```
# World Life Expectancy Project (Exploratory Data Analysis)

# Alex says that there are two parts to exploratory data analysis, or EDA.
# The first part is in conjunction with data cleaning and the goal is to find out
# what's wrong with the dataset and should be fixed. The second part focuses on
# finding insights or trends in the data that can be used in the future.

# Look at the life expectancy table and try to find anythnig potentialy helpful.
SELECT *
FROM world_life_expectancy
;

# Look at how different countries did in terms of increasing (or maybe even decreasing)
# the life expectancy of their people.
SELECT Country, MIN(`Life expectancy`), MAX(`Life expectancy`)
FROM world_life_expectancy
GROUP BY Country
ORDER BY Country DESC
;

# In the result of the above query, there were some countries
# where the min or max life expectancy was 0, which could indicate
# a data quality issue; therefore, the query above will be run again
# with a filter to exclude rows that have these zeros.
# Note: Alex uses <> for inequality. I personally use != for inequality.
# The outcome should be the same.
SELECT Country, MIN(`Life expectancy`), MAX(`Life expectancy`)
FROM world_life_expectancy
GROUP BY Country
HAVING 0 != MIN(`Life expectancy`)
        AND 0 != MAX(`Life expectancy`)
ORDER BY Country DESC
;

# Run the above query again but this time include the range (i.e., max - min)
# of the life expectancy values, and order those ranges from highest to lowest
# Note: I used a title for the ranges that's different from what Alex uses in the video.
# He says that these ranges are for a span of fifteen years; however, I noticed during the
# data cleaning process that the United States has sixteen years of data in the table. Since there
# could be some countries that have fewer years on record than other countries,
# I'll just keep the title a little more general
SELECT Country, MIN(`Life expectancy`), MAX(`Life expectancy`),
        ROUND((MAX(`Life expectancy`) - MIN(`Life expectancy`)), 1) AS
Life_Increase_Over_All_Years_on_Record
FROM world_life_expectancy
GROUP BY Country
HAVING 0 != MIN(`Life expectancy`)
        AND 0 != MAX(`Life expectancy`)
```

```
ORDER BY Life_Increase_Over_All_Years_on_Record DESC
;

# Run the above query again but sort the ranges in ascending order to see which countries
# had the smallest increase in life expectancy.
SELECT Country, MIN(`Life expectancy`), MAX(`Life expectancy`),
	ROUND((MAX(`Life expectancy`) - MIN(`Life expectancy`)), 1) AS
Life_Increase_Over_All_Years_on_Record
FROM world_life_expectancy
GROUP BY Country
HAVING 0 != MIN(`Life expectancy`)
	AND 0 != MAX(`Life expectancy`)
ORDER BY Life_Increase_Over_All_Years_on_Record ASC
;

# It looks like countries that have a high min value to begin with
# will have small differences from the max value

# I'm noticing that the results of the EDA so far are prompted by curiosity about the data.
# Like...first asking one question about a subset of the data, and then asking further questions
# about that subset, and so on and so forth.

# I do have a concern here, though. Alex is investigating min and max values for life expectancy.
# However, it's not clear where in the years recorded that these min and max values occur. It seems like
# Alex is assuming that the min value occurs during the first year on record while the max occurs
# during the last year on record, and that progress had been close to, if not definitely, linear.
# However, it is possible that there could've been some wild fluctuations over the course of the years
# that could cause the life expectancy to start at one value, dip for one reason, spike upwards for
another,
# and then return to the starting value. It may be worthwhile to do my own EDA with this dataset later
on;
# for now, it might make more sense to just follow along with what Alex is doing in the video.

# Look at the average life expectancy of all countries recorded by year.
# Round the averages to two decimal places in order for the numbers to look
# cleaner. Also, filter out rows where the life expectancy for any given year is zero.
SELECT Year, ROUND(AVG(`Life expectancy`), 2)
FROM world_life_expectancy
WHERE 0 != `Life expectancy`
GROUP BY Year
ORDER BY Year ASC
;

# Overall, it looks like there's a persistent increase of average life expectancy over the years. Worth
noting
# is that in addition to observing the central tendency, it may be worthwhile to consider spread around
that
# central tendancy (standard deviation in this case).
# Review the life expectancy table to see if there's anything else interesting to look at.
SELECT *
FROM world_life_expectancy
;

# Alex says that correlations between columns can be calculated in SQL,
# and he would like to look into any correlations between life expectancy and GDP.
SELECT Country, `Life expectancy`, GDP
FROM world_life_expectancy
;

