linhas do arquivo csv (exceto a primeira linha, pois a mesma não tinha dados apenas o nome das colunas) e coloca-las em um array multidimensional, onde a posição X era o dado completo e as posiçoes Y de X eram os dados de cada coluna; Além disso houve a necessidade de modificar as datas presentes nas tabelas originais, pois estavam em formato Brasileiro.

Após isso o Array foi transformado em uma única String em formato SQL para uma única inserção no banco de dados Legado, dado que isso consumiria menos recursos do que várias inserções para cada índice do Array;

Para a migração dos dados do sistema legado para o medical_challenge algumas alterações nos dados foram necessárias:

A colunas “sex_pac” do sistema legado estava com dados no formado “M” ou “F” enquanto a coluna “sex” do sitema nova estava no formato “Masculino” e “Feminino”

```
insert_into_db($connTemp ,"UPDATE pacientes SET sex_pac =
  CASE
    WHEN sex_pac = 'M' THEN 'Masculino'
    WHEN sex_pac = 'F' THEN 'Feminino'
    ELSE sex_pac
  END
;");
```

A coluna “medico” do sistema legado havia os nomes incompletos, então para não havar uma duplicação de dados com os já registrados na coluna nome da tabela profissionais, houve um update no nome dos médicos:

```
insert_into_db($connTemp, "UPDATE agendamentos SET medico =
  CASE
    WHEN medico = 'Pietro' THEN 'Dr. Analista Pietro'
    WHEN medico = 'Dr. Lucas' THEN 'Dr. Lucas KNE'
    ELSE medico
  END
;");
```

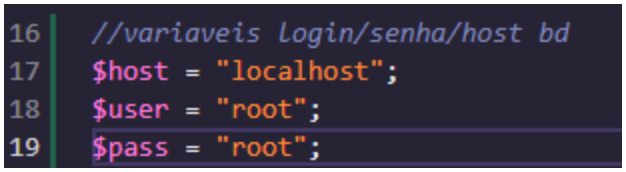
A coluna cod_paciente do banco de dados legado foi utilizada como valores para a coluna cod_referencia na tabela pacientes do banco de dados medical_challenge.

Após essas alterações para cada tabela no banco de dados medical_challenge foi utilizado um select especifico no banco de dados legado (respeitando os relecionamentos do banco de dados medical_challenge e verificando se não havia duplicação de dados) para resgastar os dados, os mesmo foram inseridos em um array, onde foi transformado em uma única string e inserido no banco de dados novo;

# Observações

1° Executar os scripts *“medical_challenge_schema.sql*” e “*0temp_schema.sql*” para a criação dos schemas Banco Legado e Banco Medical Challenge;

2° Caso *Host, User ou Passoword* do banco de dados seja diferente de “*localhost*”, “*root”* e “*root*” respectivamente, modificar os valores das variaveis: “$host” “$user” e “$pass” mostradas abaixo:



```php
//variaveis login/senha/host bd
$host = "localhost";
$user = "root";
$pass = "root";
```

3° Habilitar a extenção mysqli_connect no php.ini

# Documentação

## funções:

**open_csv**
open_csv - Recebe o path de um arquivo csv e retorna o resource do mesmo, ou false em caso de falha;
*syntax:*
open_csv(string $csv_path): resource|false

**csv_rows**
csv_rows - Recebe o resouce de um arquivo .csv e retorna um array com as linhas do arquivo csv;
*syntax:*
csv_rows(resourse $file): array

**format_date**
format_date - Recebe um array com dates e o indice das mesma e as formata para o padrão americano (y-m-d);
*syntax:*
csv_rows(array $datas)
**add_values**
add_values - Recebe um array de dados e retorna os valores em uma única string SQL
*syntax:*
add_values(array $datas, $columns_name = [], $default_value = 0): string
Sem o segundo parâmetro, add_values() não irá adicionar os nomes da coluna na string;
sem o terceiro parâmetro, add_values() irá adicionar “DEFAULT” como valor para a primeira coluna do banco de dados;

**set_general_value**

set_general_value - Recebe um array multidimensional e adiciona o valor passado em todas as posições fornecidas;
*syntax:*
set_general_values(array $array, string|int $value, int $pos): array

**set_specif_value**
set_specif_value - Recebe um array multidimensional e adiciona o valor passado apenas na posição passada;
*syntax:*
set_specif_values(array $array, int $posX, string|int $value, int $pos): array

**replace_last**
replace_last - Recebe uma string e retorna a mesma trocando os dois elementos passados;
*syntax:*
replace_last(string $str, string $char, string $replace): string

**prepare_sql_to_insert**
prepare_sql_to_insert - Recebe uma string e o nome da  tabela do banco de dados, une os mesmo com “INSERT IGNORE INTO” e retorna a string com o SQL pronto;
*syntax:*
prepare_sql_to_insert(string $sql_values, string $table_name): string

**get_columns_name**
get_columns_name - Recebe uma conexão com o banco de dados, o nome de uma tabela e retorna o nome de todas as colunas daquela tabela
*syntax:*
get_columns_name(mysqli $conn, string $table_name): array

**insert_into_db**
insert_into_db - Recebe uma conexão com o banco de dados, um SQL de INSERT e roda o SQL no banco de dados;
*syntax:*
insert_into_db(mysqli $conn, string $sql)

**select_into_db**
select_into_db - Recebe uma conexão com o banco de dados, um SQL de SELECT, roda o SQL no banco de dados e retorna o resultado;
*syntax:*

select_into_db(mysqli $conn, string $sql) : array