Proyecto 2 Hipotesis

Teresa Hernández Deméneghi

Carla Yarenni Romero Vargas

**HITO 1 IDENTIFICAR Y ELIMINAR NULOS**

---Nos arroja las filas donde al menos en una columna exista un nulo---

SELECT

track\_id,

in\_apple\_playlists,

in\_apple\_charts,

in\_deezer\_playlists,

in\_deezer\_charts,

in\_shazam\_charts

FROM

`proyecto-spotify-1.spotify\_ds.track\_in\_competition`

WHERE

track\_id IS NULL

OR in\_apple\_playlists IS NULL

OR in\_apple\_charts IS NULL

OR in\_deezer\_playlists IS NULL

OR in\_deezer\_charts IS NULL

OR in\_shazam\_charts IS NULL;

---Nos arroja los valores nulos de la columna seleccionada---

SELECT

in\_shazam\_charts

FROM

`proyecto-spotify-1.spotify\_ds.track\_in\_competition`

WHERE

in\_shazam\_charts IS NULL;

--- Nos arroja la cantidad total de nulos de la columna seleccionada---

--- in\_shazam\_charts tiene 50 nulos---

SELECT

COUNT(\*)

FROM

`proyecto-spotify-1.spotify\_ds.track\_in\_competition`

WHERE

in\_shazam\_charts IS NULL;

**HITO 1 IDENTIFICAR Y ELIMINAR DUPLICADOS**

--Este comando obtiene los duplicados de dos columnas y la arroja en una nueva columna llamada "total" y al final lo que te arroja solo son los que tengan duplicados por ello el "having count"

SELECT track\_name, count(track\_name) as total

from `proyecto-spotify-1.spotify\_ds.track\_in\_spotify`

group by track\_name

having count(track\_name) > 1;

--Esto me arroja únicamente los datos que salieron repetidos anteriormente junto con todas las demas columnas de la tabla para analizar si los demas valores se repiten

SELECT \*

from `proyecto-spotify-1.spotify\_ds.track\_in\_spotify`

where (track\_name = 'SNAP' and artist\_s\_\_name = 'Rosa Linn')

or (track\_name = 'About Damn Time' and artist\_s\_\_name ='Lizzo')

or (track\_name = 'Take My Breath' and artist\_s\_\_name = 'The Weeknd')

or (track\_name = 'SPIT IN MY FACE!' and artist\_s\_\_name = 'ThxSoMch');

--Este comando me arroja una columna en orden ascendente

--Podemos observar que hay un track id "0:00" el cual coincidimos en removerlo cuando lo pasemos a una tabla limpia.

SELECT track\_id

from `proyecto-spotify-1.spotify\_ds.track\_in\_spotify`

Order by track\_id asc;

**HITO 1 DATOS DISCREPANTES**

**SELECT**

**\***

**FROM**

**`proyecto-spotify-1.spotify\_ds.track\_in\_spotify`**

**WHERE**

**track\_name LIKE '%B%'; --REGEXCP\_REPLACE reemplaza todo lo que NO sea alfanumérico y lo pone en otra columna –**

**SELECT**

**track\_name,**

**REGEXP\_REPLACE (track\_name, r'[^a-zA-Z0-9]', ' ') AS track\_name\_clean**

**FROM**

**`proyecto-spotify-1.spotify\_ds.track\_in\_spotify`; --Este comando nos regresa en una columna especial todas las canciones que tengan caracteres especiales teniendo 330 en total–**

**SELECT**

**LOWER(track\_name),**

**REGEXP\_CONTAINS(track\_name, r'[^a-zA-Z0-9 ]') AS is\_selected**

**FROM**

**`proyecto-spotify-1.spotify\_ds.track\_in\_spotify`**

**WHERE**

**REGEXP\_CONTAINS(track\_name, r'[^a-zA-Z0-9 ]') = TRUE;**

**HITO 1 MANEJAR DATOS DISCREPANTES EN VARIABLES NUMERICAS**

**--- #Verificar, hubo un error con la variable streams, por un registro**

**SELECT MAX(released\_day) as max\_day, MIN(released\_day) as min\_day, ROUND(AVG(released\_day),2) as avg\_day,**

**MAX(released\_month) as max\_month, MIN(released\_month) as min\_month, ROUND(AVG(released\_month),2) as avg\_month,**

**MAX(released\_year) as max\_year, MIN(released\_year) as min\_year, ROUND(AVG(released\_year),2) as avg\_year,**

**MAX(in\_spotify\_charts) as max\_chart, MIN(in\_spotify\_charts) as min\_chart, ROUND(AVG(in\_spotify\_charts),2) as avg\_chart,**

