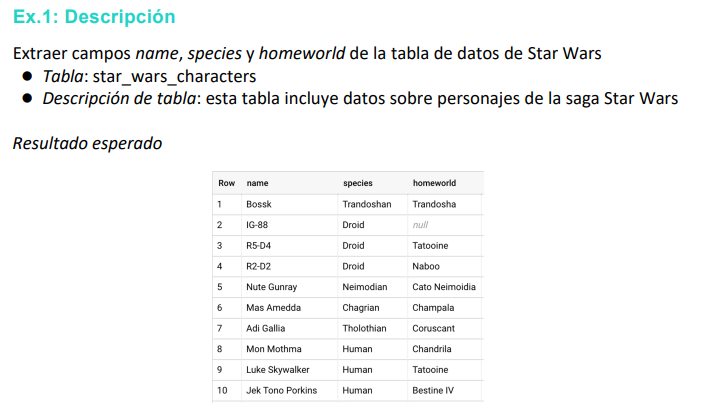
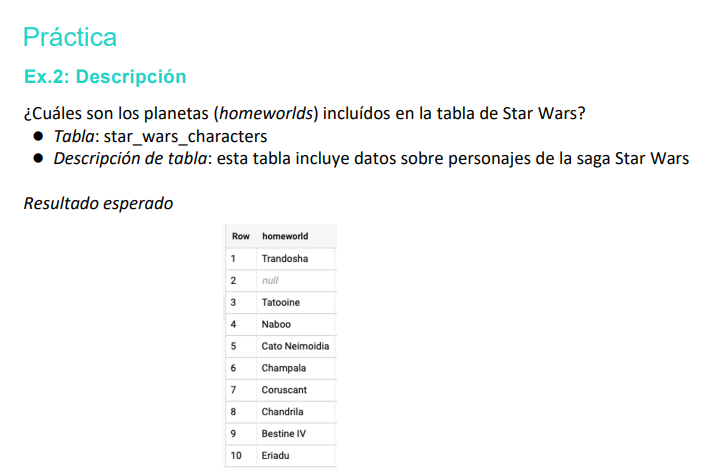
Ejercicios prácticos en clase



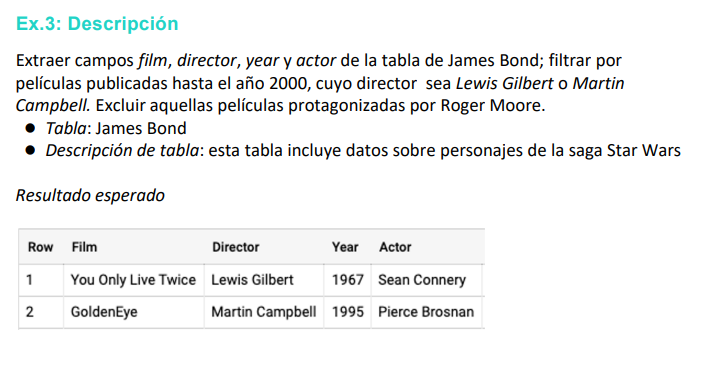
SELECT name, species, homeworld

from star\_wars\_characters\_2



SELECT DISTINCT homeworld

from star\_wars\_characters\_2

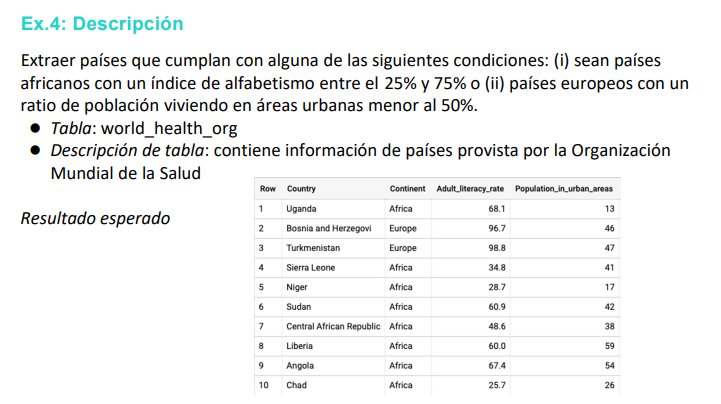


SELECT film, director, year, actor

from jamesbond

WHERE year <= 2000 and director IN ('Lewis Gilbert', 'Martin

Campbell') AND actor <> 'Roger Moore'



