In [1]:
```
!pip show astropy
```

```
Name: astropy
Version: 7.0.1
Summary: Astronomy and astrophysics core library
Home-page:
Author:
Author-email: The Astropy Developers <astropy.team@gmail.com>
License: BSD-3-Clause
Location: C:\Users\btcya\Desktop\astronomy-course-programming\astrocourse\Lib\sit
e-packages
Requires: astropy-iers-data, numpy, packaging, pyerfa, PyYAML
Required-by:
```

In [2]:
```
# Import the package
import astropy

# Check the installed version
print(astropy.__version__)
```

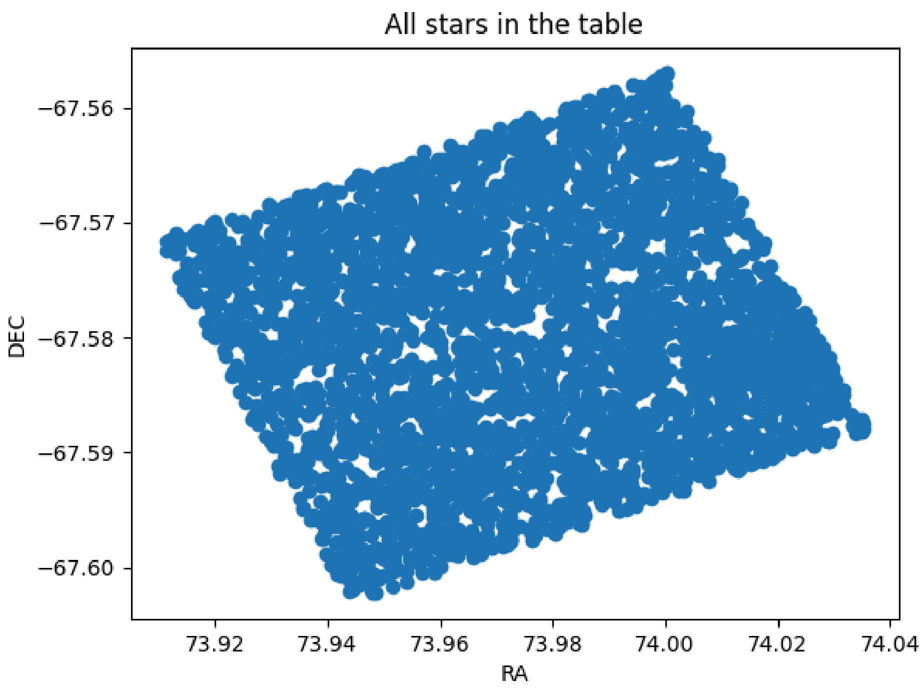
```
7.0.1
```

Explanation of the Code

1. **Import Required Libraries**:
    - `numpy` : For numerical operations.
    - `matplotlib.pyplot` : For plotting data.
    - `astropy.io.fits` : For reading FITS (Flexible Image Transport System) files.
    - `astropy.wcs` : For handling World Coordinate System (WCS) information.
    - `astropy.table` : For working with tabular data.
    - `astropy.visualization.make_lupton_rgb` : For creating RGB images (not used in this code).
    - `plt.ion()` : Enables interactive mode for Matplotlib, allowing plots to update dynamically.
2. **Open and Read FITS File**:
    - The FITS file `hst_results_nd.fits` is opened using `fits.open` .
    - The header (`hdr`) is extracted from the first HDU (Header Data Unit) to access metadata.
    - A WCS object (`wcs`) is created from the header for celestial coordinate transformations.
    - The data (`data`) is extracted from the first HDU.
3. **Read the Table from FITS File**:
    - The FITS file is read as an Astropy table (`t`), which contains tabular data such as star positions and magnitudes.
4. **Access and Print Table Information**:
    - The Right Ascension (RA) and Declination (DEC) of the first row in the table are extracted.
    - The column names of the table are retrieved and printed.