# Look at the average life expectancy for each country.
# Round these averages to one decimal place this time because the original numbers
# are rounded to one decimal place. Look at the average GDP for each country, and
# round these averages to one decimal place. I'm using aliases for these averages
# that are different from what Alex uses in order to better describe what these valaues represent.
# Also, this time around Alex doesn't filter out rows where the life expectancy
# for any given year is zero. I'll do the same as well for the sake of trying to match his output; however,
# I've kept a WHERE statement as a comment to reflect that I think that it might be worthwile
# to filter out rows where life expectancy is zero.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
;

# In the output of the above query, there are some countries with an average GDP of zero.
# Moreover, the Cook Islands have zeros for both average life expectancy and average GDP.

# Repeat the above query, but now order the average life expectancy values in ascending order.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
ORDER BY Avg_Life_Expectancy ASC
;

# Alex notes that the countries with zeros for average life expectancy are small countries
# and may not be reporting their life expectancy figures, so their fields are filled with zeros instead.
# However, the presence of zeros could be a problem when performing EDA

# Repeat the above query, but this time filter retain only the rows that have average life expectancy
values
# greater than zero and an average GDP greater than zero.
```

```
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
        AND 0 < Avg_GDP
ORDER BY Avg_Life_Expectancy ASC
;

# Repeat the above query but this time order by average GDP in ascending order.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
        AND 0 < Avg_GDP
ORDER BY Avg_GDP ASC
;

# Review the average life expectancy of all countries recorded by year.
# Round the averages to two decimal places in order for the numbers to look
# cleaner. Also, filter out rows where the life expectancy for any given year is zeros.
SELECT Year, ROUND(AVG(`Life expectancy`), 2)
FROM world_life_expectancy
WHERE 0 != `Life expectancy`
GROUP BY Year
ORDER BY Year ASC
;

# Alex says it's helpful to save past queries in order to reuse them later on;
# however, I'm inclined to copy-and-paste older queries in order to retain
# proper accounting of my thought processes

# According to Alex, it looks like the average of the average life expectancies found in the above query
# is about sixty-eight years.

# Repeat the query two above, to see how countries compare to this average of averages
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
        AND 0 < Avg_GDP
ORDER BY Avg_GDP ASC
;
```

```
# Alex notes that countries with lower aveage GDPs appear to have lower average life expectancies,
# possibly because of suboptimal healthare or social infrastructure in those countries.

# I've been under the impression that Alex would use some sort of built-in function to calculate
correlation,
# and to be real I feel more comfortable taking that approach. Eyeballing correlations feels risky to me
# becuase human error can negatively impact analysis. I did a quick Google search, and one source states
that there is
# a function that can calculate correlation values between two columns - CORR(Y, X) - but it's not
supported by MySQL
# (source = https://datacomy.com/sql/functions/aggregation/correlation/). However, after further
investigation
# another source points out that MySQL can be "extended" with a package of statistical functions,
including
# the above correlation function (source = https://sqlstat.sourceforge.net/index.html). For now, I think I'll
just
# do my best to follow along with what Alex is doing in the video.

# Repeat the above query but sort the average GDPs in descending order.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(GDP), 1) AS
Avg_GDP
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
	AND 0 < Avg_GDP
ORDER BY Avg_GDP DESC
;

# It looks like countries with high average GDPs also have high average life expectancies.

# Alex says that in order to confirm that there is a correlation between these two columns,
# other tools like Tableau or Power BI can generate visualizations based off of this data. I suppose
# that there really are limitations to what SQL (or at least MySQL) can do.

# Attempt to place average life expectancies and average GDP values into bins, and then
# filter countries based off of those bins. First though look at the full life expectancy table
# in order to guesstimate a halfway point of the GDP values for binning purposes.
SELECT *
FROM world_life_expectancy
ORDER BY GDP
;

# Assign numerical bin tags based off of that guesstimated halfway point and then add up the number of
items per bin
# to determine how many high GDP rows there are
SELECT
```

```
SUM(CASE
        WHEN 1500 <= GDP THEN 1
   ELSE 0
END) AS High_GDP_Count
FROM world_life_expectancy
ORDER BY GDP
;