SELECT country, continent, adult\_literacy\_rate, population\_in\_urban\_areas

FROM world\_health\_org

WHERE (continent= 'Africa' AND adult\_literacy\_rate BETWEEN 25 and 75) OR (continent= 'Europe' AND population\_in\_urban\_areas < 50)



select country, continent, gross\_income\_per\_capita

FROM world\_health\_org

WHERE continent= 'Africa'

order by gross\_income\_per\_capita DESC

LIMIT 5



SELECT movie\_title, director\_me, imdb\_score

FROM imdb\_movies

WHERE title\_year >= 1980 and country is not 'USA'

order by imdb\_score DESC

limit 10

**Otros ejercicios para practicar**

1.- De la tabla videogames\_games, ordenar por año y mostrar resultados del nombre del juego que fue publicado por Nintendo en el año 1995 en adelante, que no se llame “Donkey Konga 2”, que vendiera entre cero y 0,32 en Europa, entre cero y 0,31 en America, y entre cero y 0,3 en otros territorios.

**select name, publisher, year, eu\_sales, na\_sales, other\_sales**

**FROM videogames\_games**

**where publisher= 'Nintendo' and year> 1995 and name <> 'Donkey Konga 2' and eu\_sales BETWEEN 0 and 0.32 and na\_sales BETWEEN 0 and 0.31 and other\_sales BETWEEN 0 and 0.3**

**ORDER BY year**

2.- Tabla: star\_wars\_characters. Nombre de los personajes de Star Wars que tengan los ojos distintos a rojo y que sean de los mundos Chandrila, Stewjon o Tatooine. Añadir un nuevo campo “tamaño” que nos diga si es “grande” cuando la altura  sea mayor o igual a 200 y “pequeño” si es menor de 200. Por último ordenar por altura de menor a mayor.

**SELECT name, eye\_color, homeworld, height,**

**CASE**

**WHEN height >=200 THEN 'Grande'**

**else 'Pequeño'**

**END AS Tamaño**

**from star\_wars\_characters\_2**

**where eye\_color <> 'red' and homeworld in ('Chandrila', 'Stewjon', 'Tatooine')**

3.- De la tabla fortune, queremos ver sólo las 100 primeras compañías según el rango, sin incluir a Citigroup, viendo sólo las compañías dentro del sector financiero; industria diversified financials y crearemos un nuevo campo de rentabilidad en el que divida a las compañías entre rentables y no rentables si sus beneficios son mayores o menores a 1000.

**select company, rank, sector, industry, profits,**

**CASE**

**WHEn profits >=1000 then 'Rentables'**

**else 'no Rentables'**

**end as Rentabilidad**

**from fortune**

**where company <> 'Citigroup' and sector= 'Financials' and industry= 'Diversified Financials'**

**LIMIT 10**

4.- En la tabla gobierno\_paro, filtrar por el mes de ‘Enero de 2018’ y el paro de hombres y mujeres de entre 25 y 45 años entre 1100 y 20 y ordenar los municipios con esos filtros de mayor paro registrado (del total) a menor. Por último, añadir una nueva columna llamada ‘Comparación de paro’, agrupando el número total de parados en: mucho si es mayor que 1000, medio si es mayor que 500 y poco si es por debajo o igual a 500.

**select provincia, mes, comunidad\_autonoma, paro\_hombre\_edad\_25\_\_45, paro\_mujer\_edad\_25\_\_45, municipio, total\_paro\_registrado,**

**CASE**

**when total\_paro\_registrado>=1000 then 'mucho'**

**when total\_paro\_registrado>500 THEN 'medio'**

**ELSE 'poco'**

**END AS 'Comparaciòn de paro'**

**from gobierno\_paro**

**WHERE mes= 'Enero de 2018' and paro\_hombre\_edad\_25\_\_45 BETWEEN 20 and 1100 AND paro\_mujer\_edad\_25\_\_45 BETWEEN 20 and 1100**

**ORDER BY total\_paro\_registrado desc**

5.- En la tabla star\_wars\_characters, buscar el personaje que sea de un planeta que comience por la letra T y que no termine por la letra d, que mida más de 200 y que no sea un droide. Pista: el caracter comodin es %.

**SELECT name, homeworld, species, height**

**FROM star\_wars\_characters\_2**

**WHERE homeworld like 'T%' and homeworld not like '%d' and height>200 and species <> 'Droid'**

6.- En la tabla videogames\_games, queremos ver los nombres de los juegos de platform\_code ‘DS’ y años ordenados por año, con las ventas en las tres regiones principales (NA, EU y JP).

