--Ej 0: Extraer top 10 personajes de Star Wars con mayor masa (mass)

select name, mass

from star\_wars\_characters\_buena

--WHERE mass IS NOT NULL

WHERE mass != 'NA'

ORDER BY mass DESC

limit 10;

--Castear a float

select name, CAST(mass as float) as mass\_cast

from star\_wars\_characters\_buena

WHERE mass != 'NA'

ORDER BY mass\_cast DESC

limit 10;

select name, mass as float

from star\_wars\_characters\_buena

ORDER BY mass\*1.0 DESC

limit 10;

--Ej 1: Agregar las siguientes métricas para todos los países africanos,

--● average GDP

--● total population

--● number of countries

SELECT

round(AVG(gross\_income\_per\_capita), 2) as 'Africa average GDP',

SUM(population\_in\_thousands) as 'total population',

COUNT(country) as 'nº of countries'

from world\_health\_org

WHERE continent = 'Africa'

--Ej2: Calcular número de personajes según planeta (homeworld). Evitar personajes sin información

--sobre planeta de origen.

SELECT homeworld, COUNT(name) AS 'count\_characters'

FROM star\_wars\_characters\_buena

WHERE homeworld != 'NA'

GROUP BY homeworld

order by 2 DESC;

--Solo los que tienen +10 personajes (HAVING)

SELECT homeworld, COUNT(name) AS 'count\_characters'

FROM star\_wars\_characters\_buena

WHERE homeworld != 'NA'

GROUP BY homeworld

having count\_characters >= 10

order by 2 DESC;

--Ej 3: Calcular el total de salario percibido por cada actor en todas las películas.

--Omitir películas sin data sobre salario.

SELECT actor, round(sum(bond\_actor\_salary), 2) as 'bond\_salary'

from jamesbond\_new

WHERE bond\_actor\_salary != 'NA'

GROUP BY actor

order by bond\_salary DESC;

--Usando HAVING

SELECT actor, round(sum(bond\_actor\_salary), 2) as 'bond\_salary'

from jamesbond\_new

GROUP BY actor

HAVING bond\_salary > 0

order by bond\_salary DESC;

--Ej.4: ¿Podemos asegurar que las películas de acción tienen de media mejor valoración que el resto de películas? Extraer total de películas

--y media de IMDB score para películas de acción vs. el resto (de forma conjunta).

SELECT

CASE

WHEN genres LIKE '%Action%' THEN 'Action Movies'

ELSE 'Rest Of Movies'

END as new\_genres,

count(movie\_title) as 'total\_movies', round(avg(imdb\_score), 4) as 'AVG\_imdb\_score'

from imdb\_movies

group by new\_genres

order by 3 DESC;

--Ej. 5: Calcular la facturación (box office) según director. Filtrar por aquellos directores que hayan generado más de 1500 en el total de

--facturación (todas las películas).

SELECT director, round(sum(box\_office), 2) as total\_box\_office

FROM jamesbond\_new

GROUP BY director

HAVING total\_box\_office > 1500

order by 2 desc;

--Ej. 6: Calcular número total de álbumes según sub metal genre, filtrar por aquellos subgéneros con al menos 10 álbumes.

SELECT sub\_metal\_genre, COUNT(album) as count\_albums

FROM rolling\_top\_albums\_1

GROUP BY sub\_metal\_genre

HAVING count\_albums >= 10

ORDER by 2 desc;

--Ej. 7:¿Cuántos artistas hay incluídos en el dataset cuyo nombre incluye las palabras ‘god’, ‘death’ or ‘black’?

SELECT

CASE

WHEN lower(artist) like '%god%' THEN 'GOD'

WHEN lower(artist) like '%death%' THEN 'DEATH'

WHEN lower(artist) like '%black%' THEN 'BLACK'

end as artist\_keyword,

COUNT(DISTINCT artist) as count\_artist

from rolling\_top\_albums\_1

where artist\_keyword is not NULL

GROUP BY artist\_keyword

order by count\_artist desc;

--Ej. 8: Extraer media mensual del la cotización (open rate) y volumen de operación (volume) del bitcoin desde el año 2016.

SELECT

strftime('%m', date) as month, strftime('%Y', date) as year, avg(volume\_usd) as AVG\_volume\_usd

FROM bitcoin\_daily\_rates\_formatdate

GROUP BY date

order by date ASC

-- Ej. 8: Extraer media mensual del la cotización (open rate) y volumen de operación (volume) del bitcoin desde el año 2016.

SELECT

EXTRACT(month FROM date) as month,

EXTRACT(year FROM date) as year,

avg(volume\_usd) as AVG\_volume\_usd

FROM bitcoin\_daily\_rates\_formatdate

GROUP BY date

order by date ASC

-- Ej. 9: ¿Cuál fue la semana con el valor mayor de cotización? Utilizar cotización high.

SELECT

EXTRACT(week FROM date) as 'nº semana',

max(high) as 'mayor cotización'

from amazon\_stocks\_formatdate

group by 1

order by 2 DESC

limit 1

-- otra forma

SELECT

EXTRACT(week FROM date) as 'nº semana',

high as 'mayor cotización'

from amazon\_stocks\_formatdate

order by high DESC

limit 1;

SELECT

EXTRACT(week FROM date) as 'nº semana',

round(max(high), 2) as 'mayor cotización'

from amazon\_stocks\_formatdate

group by 1

-- Ej. 10: Calcular el total de sesiones según canal para octubre de 2019. Crear métricas específicas agregadas para cada dispositivo.

SELECT channelgrouping,

sum(case when devicecategory = 'mobile' then sessions END) AS mobile\_sessions,

sum(case when devicecategory = 'mobile' then sessions END) AS desktop\_sessions,

sum(case when devicecategory = 'mobile' then sessions END) AS tablet\_sessions,

sum(sessions) AS total\_sessions

from google\_analytics\_formatdate

where EXTRACT(year FROM date) = 2019 and EXTRACT(month FROM date) = 10

group by channelgrouping