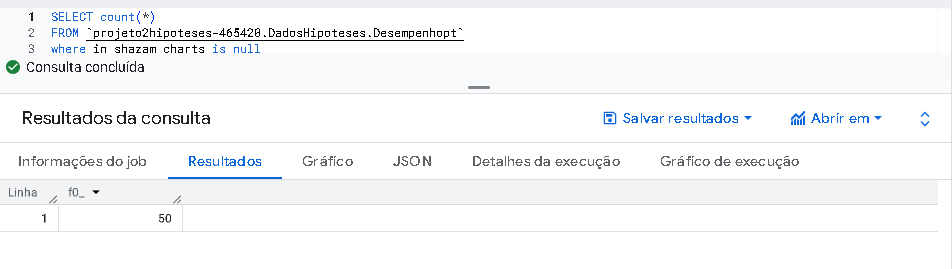
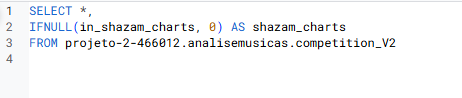
5.1.2 Valores Nulos

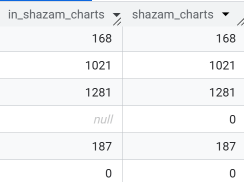


**Tabela Track*in*competition**

* **in*shazam*charts:** Presença e classificação da música nas paradas da Shazam.

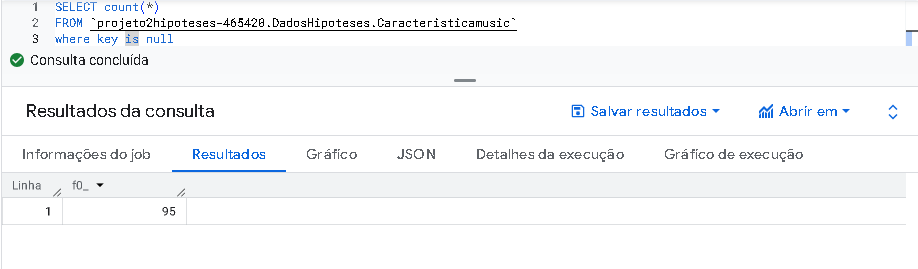
Foram encontrados **50** valores nulos na coluna **in*shazam*charts**.





Depois usei o comando except para excluir a coluna in\_shazam\_charts que continha null e salvei a versão atualizada apenas com a coluna shazam\_charts para trabalhar em cima dela criando novas variáveis

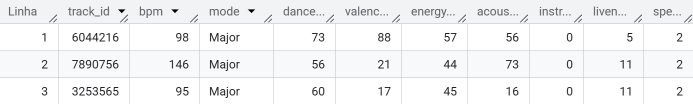




**Tabela Track*technical*info**

* **key**: Tom musical da música.

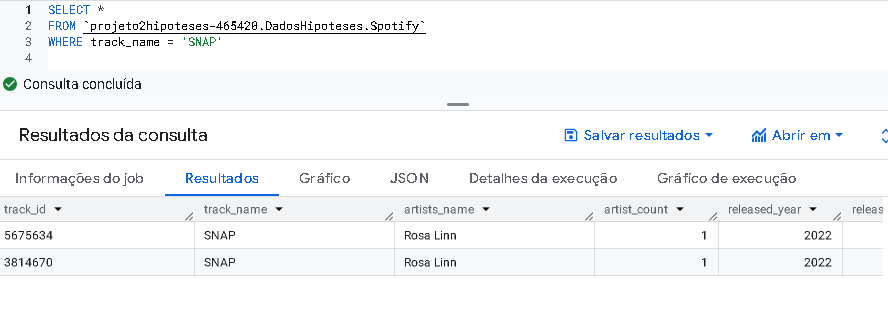
Foram encontrados **95** valores nulos na coluna **key**.