**SELECT name, platform\_code, year, na\_sales, eu\_sales, jp\_sales,**

**CASE**

**WHEN na\_sales> eu\_sales and na\_sales> jp\_sales THEN 'NA'**

**WHEN eu\_sales> na\_sales and eu\_sales> jp\_sales THEN 'EU'**

**WHEN jp\_sales>eu\_sales and jp\_sales>na\_sales THEN 'JP'**

**end as best\_selling\_region**

**from videogames\_games**

**WHERE platform\_code= 'DS'**

**order by year**

7.-Cuáles son las películas en USA, ordenadas de forma descendente por duración en la tabla de imdb\_movies. añadir el nuevo campo ‘’SON MUY LARGAS’’ a las que duren  de 120 minutos a 200  minutos y ‘’son cortas’’ las que duren menos 120.

**select movie\_title, duration, country,**

**CASE**

**when duration BETWEEN 120 and 200 then 'muy largas'**

**when duration<120 then 'son cortas'**

**end as 'Tipo'**

**FROM imdb\_movies**

**where country= 'USA'**

**order by duration desc**

8.- De la tabla gobierno\_paro, cuál es la cifra de mujeres de 45 años que estaban en paro, donde la cifra de parados sin empleo anterior era cero, durante el mes de marzo de 2018, ordenado por municipios por órden alfabético descendente.

Rta: La cifra era 21.

**SELECT paro\_mujer\_edad\_\_\_45, paro\_sin\_empleo\_anterior, mes, municipio**

**from gobierno\_paro**

**WHERE paro\_sin\_empleo\_anterior = 0 and mes= 'Marzo de 2018'**

**ORDER by municipio DESC**

9.- Las 50 películas a color de USA, desde el año 2000 a 2010 con un campo nuevo llamado 'Calificación según Facebook' donde se evidencien tres variables: si tiene menos de 5.000 likes es 'Regular'; si tiene entre 5.000 y 10.000 likes es 'Buena' y si tiene más de 10.000 likes es 'Muy Buena'.

**SELECT color, country, title\_year, movie\_facebook\_likes,**

**CASE**

**when movie\_facebook\_likes > 10000 then 'muy buena'**

**when movie\_facebook\_likes BETWEEN 5000 and 10000 then 'buena'**

**else 'regular'**

**end as 'Calificación segun FB'**

**from imdb\_movies**

**WHERE color= 'Color' and title\_year BETWEEN 2000 and 2010**

**order by title\_year DESC**

**Limit 50**

**OTROS EJERCICIO****S PROPUESTOS**

1. De la tabla de videojuegos(games), saca los resultados de ventas de los juegos más vendidos (según el ranking) en cada una de las regiones en la década de los 90 y 2000. Además crea una nueva columna en la que el juego que llegue o supere 1 en alguna región se llame <<Top>>, los demás <<meh>>, la columna se llama ¿qué tal le fue? Mostrar los 20 primeros resultados. Solo si el nombre del juego no contiene “brain”.

**select rank, name, year, na\_sales, eu\_sales, jp\_sales,other\_sales,**

**CASE**

**WHEN eu\_sales>= 1 or jp\_sales>= 1 or na\_sales>= 1 or other\_sales>= 1 THEN 'Top'**

**else 'Meh'**

**end as 'Que tal le fue?'**

**from videogames\_games**

**where name != 'Brain' AND year BETWEEN 1990 and 2000 --and name not like '%Brain' or name not LIKE 'Brain%'**

**order by rank DESC**

**Limit 20**

1. De la tabla rolling\_top\_albums\_1, los álbumes entre el 1980 y 2000, saber el ranking y el nombre del álbum, crear un nuevo campo llamado "TOP" en el que el ranking que sea menor de 10 sea "TOP 10", entre 11 y 50 sea "TOP 50" y el resto sea "TOP 100" y ordenar según el ranking

select album, release\_year, albumid\_rank,

CASE

WHEN albumid\_rank> 50 then 'Top 100'

WHEN albumid\_rank BETWEEN 11 and 50 then 'Top 50'

ELSE 'Top 10'

end AS 'Top'

FROM rolling\_top\_albums\_1

WHERE release\_year BETWEEN 1980 and 2000

order by albumid\_rank desc

1. De la tabla star\_wars\_characters\_2. Nombre de los personajes de Star Wars no humanos que midan menos de 200 y pesen más de 50, nacidos entre el año 20 y 60 y ordenados de más viejo a más joven. Añadir un nuevo campo “Rol” que nos diga si es un “Lord Sith” si la especie es Gungan, si no, que diga que es un “Parguela”.

