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, na\_sales, eu\_sales, other\_sales

FROM videogames\_games

WHERE year >= 1995 AND NOT name LIKE '%Donkey Konga 2%' AND

na\_sales BETWEEN 0 AND 0.31 AND

eu\_sales BETWEEN 0 AND 0.32 AND

other\_sales BETWEEN 0 AND 0.30

AND publisher = 'Nintendo'

ORDER BY year

LIMIT 1

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,

CASE

WHEN height > 200 THEN 'GRANDE'

ELSE 'PEQUEÑO'

END tamaño

from star\_wars\_characters\_2

WHERE eye\_color <> 'red' AND

homeworld IN ('Chandrila', 'Stewjon','Tatooine')

ORDER BY height

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 rank, company, sector, industry, profits,

CASE

WHEN profits >= 1000 THEN 'rentable'

ELSE 'No rentable'

END as rentabilidad

from fortune

WHERE company <> 'Citigroup'

AND sector = 'Financials' AND industry = 'Diversified Financials'

AND rank <= 100

ORDER BY rank

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

CASE

WHEN total\_paro\_registrado >= 1000 THEN 'mucho'

WHEN total\_paro\_registrado > 500 THEN 'medio'

ELSE 'poco'

END AS comparacion\_de\_paro

FROM gobierno\_paro

WHERE paro\_hombre\_edad\_25\_\_45 BETWEEN 20 AND 1100 AND paro\_mujer\_edad\_25\_\_45 BETWEEN 20 AND 1100

ORDER BY total\_paro\_registrado, municipio

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, height, species, homeworld 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).

Teniendo en cuenta sólo estas regiones (ignorar other\_sales), queremos determinar en qué región ha vendido mejor cada juego. Para ello, generar un campo nuevo best\_selling\_region cuyo valor sea ‘NA’, si NA\_sales es mayor que EU\_sales y que JP\_sales; ‘EU’ en caso de que EU\_sales sea mayor que NA\_sales y JP\_sales o ‘JP’ en el el tercer caso.

SELECT name, platform\_code, year, publisher,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 > na\_sales and jp\_sales > eu\_sales THEN 'JP'

END as best\_seling\_region

FROM videogames\_games

WHERE platform\_code = 'DS'

ORDER BY year, na\_sales DESC ,eu\_sales DESC, jp\_sales DESC

7.- Cuales son las películas en USA, ordenados 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 'CORTAS'

-- ELSE 'FUERA DE RANGO'

END AS largo

FROM imdb\_movies

WHERE country = 'USA'

ORDER BY largo 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.

SELECT mes, comunidad\_autonoma, provincia, municipio,paro\_mujer\_edad\_\_\_45,paro\_sin\_empleo\_anterior FROM gobierno\_paro

WHERE paro\_sin\_empleo\_anterior = 0 AND codigo\_mes = 201803

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 movie\_title, color, title\_year, country,

CASE

WHEN movie\_facebook\_likes > 10000 THEN 'Muy Buena'

WHEN movie\_facebook\_likes BETWEEN 5000 AND 10000 THEN 'Buena'

WHEN movie\_facebook\_likes < 5000 THEN 'Regular'

END AS calificacion\_segun\_facebook

FROM imdb\_movies

WHERE color = 'Color' AND country = 'USA'and title\_year BETWEEN 2000 AND 2010

LIMIT 50