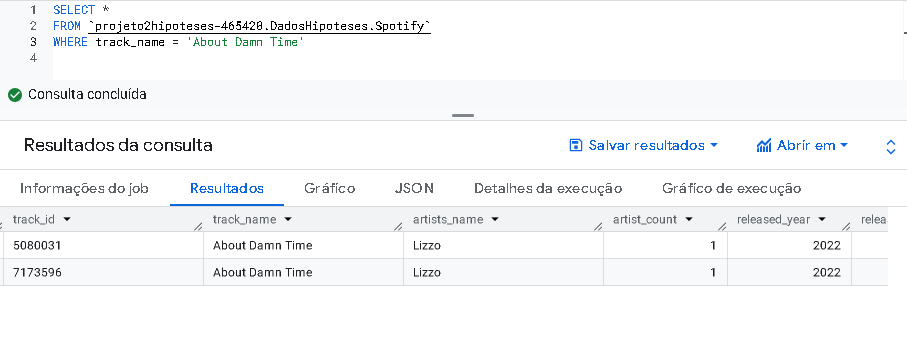


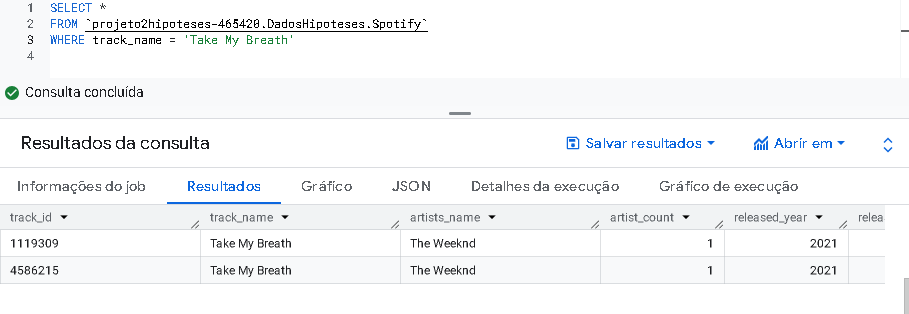
removi a coluna key e salvei uma versão atualizada para trabalhar em cima dela nas próximas etapas

5.1.3 Valores Duplicados

****

****

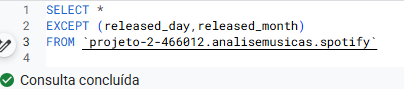




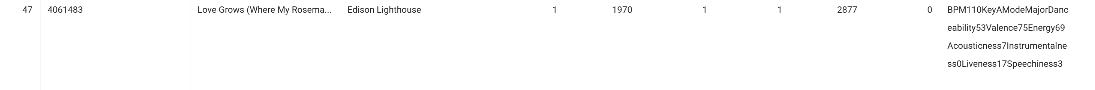
**VALORES DUPLICADOS = 4.**

5.1.4 Dados fora do escopo

**Tabela Spotify**



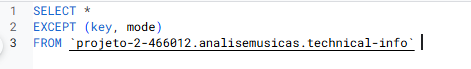
Optei por remover dia e mês de lançamento porque ao que entendi essas colunas apenas contavam a quantidades de dias ou meses em que a música foi lançada, sendo assim todos as faixas tinham numero 1 para essas colunas. Então penso que não vai fazer diferença nas análises



Achei essa linha 47 da tabela spotify estranha, no lugar da quantidade de streaming tá um compilado das informações da tabela de informações técnicas.

Essa linha eu exclui antes de subir a tabela no bigquery porque dava msg de erro pra mim então exclui ela antes.

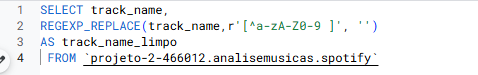
**Tabela technical-info**

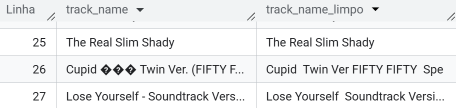


Removi key por contar com boa quantidade de nulos e tinha removido também a coluna ‘mode’, porém fui pesquisar e achei interessante deixarmos, vou deixar um comentário sobre o que é .