**SELECT name, height, mass, birth\_year, species,**

**CASE**

**WHEN species= 'Gungan' THEN 'Lord Sith'**

**ELSE 'Parguela'**

**end AS 'Rol'**

**from star\_wars\_characters\_2**

**WHERE species != 'Human' and height< 200 and mass> 50 and birth\_year BETWEEN 20 and 60**

**ORDER by birth\_year desc**

1. De la tabla rolling\_top\_albums\_1 asegurate de que no tenemos en cuenta ninguna aberración grindcore ni del sello A&M. Vamos a centrarnos solo en álbumes de más de 35' ya que menos es lo que dura el descanso del desayuno. Muestra una tabla donde la primera columna se llame 'NUESTRA CRÍTICA' y se valore los discos según su ranking de popularidad; si están en el top 50 o no y si su puntuación es mayor que 4 o no, dando como resultados:

'MASTERPIECE RESPETADA' (pop - punt alta)

'PEPINO CAYENDO EN EL OLVIDO' (no pop - puntuación alta)

'ESTUVO BIEN, PERO PORQUE IBA BORRACHO' (no pop - puntuación baja)

'SUENA, PERO SOBREVALORADO' (pop - puntuación baja)

En esta misma columna deja claro que 'NUESTRO TOP1' es el disco de Gojira, From Mars to Sirius y que si el disco salió antes del 80 y tiene más de un 4 de puntuación es un 'DECRÉPITO DISCO QUE NO SALE NI CON AGUA HIRVIENDO'. Tras esta columna, muestra también el nombre del disco, el grupo y el género.

**select CASE**

**WHEN artist= 'Gojira' then 'Top 1'**

**when release\_year < 1980 and rating >= 4 then 'DECRÉPITO DISCO QUE NO SALE NI CON AGUA HIRVIENDO'**

**when albumid\_rank >= 50 and rating >= 4 then 'MASTERPIECE RESPETADA'**

**when albumid\_rank >= 50 and rating < 4 THEN 'SUENA, PERO SOBREVALORADO'**

**when albumid\_rank < 50 and rating >= 4 then 'PEPINO CAYENDO EN EL OLVIDO'**

**when albumid\_rank < 50 and rating < 4 THEN 'ESTUVO BIEN, PERO PORQUE IBA BORRACHO'**

**END AS 'Nuestra Crìtica' ,**

**artist, album, sub\_metal\_genre**

**FROM rolling\_top\_albums\_1**

**where sub\_metal\_genre != 'Alternative Metal' and label <> 'A&M' and duration> 35**

**order by albumid\_rank DESC**

1. En la tabla 'videogames\_consoles' traduce al español todos los campos, luego crea un campo llamado 'Rentabilidad' donde si es mayor a 100 millones de unidades de consolas vendidas será 'Alta Rentabilidad' si es mayor a 50 millones de unidades consolas vendidas será 'Rentable' y si es menor a 50 millones de unidades de consolas vendidas será 'No rentable' luego crea otro campo llamado 'Estado' donde indican si las consolas están en 'Activa' o 'Descontinuada', luego crea otro campo llamado 'Compañía' donde indican si una consola pertenece a la compañía 'Playstation','Xbox','Nintendo' o 'Otras compañías' sin no pertenecen a playstation, xbox o nintendo, luego muestra las 15 consolas más vendidas.

**select console\_name, units\_sold\_million,**

**CASE WHEN units\_sold\_million> 100 then 'Alta Rentabilidad'**

**WHEN units\_sold\_million BETWEEN 50 and 100 THEN 'Rentable'**

**ELSE 'No rentable'**

**end as 'Rentabilidad',**

**discontinued,**

**case when discontinued= '' then 'Activa'**

**else 'Inactiva'**

**end as 'Estado', platform\_code**

**FROM videogames\_consoles**

**order by units\_sold\_million DESC**

**LIMIT 15**

1. En la tabla imdb\_movies, ¿cómo de alto, medio o bajo ha sido el presupuesto

que se invirtió en el top 10 de las películas con más likes en Facebook?

Considerando que presupuesto alto son aquellas en las que se invirtió más

de 99000000, que las de presupuesto medio invirtieron entre 99000000 y

10000000 y que las de presupuesto bajo invirtieron menos de 10000000.

**select movie\_title, gender, movie\_facebook\_likes, budget,**

**CASE WHEN budget>99000000 then 'Presupuesto Alto'**

**WHEN budget BETWEEN 10000000 and 99000000 THEN 'Presupuesto medio'**

**ELSE 'Presupuesto bajo'**

**end as 'Presupuesto'**

**from imdb\_movies**

**order by movie\_facebook\_likes DESC**

**LIMIT 10**

**Ejercicio propuesto**

Cuando se habla del nacimiento del cine se toma como referencia la fecha del 28 de diciembre de 1895, en la que se proyectaron al público las primeras películas realizadas por los hermanos Auguste y Louis Lumière, en la memorable sesión realizada en el Salón Indio del Gran Café de París.