**MAX(in\_spotify\_playlists) as max\_playlists, MIN(in\_spotify\_playlists) as min\_playlists, ROUND(AVG(in\_spotify\_playlists),2) as avg\_playlists,**

**-- MAX(streams) as max\_streams, MIN(streams) as min\_streams, ROUND(AVG(streams),2) as avg\_streams *--Error por tipo string***

**HITO 1 CREAR NUEVAS VARIABLES**

**SELECT CAST(track\_id AS INT64) AS track\_id\_clean,**

**REGEXP\_REPLACE(track\_name, r'[^a-zA-Z0-9 ]' , ' ') AS track\_name\_clean,**

**REGEXP\_REPLACE(artist\_s\_\_name, r'[^a-zA-Z0-9 ]' , ' ') AS artist\_s\_\_name\_clean,**

**released\_year, released\_month, CAST(CONCAT(released\_year ,'-', released\_month, '-', released\_day) AS date) as release\_date,**

**in\_spotify\_charts, in\_spotify\_playlists, in\_spotify\_charts + in\_spotify\_playlists AS in\_spotify\_total,**

**CAST(streams AS INT64) AS streams\_clean**

**FROM `proyecto-spotify.spotify\_ds.track\_in\_spotify`**

**HITO 1 LIMPIEZA DE DATOS**

**SELECT**

**CAST(track\_id AS INT64) AS track\_id\_clean,**

**REGEXP\_REPLACE (track\_name, r'[^a-zA-Z0-9]', ' ') AS track\_name\_clean,**

**REGEXP\_REPLACE (artist\_s\_\_name, r'[^a-zA-Z0-9]', ' ') AS artist\_s\_\_name\_clean,**

**--Utilizando CAST y CONCAT obtuve una nueva columna con la fecha de lanzamiento con tipo de dato DATE--**

**released\_year, released\_month,**

**CAST(CONCAT(released\_year,-released\_month,-released\_day)AS DATE) AS released\_date,**

**-- utilizando un operador matemático sume ambas columnas para obtener las playlists o charts en total--**

**in\_spotify\_charts,in\_spotify\_playlists, in\_spotify\_charts + in\_spotify\_playlists AS in\_spotify\_total,**

**CAST(streams AS INT64) AS streams\_clean**

**FROM**

**`proyecto-spotify-1.spotify\_ds.track\_in\_spotify`**

**WHERE--Aquí le decimos que NO nos traiga nada que tenga B para eliminar el valor erróneo en streams y el de track\_id**

**streams NOT LIKE '%B%'**

**AND track\_id NOT LIKE '%:%'**

**AND track\_id != '5675634'**

**AND track\_id != '3814670'**

**AND track\_id != '7173596'**

**AND track\_id != '5080031'**

**AND track\_id != '1119309'**

**AND track\_id != '4586215'**

**AND track\_id != '4967469'**

**AND track\_id != '8173823'**

**HITO 1 UNIÓN DE TABLAS, OBTENCIÓN DE CUARTILES Y CATEGORÍAS**

**WITH Quartiles AS(**

**SELECT track\_id\_clean, streams\_clean, bpm, in\_total\_charts, in\_total\_playlists,danceability\_\_,valence\_\_,energy\_\_,acousticness\_\_, instrumentalness\_\_, liveness\_\_, speechiness\_\_,**