5.1.5 Dados discrepantes\_categorica

**TABELA SPOTIFY**





Fiz esse mesmo passo que nem você.

5.1.6 Dados discrepantes\_numericas

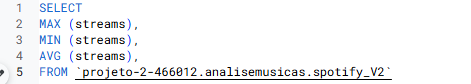
TABELA SPOTIFY



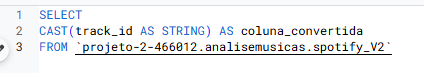
usei o comando acima para remover linhas com dados não numéricos da coluna streams

salvei a tabela resultante no drive e importei no bigquery como spotify-v2

realizei a consulta para max, min e média nessa tabela v2



5.1.7 Alterar tipo dados



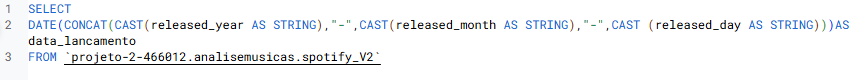
A planilha original estava com a coluna streams no formato string, porém ao tentar converter o formato para INTEGER, o comando deu erro devido à linha 47 que contém dados não numéricos. Então fiz a conversão da coluna track\_id que estava como INTEGER para STRING

Quando importei a versão 2 da planilha spotify já sem a linha 47, indiquei o formato CSV e dessa forma os tipos de dados já vieram adequados. A coluna streams já veio no formato INTEGER e não STRING como na tabela original.

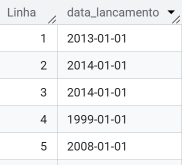
A minha tabela ja veio convertida como integer quando subi ela no bigquery.

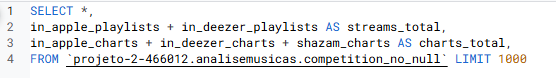
5.1.8 Novas variaveis

**TABELA SPOTIFY**

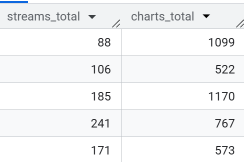


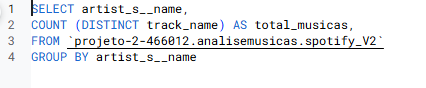
criei a variável data de lançamento unindo dia mês e ano do lançamento em uma nova coluna