De acuerdo a lo anterior realiza una consulta para saber cuántos likes recibió en redes sociales (donde participaron más de mil usuarios) un director norteamericano dedicado a dirigir películas para hacer reír al público en época de navidad.

¡Si has llegado a la respuesta, ubica enseguida de la misma una frase que diga: **lo encontre!!**

Soluciòn:

**select movie\_title, director\_me,director\_facebook\_likes,**

**case WHEN director\_facebook\_likes= 54 then 'Lo Encontre!!'**

**end as 'Busqueda',**

**gender,num\_voted\_users,country**

**from imdb\_movies**

**WHERE num\_voted\_users> 1000 and country= 'USA' and gender like 'C%' and gender like '%y'**

**order by director\_facebook\_likes**

**Ejercicios propuestos por la clase**

**HECTOR**

TABLA: world\_health\_org

Queremos averiguar cómo ha sido el crecimiento de los 5 países más poblados en África, para ello necesitamos saber si su crecimiento ha sido alto, medio o ha decrecido según su tasa de crecimiento anual.

De manera que, si es mayor que 2 es un crecimiento ALTO, si está entre 0 y 2 es un crecimiento MEDIO y si es negativo, su población HA DECRECIDO. Ordena el resultado según el país más poblado al menos

**SELECT country, continent, population\_in\_thousands, population\_annual\_growth\_rate,**

**CASE when population\_annual\_growth\_rate>2 then 'Alto'**

**WHEN population\_annual\_growth\_rate BETWEEN 0 and 2 then 'Medio'**

**ELSE 'Decreciò'**

**end as Crecimiento**

**from world\_health\_org**

**where continent= 'Africa'**

**order by population\_in\_thousands DESC**

**limit 5**

**SERGIO**

TABLA: imdb\_movies

Encuentra las 25 películas del género 'Comedy' y producidas en Estados Unidos con la puntuación IMDB más alta. Muestra el título de la película, el presupuesto y el nombre del actor principal (Actor 1).

Además, crea una columna llamada 'Opinion' en la que si los likes de la película en facebook son mayores de 15000 y la duración es mayor de 100 minutos, nos aparezca 'Aclamada pero larga'; si los likes de la película son mayores de 15000 pero la duración es menor de 100 minutos, nos aparezca 'Ligerita y entretenida'; si los likes son menores de 15000 pero la duración es menor de 100 minutos, nos aparezca 'Corta, pero aburrida'; y finalmente si los likes son menores de 15000 pero la duración es mayor de 100 minutos, nos aparecerá 'Ahorrate el sufrimiento'.

Descarta también las películas que tengan un presupuesto NULL.

**SELECT**

**movie\_title AS 'Película',**

**budget AS Presupuesto,**

**actor\_1\_me AS Actor,**

**CASE**

**WHEN movie\_facebook\_likes > 15000 AND duration > 100 THEN 'Aclamada pero larga'**

**WHEN movie\_facebook\_likes > 15000 AND duration <= 100 THEN 'Ligerita pero entretenida'**

**WHEN movie\_facebook\_likes <= 15000 AND duration <= 100 THEN 'Corta, pero aburrida'**

**WHEN movie\_facebook\_likes <= 15000 AND duration > 100 THEN 'Ahorrate el sufrimiento'**

**END AS Opinion**

**FROM imdb\_movies**

**where**

**gender LIKE '%Comedy%'**

**AND country = 'USA'**

**AND budget != 'NULL'**

**ORDER BY imdb\_score desc**

**LIMIT 25**

**MARINA**

TABLA: imdb\_movies

Indica las películas en japonés cuyo director empiece por la letra ‘H’. Además crea un campo nuevo que se llame ‘Actor favorito’ y ordenarlo según el número de likes del 1º actor en Facebook y  sea: ‘Sí quiero’ sí tiene 500 o más , ‘Si insistes’ si tiene entre 500 y 100, y ‘No gracias’ si tiene menos de 100.

**TOMAS**

TABLA: bitcoin\_daily\_rates\_formatdate

* Filtrar la información desde el año 2017 (inclusive) hasta el 2018 (inclusive).

* Crear un campo llamado “Precio” que determine:
  + Si el BTC cerró con un precio superior al de apertura, “Subió”.
  + Si el BTC cerró con un precio inferior al de apertura, “Bajó”.

