/\*  
 Instructions:  
 - Create a branch named "wine"  
 - Solve the data-requests from this file using SQL.  
 - Add/Commit/Push your changes to github (individual).  
 Note: You can work along with your project team.  
 \*/  
  
-- EX.1) Get the top 10 countries with more population density  
  
SELECT  
 name,  
 population / area\_km2 AS "density"  
FROM  
 country  
WHERE  
 area\_km2 > 0  
ORDER BY density DESC -- DESC es descendente  
LIMIT 10  
;  
  
-- EX.2) Get the count of male/female tasters.  
SELECT DISTINCT gender FROM taster; --DISTINCT AGARRA LOS VALORES UNICOS  
  
SELECT  
 gender,  
 *count*(\*)  
FROM  
 taster  
WHERE  
 --LOWER(gender) IN ('male', 'female')  
 --LOWER(gender) NOT LIKE 'undef%'  
 *LOWER*(gender) LIKE '%male'  
GROUP BY  
 gender  
;  
  
  
-- EX.3) Get the percentage of male/female tasters.  
WITH taster\_valid AS (  
 SELECT  
 \*  
 FROM  
 taster  
 WHERE  
 *lower*(gender) IN ('male', 'female')  
), taster\_gender\_agg as (  
 SELECT  
 gender,  
 *COUNT*(\*)::NUMERIC(7,2) gender\_sum --7 ES LA PARTE ENTERO Y 2 SON LOS DECIMALES  
 FROM  
 taster\_valid  
 GROUP BY  
 gender  
), taster\_total AS (  
 SELECT  
 *COUNT*(\*)::NUMERIC(7, 2) total  
 FROM  
 taster\_valid  
)  
SELECT  
 gender,  
 *TRUNC*(100 \* gender\_sum / total, 2) percentage  
FROM  
 taster\_gender\_agg,  
 taster\_total  
;  
-- EX.4) How many countries share the same first digit on their country-code?  
-- Show only those digits with more than 20 countries.  
SELECT DISTINCT (code, 1) FROM country;  
  
SELECT  
 *LEFT*(code, 1) first\_digit,  
 *COUNT*(\*) country\_count  
FROM  
 country  
GROUP BY  
 first\_digit  
HAVING  
 *COUNT*(\*) > 20  
;  
  
-- EX.5) Get the % of countries are not labeled as a trillion usd gdp and  
-- do have a null happiness\_score.  
SELECT \* FROM country;  
  
SELECT  
 100 \* *COUNT*(\*) / *MAX*(t.total) Percentage  
FROM  
 country,  
 (SELECT *COUNT*(\*) total FROM country) t  
WHERE  
 --LOWER(SPLIT\_PART(gdp\_usd, ' ', 2)) != 'trillion'  
 *LOWER*(gdp\_usd) NOT LIKE '%trillion'  
 AND happiness\_score IS NULL  
;  
  
  
-- COUNTRY ANALYSIS  
  
-- A) Get the average happiness\_score of the countries labeled with a GDP  
-- of "billion" and "trillion".  
SELECT  
 *AVG*(happiness\_score)  
FROM  
 country  
WHERE  
 *LOWER*(*SPLIT\_PART*(gdp\_usd, ' ', 2)) IN ('billion', 'trillion')  
;  
  
-- B) Show a table with the country name, population, area, gdp, and happiness core of the  
-- the G7 countries (i.e., `Canada`, `France`, `Germany`, `Italy`, `Japan`, `United Kingdom`, `United States`)  
-- order by happiness\_score (DESC).  
SELECT  
 name,  
 population,  
 area\_km2,  
 gdp\_usd,  
 happiness\_score  
FROM  
 country WHERE name IN ('Canada', 'France', 'Germany', 'Italy', 'Japan', 'United Kingdom', 'United States')  
ORDER BY  
 happiness\_score DESC  
;  
  
  
-- C) Create a custom score called "score" using this formula: happiness\_score \* density  
-- Where `density` is population / area\_km2. Show the top 10 countries (name) with greater score.  
WITH  
 country\_density AS (SELECT name, *DIV*(population, area\_km2) AS "density"  
FROM  
 country  
WHERE  
 area\_km2 > 0),  
 country\_score AS (SELECT name, happiness\_score FROM country WHERE area\_km2 > 0)  
SELECT  
 country\_score.name, (country\_score.happiness\_score \* country\_density.density) AS score  
FROM  
 country\_density, country\_score  
WHERE  
 country\_score.name = country\_density.name  
AND (country\_score.happiness\_score \* country\_density.density) IS NOT NULL ORDER BY score DESC LIMIT 10  
;  
  
-- D) Get the number of wines per variety, ordered by the latter (desc)  
SELECT  
 variety,  
 *COUNT*(variety)  
FROM  
 wine  
GROUP BY  
 variety  
ORDER BY  
 *COUNT*(variety) DESC  
;  
  
  
-- E) How many wines are registered per country? Show country name and number of wines, ordered by the latter (desc).  
SELECT  
 name,  
 *COUNT*(name)  
FROM  
 wine JOIN country on wine.country\_id = country.id  
GROUP BY  
 name  
ORDER BY  
 *COUNT*(name) DESC  
;  
  
  
-- REVIEW ANALYSIS  
  
-- F) What's the average wine price and points per province?  
-- Show the province, avg-price, avg-points when the avg-points are grater than 85.  
-- Ordered by avg-price and then by avg-points.  
SELECT  
 province,  
 *AVG*(price),  
 *AVG*(points)  
FROM  
 review  
 JOIN wine on review.wine\_id = wine.id  
GROUP BY  
 province  
HAVING  
 *AVG*(points) > 85  
ORDER BY  
 *AVG*(price) DESC, *AVG*(points) DESC  
;  
  
-- G) What's the average wine price and points of the countries with more than a 7 in their happiness score?  
-- Show the country, avg-price, avg-points.  
-- Ordered by avg-points and then by avg-price.  
SELECT  
 name,  
 *AVG*(points),  
 *AVG*(price)  
FROM  
 review  
 JOIN wine on review.wine\_id = wine.id  
 JOIN country on wine.country\_id = country.id  
GROUP BY  
 name,  
 happiness\_score  
HAVING  
 happiness\_score > 7  
ORDER BY  
 *AVG*(points) DESC, *AVG*(price) DESC  
;  
  
  
-- H) What's the min, avg, and max wine points per taster gender (excluding undefined) and wine variety starting with "Cabernet".  
-- Order by: variety, gender  
SELECT  
 variety,  
 gender,  
 *MIN*(points),  
 *AVG*(points),  
 *MAX*(points)  
FROM  
 review  
 JOIN wine on review.wine\_id = wine.id  
 JOIN taster on review.taster\_id = taster.id  
WHERE  
 *LOWER*(gender) NOT IN ('undefined') AND variety LIKE 'Cabernet%'  
GROUP BY  
 variety,  
 gender  
ORDER BY  
 variety,  
 gender  
;  
  
  
-- I) Create the following custom score called "wine\_quality\_and\_happiness\_index": happiness\_score \* avg(points) / 100  
-- Get the score per country and order by the value (desc).  
SELECT  
 name,  
 (*AVG*(happiness\_score) \* *AVG*(points) / 100) AS wine\_quality\_and\_happiness\_index  
FROM  
 country  
 JOIN wine on country.id = wine.country\_id  
 JOIN review on wine.id = review.wine\_id  
GROUP BY  
 name  
ORDER BY wine\_quality\_and\_happiness\_index DESC  
;