criei uma coluna total de streams e outro coluna total de charts

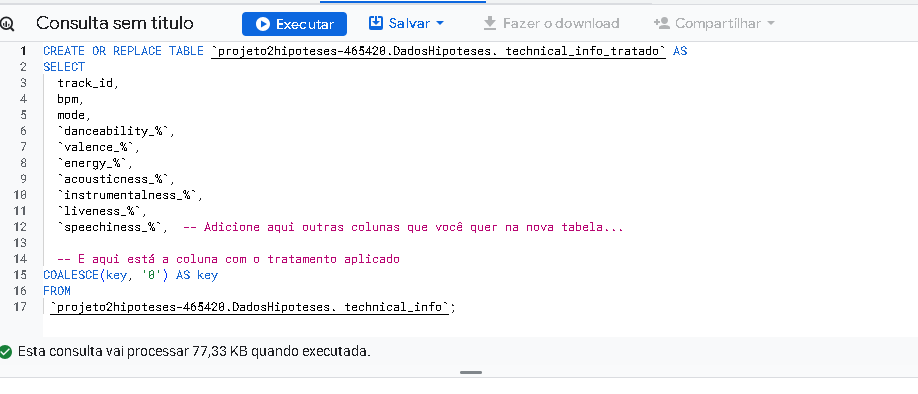


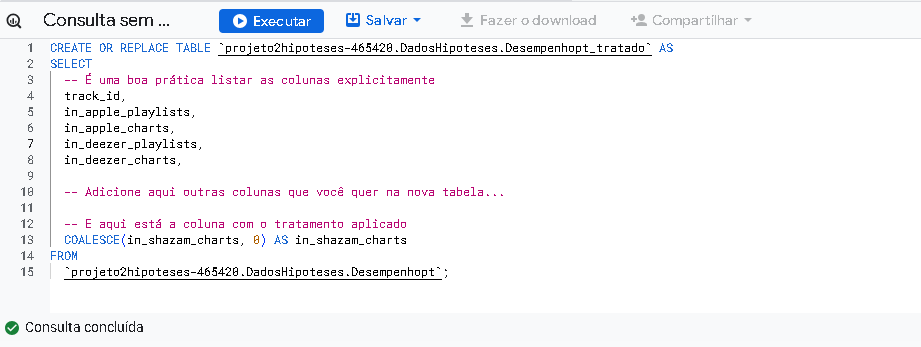


Coluna total\_musicas por artista

Tratamento Nulos

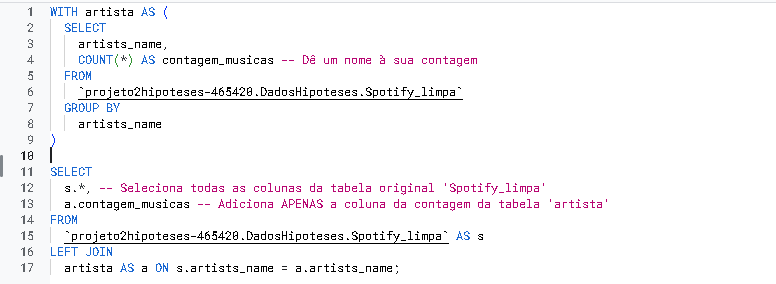
**TRATAMENTO NULOS**

****

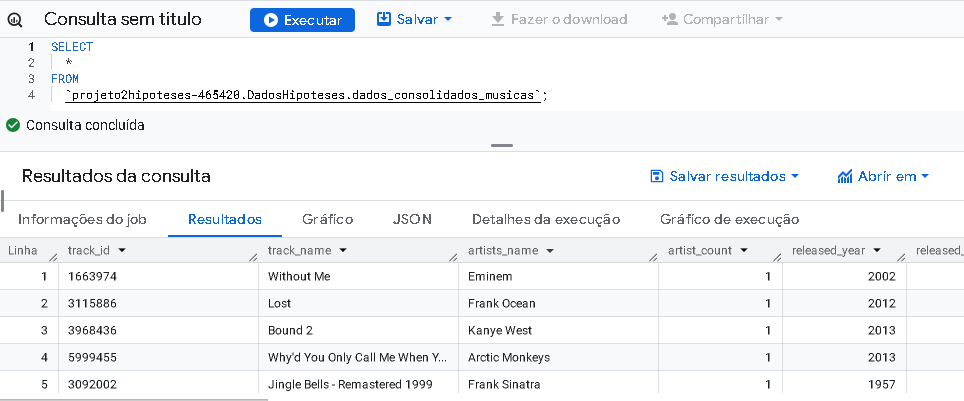
****

Foi criado uma nova tabela com os dados nulos transformados em 0. Tanto da tabela desempenho quanto da technica\_info.

Unir Tabelas



Nessa tabela o nome do artista só aparece uma vez e a quantidade de musicas que ele tem .



Criei essa tabela dados\_consolidados unindo as três tabelas.