**NTILE(4) OVER (ORDER BY streams\_clean) AS quartile\_streams,**

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

**NTILE(4) OVER (ORDER BY in\_total\_charts) AS quartile\_charts,**

**NTILE(4) OVER (ORDER BY in\_total\_playlists) AS quartile\_playlists,**

**CASE**

**WHEN danceability\_\_ <= 25 THEN 1**

**WHEN danceability\_\_ <= 50 THEN 2**

**WHEN danceability\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_danceability,**

**CASE**

**WHEN valence\_\_ <= 25 THEN 1**

**WHEN valence\_\_ <= 50 THEN 2**

**WHEN valence\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_valence,**

**CASE**

**WHEN energy\_\_ <= 25 THEN 1**

**WHEN energy\_\_ <= 50 THEN 2**

**WHEN energy\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_energy,**

**CASE**

**WHEN acousticness\_\_ <= 25 THEN 1**

**WHEN acousticness\_\_ <= 50 THEN 2**

**WHEN acousticness\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_acousticness,**

**CASE**

**WHEN instrumentalness\_\_ <= 25 THEN 1**

**WHEN instrumentalness\_\_ <= 50 THEN 2**

**WHEN instrumentalness\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_instrumentalness,**

**CASE**

**WHEN liveness\_\_ <= 25 THEN 1**

**WHEN liveness\_\_ <= 50 THEN 2**

**WHEN liveness\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_liveness,**

**CASE**

**WHEN speechiness\_\_ <= 25 THEN 1**

**WHEN speechiness\_\_ <= 50 THEN 2**

**WHEN speechiness\_\_ <= 75 THEN 3**

**ELSE 4**

**END as quartile\_speechiness,**

**FROM**

**`proyecto-spotify-1.spotify\_ds.all\_data`**

**)**

**SELECT**

**a.\*,**

**Quartiles.quartile\_streams,**

**Quartiles.quartile\_bpm,**

**Quartiles.quartile\_charts,**

**Quartiles.quartile\_playlists,**

**Quartiles.quartile\_danceability,**

**Quartiles.quartile\_valence,**

**Quartiles.quartile\_energy,**

**Quartiles.quartile\_acousticness,**

**Quartiles.quartile\_instrumentalness,**

**Quartiles.quartile\_liveness,**

**Quartiles.quartile\_speechiness,**

**IF(Quartiles.quartile\_streams = 4, 'Alto','Bajo') AS streams\_category,**

**IF(Quartiles.quartile\_bpm = 4, 'Alto','Bajo') AS bpm\_category,**

**IF(Quartiles.quartile\_charts = 4, 'Alto','Bajo') AS charts\_category,**

**IF(Quartiles.quartile\_playlists = 4, 'Alto','Bajo') AS playlists\_category,**

**IF(Quartiles.quartile\_danceability = 4, 'Alto','Bajo') AS danceability\_category,**

**IF(Quartiles.quartile\_valence = 4, 'Alto','Bajo') AS valence\_category,**

**IF(Quartiles.quartile\_energy = 4, 'Alto','Bajo') AS energy\_category,**

**IF(Quartiles.quartile\_acousticness = 4, 'Alto','Bajo') AS acousticness\_category,**

**IF(Quartiles.quartile\_instrumentalness = 4, 'Alto','Bajo') AS instrumentalness\_category,**

**IF(Quartiles.quartile\_liveness = 4, 'Alto','Bajo') AS liveness\_category,**

**IF(Quartiles.quartile\_speechiness = 4, 'Alto','Bajo') AS speechiness\_category,**

**FROM `proyecto-spotify-1.spotify\_ds.all\_data` a**

**LEFT JOIN Quartiles**

**ON a.track\_id\_clean= Quartiles.track\_id\_clean;**

**HITO 1 CORRELACIÓN**

**SELECT**

**CORR(acousticness\_\_,streams\_clean) AS corraco\_streams,**

**CORR(danceability\_\_, streams\_clean) AS corrdan\_streams,**

**CORR(liveness\_\_,streams\_clean) AS corrliv\_streams,**

**CORR(energy\_\_, streams\_clean) AS correner\_streams,**

**CORR(valence\_\_, streams\_clean) AS corrval\_streams,**

**CORR(instrumentalness\_\_, streams\_clean) AS corrinst\_streams,**

**CORR(speechiness\_\_, streams\_clean) AS corrspeech\_streams,**

**from `proyecto-spotify-1.spotify\_ds.all\_data`**