# Run another query that's similar to the one above but with an extra column
# similar to the High_GDP_Count column.
# The NULL value in the CASE statement inside of the AVG() function is important
# for not bringing down the avarage of the life expectancies for the high GDP countries.
# I used a column name for the average that's different from the column name used by Alex
# in order for me to better understand what's going on.
# Reminder: The average value in the ouput of the query below is the average for the
# rows labeled as high GDP, not for all of the countries.
SELECT
SUM(CASE WHEN 1500 <= GDP THEN 1 ELSE 0 END) AS High_GDP_Count,
AVG(CASE WHEN 1500 <= GDP THEN `Life expectancy` ELSE NULL END) AS
Avg_High_GDP_Life_expectancy
FROM world_life_expectancy
ORDER BY GDP
;

# Repeat the above query but now also take into consideration the low GDP countires.
SELECT
SUM(CASE WHEN 1500 <= GDP THEN 1 ELSE 0 END) AS High_GDP_Count,
AVG(CASE WHEN 1500 <= GDP THEN `Life expectancy` ELSE NULL END) AS
Avg_High_GDP_Life_expectancy,
SUM(CASE WHEN GDP <= 1500 THEN 1 ELSE 0 END) AS High_GDP_Count,
AVG(CASE WHEN GDP <= 1500 THEN `Life expectancy` ELSE NULL END) AS
Avg_Low_GDP_Life_expectancy
FROM world_life_expectancy
ORDER BY GDP
;

# Alex is using his understanding of the data derived from previous queries to look around
# and find anything interesting. He also says that the work done here with life expectancy and GDP
# can be applied to just about any column.

# I do have a conern about the output of the above query. It is my understanding that some countries
# have 0 as their GDP value, so they'd be considered low GDP countries by one of the CASE statements.
# Consider the following query
SELECT Country, `Status`, AVG(`Life expectancy`)
FROM world_life_expectancy
WHERE GDP = 0
GROUP BY Country, `Status`
ORDER BY `Status` ASC, Country ASC
```

```
;

# Based on the output of the above query, the collection of countries with GDP values of zero includes
both developed countries
# (e.g., the UK and the US) and developing countires (e.g., Iraq and Iran). Interestingly, the Republic of
Korea
# (i.e., South Korea) is classified as developing.

SELECT Country, `Year`, `Status`
FROM world_life_expectancy
WHERE Country = 'Republic of Korea'
;

# Based off of the above query, the Republic of Korea had been a developing country for several years
even though now
# I'd consider this country to be an economic and cultural powerhouse.

# Alas, my own curiosity and growing understanding of this data is taking me along my own EDA,
# so I might as well for now continue following along with Alex.

# Alex would like to investigate average life expectancy for the different country statuses. Currently there
# are only two statuses, developing and developed.
SELECT `Status`, ROUND(AVG(`Life expectancy`), 1)
FROM world_life_expectancy
GROUP BY `Status`
;

# Alex would like to investigate the number of countries classified as either
# developing or developed
SELECT `Status`, COUNT(DISTINCT Country)
FROM world_life_expectancy
GROUP BY `Status`
;

# Based off of the result of the above query, Alex figures that the average life expectancy
# can be skewed in favor of the developed countries because they have so few countries to bring their
value down.
# I have an understanding of what he's trying to argue, but at the same time I also have a gut feeling
# that there might be a flaw in the logical progression of his argument, either explicity mentioned or
implied.
# Maybe I'll be able to better articulate the rationale of my gut instinct later on, but perhaps the
development status
# has more of an impact on average life expectancy rather than number of members in each status, and
perhaps
# the number members in status has more of an impact on the distribution around that mean (i.e., the
standard deviation)
# than on the average itself.
# Alex would like to combine the two queries above into one.
SELECT `Status`, COUNT(DISTINCT Country), ROUND(AVG(`Life expectancy`), 1)
FROM world_life_expectancy
GROUP BY `Status`
;

# Alex would now like to look into the BMI.
# First look at the table once more to determine what to compare the BMI to.
SELECT *
FROM world_life_expectancy
;

# Alex has decided to look at average life expetancy and average BMI over the recorded years
# and group those values by country. I've opted to use column headers that better help me
# keep track of what's happening.
# Alex points out that while it can be worthwhile to rewrite code from scratch for learning purposes,
# in the real world he's more inclined to just reuse code that's been written already in order
# to save some time.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(BMI), 1) AS
Avg_BMI
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
        AND 0 < Avg_BMI
ORDER BY Avg_BMI DESC
;

# Alex is initially shocked by these BMI numbers because 60+ just seems so high.
# At the same time, I'm thinking that there are no units displayed in the column names,
# and as far as I can recall Alex hasn't mentioned a data dictionary, and if such a
# dictionary exists, then maybe some of the BMI values can make sense.

# Next, Alex tries to find any correlation between average live expectancy and
# average, and based off of the output from the above query it looks like countries
# with above-average life expectancy values have high BMI values. Alex admits that he's not a doctor,
# but as far as he understands, high BMI values relates to higher likelihoods of heart attacks
# occurring. Again, it's not clear what the units of BMI are here, especially since not every country
# will use the same units of measurement. A quick Google search on units used for BMI shows that
# according to Wikipedia, BMI can be expressed in kg/meter-squared, and that when converting to
pounds and feet,
# a convesion factor of 4.88 needs to be used (source =
https://en.wikipedia.org/wiki/Body_mass_index).