Usei esse codigo pra unir as tabelas:

CREATE OR REPLACE TABLE `projeto2hipoteses-465420.DadosHipoteses.dados\_consolidados\_musicas` AS

SELECT

-- Colunas da tabela do Spotify

s.track\_id,

s.track\_name,

s.artists\_name,

s.artist\_count,

s.released\_year,

s.released\_month,

s.released\_day,

s.in\_spotify\_playlists,

s.in\_spotify\_charts,

s.streams,

-- Colunas da tabela de concorrência

c.in\_apple\_playlists,

c.in\_apple\_charts,

-- Colunas da tabela de informações técnicas

t.bpm,

t.key,

t.mode,

t.`danceability\_%`,

t.`valence\_%`,

t.`energy\_%`,

t.`acousticness\_%`,

t.`instrumentalness\_%`,

t.`liveness\_%`,

t.`speechiness\_%`

FROM

`projeto2hipoteses-465420.DadosHipoteses.Spotify\_limpa` AS s

INNER JOIN

`projeto2hipoteses-465420.DadosHipoteses.Desempenhopt\_tratado` AS c ON s.track\_id = c.track\_id

INNER JOIN

`projeto2hipoteses-465420.DadosHipoteses. technical\_info\_tratado` AS t ON s.track\_id = t.track\_id;

Quartiles

SELECT

a.\*,

q.quartiles\_streams,

CASE

WHEN q.quartiles\_streams = 4 THEN "alto"

WHEN q.quartiles\_streams = 3 THEN "médio"

WHEN q.quartiles\_streams = 2 THEN "baixo"

ELSE "muito baixo"

END AS classificacao\_streams

FROM `projeto-2-466012.analisemusicas.Tabela\_Final` a

LEFT JOIN (

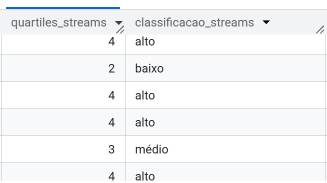
SELECT streams,

NTILE(4) OVER (ORDER BY streams) AS quartiles\_streams

FROM `projeto-2-466012.analisemusicas.Tabela\_Final`

) q

ON a.streams = q.streams



Peguei com o copilot um código que separou e classificou os quartis de todas as características que eu precisava de uma vez só:

CREATE OR REPLACE TABLE `projeto-2-466012.analisemusicas.quartis` AS

WITH base AS (

SELECT \*,

NTILE(4) OVER (ORDER BY streams) AS quartil\_streams,

NTILE(4) OVER (ORDER BY bpm) AS quartil\_bpm,

NTILE(4) OVER (ORDER BY danceability\_\_) AS quartil\_danceability,

NTILE(4) OVER (ORDER BY valence\_\_) AS quartil\_valence,

NTILE(4) OVER (ORDER BY energy\_\_) AS quartil\_energy,

NTILE(4) OVER (ORDER BY acousticness\_\_) AS quartil\_acousticness,

NTILE(4) OVER (ORDER BY instrumentalness\_\_) AS quartil\_instrumentalness,

NTILE(4) OVER (ORDER BY liveness\_\_) AS quartil\_liveness,

NTILE(4) OVER (ORDER BY speechiness\_\_) AS quartil\_speechiness

FROM `projeto-2-466012.analisemusicas.Tabela\_Final`

)

SELECT

\*,

CASE quartil\_streams WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_streams,

CASE quartil\_bpm WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_bpm,

CASE quartil\_danceability WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_danceability,

CASE quartil\_valence WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_valence,

CASE quartil\_energy WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_energy,

CASE quartil\_acousticness WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_acousticness,

CASE quartil\_instrumentalness WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_instrumentalness,

CASE quartil\_liveness WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_liveness,

CASE quartil\_speechiness WHEN 4 THEN 'alto' WHEN 3 THEN 'médio' WHEN 2 THEN 'baixo' ELSE 'muito baixo' END AS classificacao\_speechiness

FROM base

Correlação

SELECT

CORR(streams, total\_playlists) AS corr\_streams\_playlists,

CORR(streams,quartiles\_streams) AS corr\_streams\_quartiles\_streams,

CORR(streams,danceability\_\_) AS corr\_streams\_danceability\_\_,

CORR(streams,bpm) AS corr\_streams\_bpm

FROM `projeto-2-466012.analisemusicas.quartiles`