* Crear un campo llamado “Sensación/Volumen” que indique:

* Si el cierre (close) es menor a 7000 y el volumen en BTC es menor a 10000, “Miedo - Volumen bajo”.
* Si el cierre es menor a 7000 y el volumen en BTC está entre 10000 y 20000, “Miedo - Volumen medio”.
* Si el cierre es menor a 7000 y el volumen en BTC es mayor a 20000, “Miedo - Volumen alto”.

* Si el cierre está entre 7000 y 12000 y el volumen en BTC es menor a 10000, “Incertidumbre  - Volumen bajo”.
* Si el cierre está entre 7000 y 12000 y el volumen en BTC está entre 10000 y 20000, “Incertidumbre  - Volumen medio”.
* Si el cierre está entre 7000 y 12000 y el volumen en BTC es mayor a 20000, “Incertidumbre  - Volumen alto”.

* Si el cierre es mayor a 12000 y el volumen en BTC es menor a 10000, “Codicia - Volumen bajo”.
* Si el cierre es mayor a 12000 y el volumen en BTC está entre 10000 y 20000, “Codicia - Volumen medio”.
* Si el cierre es mayor a 12000 y el volumen en BTC es mayor a 20000, “Codicia - Volumen alto”.

* Asegúrate de que se visualicen todos los campos mencionados, sumado además el de volumen en USD.
* Ordenar la información por fecha

**MIRA**

TABLA: rolling\_top\_albums\_1

Queremos crear un nuevo campo de canciones viejas y nuevas donde: canciones antes o igual del 1950 son viejas y después del 1950 son nuevas.

Buscar los artista que tienen letras A, M y B al principio del nombre pero que no se acaban a la letra T y ordenar el Álbum Rank de alto a bajo

**FRANMA**

TABLA: videogames\_console

Indica a qué compañía pertenece cada consola de microsoft, nintendo y sony  y además clasifica según las ventas la acogida de las consolas de nintendo, xbox y playstation . siendo que:

* Si vendieron más de 100 millones fue excelente
* Si vendieron entre 50 y 100 millones fue buena
* Si vendieron menos de 20 fue lamentable
* En el resto de casos aceptable

**SELECT console\_name,**

**CASE WHEN console\_name like '%Gam%' OR console\_name like '%Nint%' then 'Nintendo'**

**When console\_name like '%Xbox%' then 'Xbox'**

**WHEN console\_name LIKE '%Pla%' then 'Playstation'**

**ELSE 'Otras compañias'**

**end as compañia,**

**CASE WHEN units\_sold\_million>100 then 'Excelente'**

**when units\_sold\_million BETWEEN 50 and 100 then 'Buena'**

**when units\_sold\_million <20 then 'lamentable'**

**else 'Aceptable'**

**end as acogida**

**from videogames\_consoles**

**where compañia<> 'Otras compañias'**

**RODRIGO**

TABLA: rolling\_top\_albums\_1

Queremos traducir todos los campos al español, una vez traducidos queremos ordenar de forma alfabetica descendente el spotify albums, ademas crearon 5 campos, un campo que se llamara “segundos” donde entre 40 y 50 seg sera “mas segundos” entre 20 y 40 “medio” y menos de 20 “poco segundo”, otro campo “los 4 grandes”donde si el artista es megadeth dara “molon” si es anthrax” super molon” si es slayer sera “maquinas” si es metallica sera “ el primero”, otro campo” reyes del nu metal “donde si los artistas son Korn, Slipknot y System of a Down sera “idolos”, otro campo “tiempo total” donde si la duracion dura mas de tres cuartos de hora sera “currado” si es entre tres cuartos de hora y 25 minutos sera “medio currao” y todo lo q dure menos tiempo sera “ baratija”, otro campo “no info” donde si en spotify album no tiene referencia alguna sera “nada en spoti”. Concluir diciendo que no

quiero ver en ningun campo al artista nightwish, el album Psalm 69: The Way to Succeed and the Way to Suck Eggs, el año de publicacion 1997 y de etiqueta music for nations. Enseñame los 10 primeros de todo esto.