# So yeah...a data dictionary with details about the data here can be really helpful, ya know?

# Alex reasons something to the effect that countries that are developed and have high GDP values are
likely
# to have well-fed populations which can contribute to high BMI values.
# Run the same query as above but this time sort the average BMI values in ascending order
# to look at countries with low BMI values.
SELECT Country, ROUND(AVG(`Life expectancy`), 1) AS Avg_Life_Expectancy, ROUND(AVG(BMI), 1) AS
Avg_BMI
FROM world_life_expectancy
# WHERE 0 != `Life expectancy`
GROUP BY Country
HAVING 0 < Avg_Life_Expectancy
	AND 0 < Avg_BMI
ORDER BY Avg_BMI ASC
;

# So with the exception of Viet Nam, some countries with low average BMI values
# also have lower than average life expectancy values. This could be a result of
# limited access to food or maybe an inclination to just eat less.

# At any rate, any findings here can be expanded upon with visualiation tools
# in future data analyses.

# Look at the the whole life expectancy table again
SELECT *
FROM world_life_expectancy
;

# Alex would like to look at adult mortality to see how many people are dying each year
# in a country and how that count relates to life expectancy.

# I'm already thinking about the criteria for adulthood because
# the age of majority can vary from country to country, and some teens might
# be excluded from the caluclation for adult mortality on a country-by-country basis.
# Also, even if an age is established as a universal threshold for adulthood for all countries,
# neither infant deaths nor child deaths will necessarily be included in adult mortality
# calculations. Morever, it is my understanding that average life expectancy can be brought down
# by infant or child deaths, so the exclusion of some data in one calculation and inclusion
# of that same data in another calculation can make comparing those colculations
# questionable.

# At any rate, Alex mentions a rolling total for his next analysis,
# which will showcase a sum of the current value and the sum of all the
# values above that current value. It will involve a window function,
# which will include a partition over a subset of the overall table.
# Here is a source of information that I found with a Google search on rolling sums
# in SQL: https://medium.com/@mattdamberg/calculating-running-total-in-sql-850c5b072513
SELECT Country,
Year,
`Life expectancy`,
`Adult Mortality`,
SUM(`Adult Mortality`) OVER(PARTITION BY Country ORDER BY `Year`) AS Rolling_Total
FROM world_life_expectancy
;

# Alex suspects that the fact that only three adults died in Afghanistan in 2009
# is a data quality issue.

# Repeat the above query once more, but try to focus on the data associated with
# the United States of America.
# Note: With the query that Alex will end up using,
# some other countries with the word 'United' in their name can show up.
SELECT Country,
Year,
`Life expectancy`,
`Adult Mortality`,
SUM(`Adult Mortality`) OVER(PARTITION BY Country ORDER BY `Year`) AS Rolling_Total
FROM world_life_expectancy
WHERE Country LIKE '%United%'
;

# Out of curiosity, I looked up the definition of adult mortality,
# and it looks like it is a count per 1,000 persons.
# (Source = https://datahelpdesk.worldbank.org/knowledgebase/articles/114956-what-is-the-definition-
of-adult-mortality)
# With that said, I'm not sure if applying the rolling total over adult mortality
# really makes sense. Nevertheless, I'll follow along with Alex over the course
# of this video.

# Review the whole life expectancy table to see if there's a column for total population for each country.

SELECT *
FROM world_life_expectancy
;

# No such column exists in the table, but Alex says that that data might be something that would have to
be
# acquired from online and then joined to the world life expectancy table. If country population could be
# factored in, Alex says that he'd like to see country population, life expectancy, and adult mortality
together.

# By the end of the video, a lot of different things had been observed. Alex recounts all of the work that
had been done,
# and he points out that even more work can be done with what's present in the table. I'd imagine even
more
# work can be done of extra data is merged with the life expectancy table.

# Worth noting here is that somewhere along the way, I've developed a habit of putting a grave accent (`)
around
```

```
# column name if they happen to be the same as SQL keywords. That way, I make it explicitly clear
# that I want to work with a column name and not a SQL keyword, ya know?
```