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

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,

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')

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

company,

CASE

WHEN profits >= 1000 THEN 'Rentable'

ELSE 'No rentable'

END AS Rentabilidad

FROM fortune

WHERE

company != 'Citigroup' AND

sector = 'Financials' AND

industry = 'diversified financials'

ORDER BY rank DESC

LIMIT 100

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

municipio,

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

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

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, 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'

ELSE 'JP'

END AS best\_selling\_region

FROM videogames\_games

WHERE

platform\_code = 'DS'

ORDER BY year

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,

CASE

WHEN duration BETWEEN 120 AND 200 THEN 'SON MUY LARGAS'

WHEN duration < 120 THEN 'son cortas'

END AS duracion

FROM imdb\_movies

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.

SELECT

municipio,

paro\_mujer\_edad\_\_\_45

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

movie\_title,

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 Calificacion\_segun\_Facebook

FROM imdb\_movies

WHERE

color = 'Color' AND

title\_year BETWEEN 2000 AND 2010

LIMIT 50

## CONSULTAS DE COMPAÑEROS

Ordenar todos los álbumes que contengan la letra W y se sacaran entre los años 1975 y 2001 según la paciencia del público para ello crear un campo que se denomine Paciencia del público en el que los álbumes con una duración de 50 minutos sea “ADECUADO”, entre 51 y 70 sea “SOPORTABLE”, más de 70 sea “INSOPORTABLE” y lo demás “IDEAL”.

SELECT

album,

CASE

WHEN duration > 70 THEN 'INSOPORTABLE'

WHEN duration BETWEEN 51 AND 70 THEN 'SOPORTABLE'

WHEN duration = 50 THEN ‘ADECUADO’

ELSE 'IDEAL'

END AS Paciencia

FROM rolling\_top\_albums\_1

WHERE

album LIKE ('%W%') AND

release\_year BETWEEN 1975 AND 2001

ORDER BY Paciencia;

De la tabla starstar\_wars\_characters\_2. Mostrar los personajes que sean humanos, hutt, wookie o ewok con peso no nulo ordenando por peso de mayor a menor. Añadir un nuevo campo 'Recomendación de la abuela' dónde se indique que si pesan más de 1000 necesitan comer más lechuga, si su peso está entre 90 y 1000 están fuertes y si pesan menos de 90 necesitan comer más.

SELECT name,

CASE

WHEN mass > 1000 THEN 'Necesitan comer mas lechuga'

WHEN mass BETWEEN 90 AND 1000 THEN 'Estan fuertes'

ELSE 'Necesitan comer mas'

END AS Recomendacion\_de\_la\_abuela

FROM star\_wars\_characters\_2

WHERE

species IN ('Human','Hutt','Wookie','Ewok') AND

mass NOT NULL

ORDER BY mass DESC;

De la tabla imdb\_movies. Cuales son las 10 películas que tienen mayor  puntuación de imdb, y de esas 10 cuáles son las de mayor número de votos de los usuarios.  Visualizar solo las que no sean western. Crea un campo que clasifique las películas por su número de votos de los usuarios que se llame “votos” donde si tiene más de 7 se llamen top, si tienen entre 5 y 7  se llamen se puede ver y si es menos que se llame no pierdas el tiempo.

SELECT movie\_title, imdb\_score, num\_voted\_users,

CASE

WHEN num\_voted\_users > 7 THEN 'Top'

WHEN num\_voted\_users BETWEEN 5 AND 7 THEN 'Se puede ver'

ELSE 'No pierdas el tiempo'

END AS Que\_tal\_es

FROM imdb\_movies

WHERE

gender != 'western'

ORDER BY imdb\_score DESC, num\_voted\_users DESC

LIMIT 10;