SELECT albumid\_rank as "ranking del album", artist as "artista", album as "album", release\_year as "año de lanzamiento", spotify\_album as "album spotify", description as "descripcion", wiki as "wikipedia", duration as "duracion", minutes as "minutos", seconds as "segundos", total\_seconds as "total de segundos", label as "etiqueta", sub\_metal\_genre as "genero", rating as "puntuacion", rolling\_stone\_rating "puntuacion de rolling stones",

case

when seconds BETWEEN 40 and 50 then "mas segundos"

when seconds BETWEEN 20 and 40 then "medio"

when seconds < 20 then "poco segundo"

end as "segundos",

case

when artist = "Megadeth" then "molon"

when artist = "Anthrax" then "super molon"

when artist = "Slayer" then "maquinas"

when artist = "Metallica" then "el primero"

end as "los 4 grandes",

case

when artist in ("Korn", "Slipknot","System of a Down") then "idolos"

end as "reyes del nu metal",

case

when duration > "45:00:00" then "currado"

when duration BETWEEN "25:00:00" and "45:00:00" then "medio currao"

when duration < "25:00:00" then "baratija"

end as "tiempo total",

case

when spotify\_album not like "%\_%" then "nada en spoti"

end as "no info"

--cogeme los datos de la tabla de

from rolling\_top\_albums\_1

where artist <> "Nightwish"

and album <> "Psalm 69: The Way to Succeed and the Way to Suck Eggs"

and release\_year <> 1997

and label <> "Music for Nations"

--ordename alfabeticamente

order by spotify\_album DESC

--limitame a

limit 10

**GABI**

TABLA : videogames\_games

12 juegos de playstation entre el 1998 y el 2000 publicados por Sony.

Señala según sus ventas en el mercado norte americano si las ventas fueron buenas o malas, si son mayores a 1 millón serán buenas, si no, serán malas.

Ordena los juegos de más viejos a más nuevos.

Ejercicio mìo

Select movie\_title, director\_me,director\_facebook\_likes,

case WHEN director\_facebook\_likes= 54 then 'Lo Encontre!!'

ELSE ' '

end as 'Busqueda',

gender,num\_voted\_users,country

from imdb\_movies

WHERE num\_voted\_users> 1000

and country= 'USA'

and gender like '%Comedy%'

and director\_facebook\_likes IS not null

order by director\_facebook\_likes

Ejercicios clase 26/01/23 Jueves

Con agregaciones having, group by, sum, average.

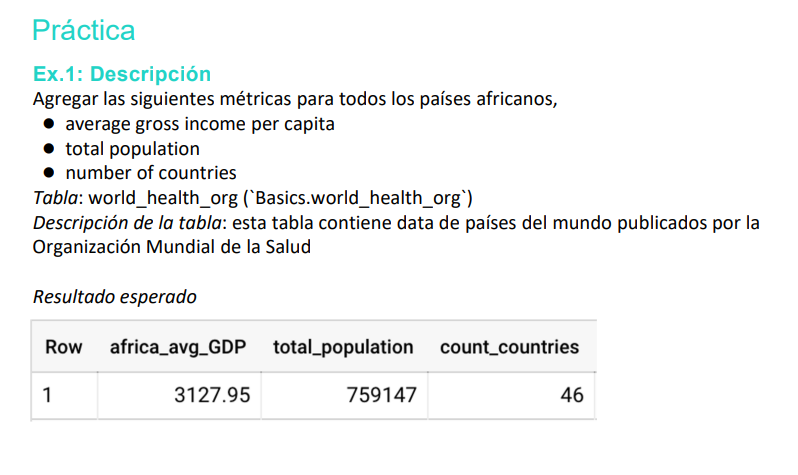


SELECT name, mass

FROM star\_wars\_characters\_2

ORDER BY mass desc

limit 10



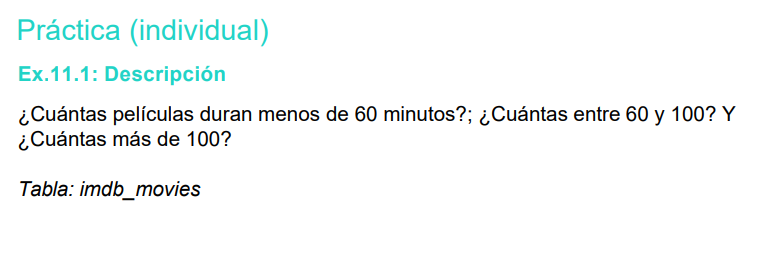
SELECT

round (avg(case when continent= 'Africa'then gross\_income\_per\_capita END),2) AS 'Africa avg GIC',

sum (case when continent= 'Africa' then population\_in\_thousands END) as 'Total Populattion',

COUNT (case when continent= 'Africa' then country end) AS count\_country

FROM world\_health\_org



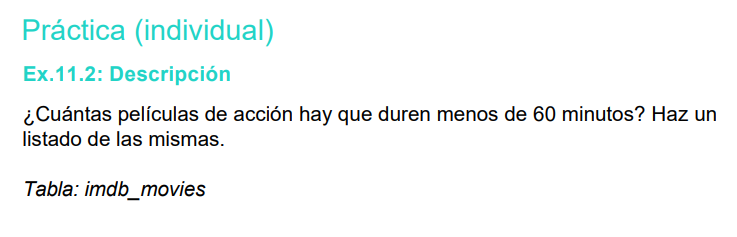
SELECT

count (case when duration <60 then duration end)as 'menor a 60',

count (case when duration BETWEEN 60 and 100 then duration end)as 'entre 60 y 100',

