Trabalho Final de SQL para Ciência de Dados - CI245

Análise de Dados com SQL sobre Ocupações dos Leitos de COVID nos Anos de 2020, 2021 e 2022.

Table of contents

1 Introdução

Este relatório tem como objetivo realizar a análise de dados sobre a ocupação dos leitos hospitalares destinados ao tratamento de COVID-19 nos anos de 2020, 2021 e 2022, utilizando SQL como ferramenta principal. A análise foi realizada com base em um banco de dados com informações obtidas através de arquivos .csv e estão disponíveis em: https:

//opendatasus.saude.gov.br/dataset/registro-de-ocupacao-hospitalar-covid-19. Tais dizem respeito a ocupação dos leitos, registros de óbitos e outras métricas relevantes.

As principais analises realizadas foram: - Detecção de anomalias: registros com picos isolados; - Análise de Texto: origem dos dados; - Evolução das altas e óbitos ao longo do tempo.

Para tanto foram utilizadas técnicas de **SQL** para manipulação e análise dos dados, com foco na extração de informações relevantes e na identificação de padrões e anomalias.

2 Conexão com Banco de Dados

3 Limpeza de Dados

Antes de fazermos qualquer análise estatística, precisamos garantir que os dados estão modelados de tal forma que sejam íntegros e de qualidade. Não à toa, já é clichê dizer que boa parte do trabalho de qualquer analista ou cientista de dados é a limpeza dos dados, afinal a máxima "garbage in, garbage out" é uma realidade.

Essa base de dados, por serem preenchidas por humanos em alguns casos, possui alguns problemas que, a **priori**, dificultaram o nosso entendimento sobre as informações que os dados nos trazem. Lendo o dicionário de dados com acuracidade, conseguimos identificar quais padrões poderiam nos guiar na validação destes dados.

Assim, para o tratamento de dados, foram realizadas algumas etapas. Para cada uma, views foram criadas para ajudar na organização do código. 1 - Remoção (filtro) de registros (notificações) com status excluídos 2 - Remoção (filtro) de registros que continham valores decimais. 3 - Validação de colunas de acordo com informações do dicionário de dados. 4 - Remoção de notificações duplicadas 5 - Criação de colunas calculadas 6 - Criação de série temporal 7 - Execução de 'Last Observation Carried Forward (LOCF)' para tratamento de valores nulos em séries temporais. 8 - Criação de **materialized view** como tabela final para facilitar uso durante consulta dos dados tratados.

3.1 0 - Criaçao de views

Antes de executarmos o **script** que cria a **materialized view** final, garantimos que já não existe no banco de dados outros objetos com o mesmo nome. Para isso, executamos o comando **"DROP VIEW IF EXISTS nome_da_view";** para cada view que vamos criar.

3.2 1 - Remoção de registros (notificações) com status excluídos

Observamos que alguns registros continham status de exclusão com valores logicos **true**, o que não faz sentido para a análise. Assim, criamos uma view que filtra esses registros para manter apenas aqueles com status não excluídos.

3.3 2 - Remoção (filtro) de registros que continham valores decimais

Alguns valores de ocupação e saída continham valores decimais (valores relativos pois nao passavam de 100), o que não faz sentido para a análise, pois esses valores deveriam ser inteiros. Assim, criamos uma view que filtra esses registros para manter apenas aqueles com valores inteiros.

3.4 3 - Validação de colunas de acordo com informações do dicionário de dados

De acordo com o dicionário de dados, estabelecemos alguns critérios para validar os valores das colunas de ocupação e saídas:

- Se *ocupacao_hospitalar_cli* for igual a zero ou nulo, então, necessariamente, *ocupacao_covid_cli* também deve ser igual a zero ou nulo.

Table 1: 1 records

sem_logica_count	total_count	sem_logica_percentual
4723	1047382	0.0045

- Se *ocupacao_hospitalar_uti* for igual a zero ou nulo, então, necessariamente, *ocupacao_covid_uti* também deve ser igual a zero ou nulo.

Table 2: 1 records

sem_logica_count	total_count	sem_logica_percentual
9047	1047382	0.0086

- – Se *ocupacao_covid_cli* for igual a zero ou nulo, então, necessariamente, *ocupacao_suspeito_cli* e *ocupacao_confirmado_cli* também devem ser iguais a zero ou nulos.

Table 3: 1 records

sem_logica_count	total_count	sem_logica_percentual
658396	1047382	0.6286

- – Se *ocupacao_covid_uti* for igual a zero ou nulo, então, necessariamente, *ocupacao_suspeito_uti* e *ocupacao_confirmado_uti* também devem ser iguais a zero ou nulos.

Table 4: 1 records

sem_logica_count	total_count	sem_logica_percentual
657853	1047382	0.6281

Com base nessas validações, percebemos que cerca de 62% dos registros das colunas *ocupacao_suspeito_uti*, *ocupacao_confirmado_uti*, *ocupacao_suspeito_cli* e *ocupacao_confirmado_cli* apresentam inconsistências e, por isso, decidimos não utilizá-las nas análises.

Ademais, como menos de 1% dos registros das colunas *ocupacao_covid_cli* e *ocupacao_covid_uti* apresentam inconsistências, criaremos uma visualização que filtre os valores consistentes dessas colunas, conforme o **script abaixo**.

3.5 4 - Remoção de notificações duplicadas

Alguns registros continham notificações duplicadas, ou seja, com a mesma data de notificação e o mesmo hospital. Para resolver isso, criamos uma view que remove as duplicatas, mantendo apenas a notificação mais recente (com base na data de atualização e criação).

3.6 5 - Criação de colunas calculadas

Para facilitar a análise, criamos colunas calculadas que agregam informações relevantes. As colunas criadas foram: *ocupacao_covid*, *ocupacao_hospitalar*, *saida_covid_obitos*, *saida_covid_altas*, *saida_covid*, *entrada_covid_calculada*.

3.7 6 - Criação de série temporal

Para facilitar a análise temporal dos dados, criamos uma série temporal que combina todas as datas de notificação com os hospitais, locais e status. Isso nos permite analisar a evolução dos dados ao longo do tempo, mesmo quando alguns registros estão ausentes.

3.8 7 - Execução de 'Last Observation Carried Forward (LOCF)' para tratamento de valores nulos em séries temporais.

Para lidar com valores nulos na série temporal, aplicamos a técnica de "Last Observation Carried Forward (LOCF)", que preenche os valores nulos com o último valor observado (data imediatamente mais recente). Isso é especialmente útil em séries temporais, onde a continuidade dos dados é importante para a análise.

3.9 8 - Criação de materialized view como tabela final para facilitar uso durante consulta dos dados tratados.

Criamos uma **materialized view** final que contém todos os dados tratados e prontos para análise. Escolhemos esse objeto porque ele permite consultas mais rápidas e eficientes, especialmente em grandes volumes de dados. Essa tabela final é ordenada por hospital, data de notificação, local e status, facilitando a consulta e análise dos dados.

4 Analise de Dados