COUNT (case when duration >100 then duration end)as 'mayor a 100'

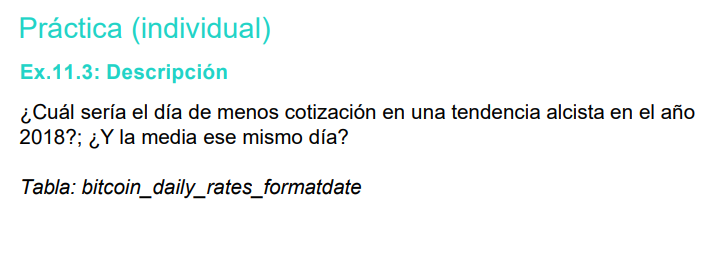
from imdb\_movies

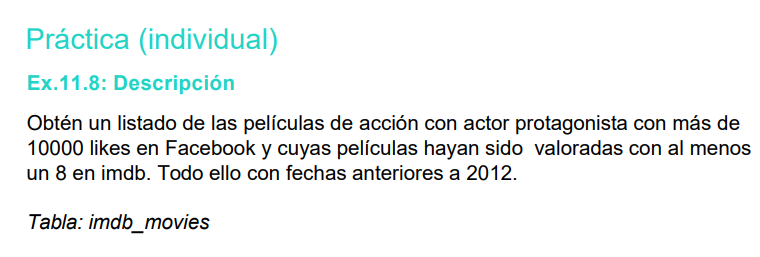


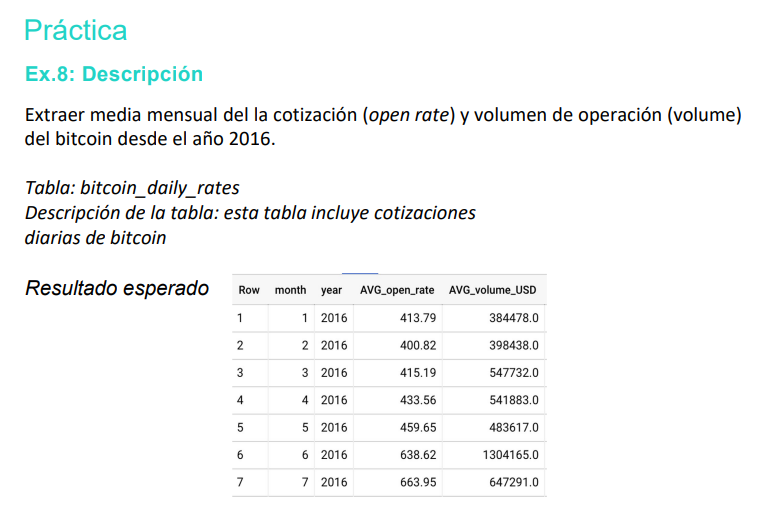
select movie\_title, duration,gender

from imdb\_movies

WHERE duration <60 and gender LIKE '%Action%'







select strftime('%Y',date) as año, strftime('%m',date) as mes,round(avg(open),2) as avg\_open,

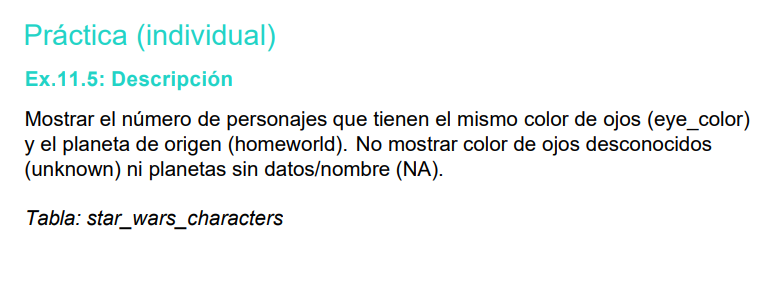
round (avg(volume\_usd),2) as avg\_usd

from bitcoin\_daily\_rates\_formatdate

where año >= '2016'--porque es texto va entre comillas.

GROUP by 1 , 2

ORDER by 1



SELECT COUNT(name), eye\_color,homeworld

FROM star\_wars\_characters\_2

group by eye\_color, homeworld

having COUNT(name) > 1 and homeworld <> 'NA'

order by COUNT(name) desc