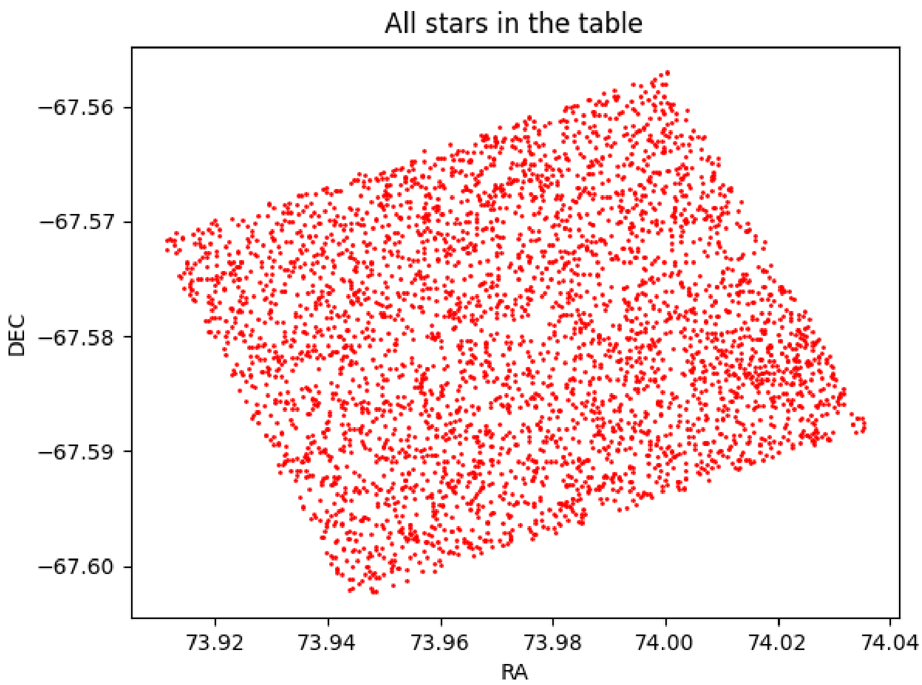
5. **Plot All Stars in the Table**:

- A scatter plot is created using the RA and DEC values of all stars in the table.
- The x-axis is labeled as "RA", and the y-axis is labeled as "DEC".
- The plot is titled "All stars in the table".



6. **Plot Stars with Smaller Red Markers**:

- Another scatter plot is created with red circular markers (`'ro'`) and a smaller marker size (`markersize=1`) for better distinction.
- The x-axis is labeled as "RA", and the y-axis is labeled as "DEC".
- The plot is titled "All stars in the table".



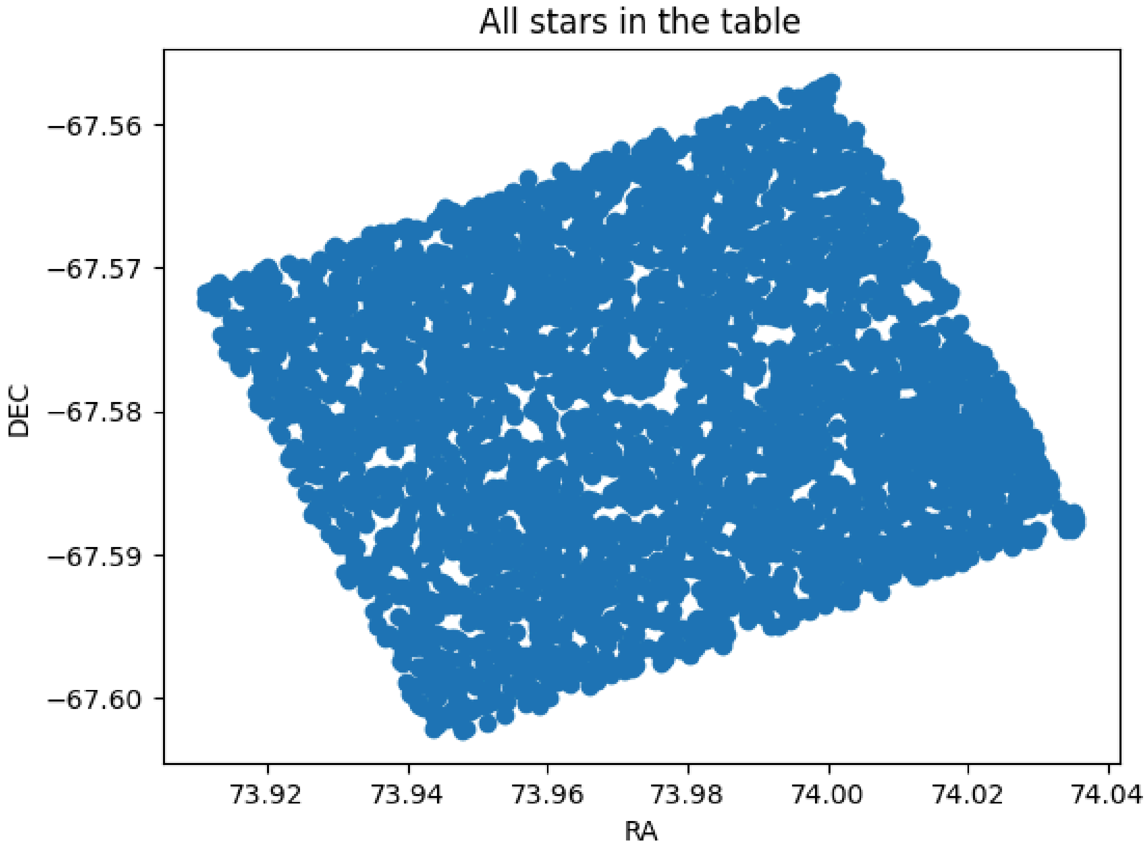
Summary:

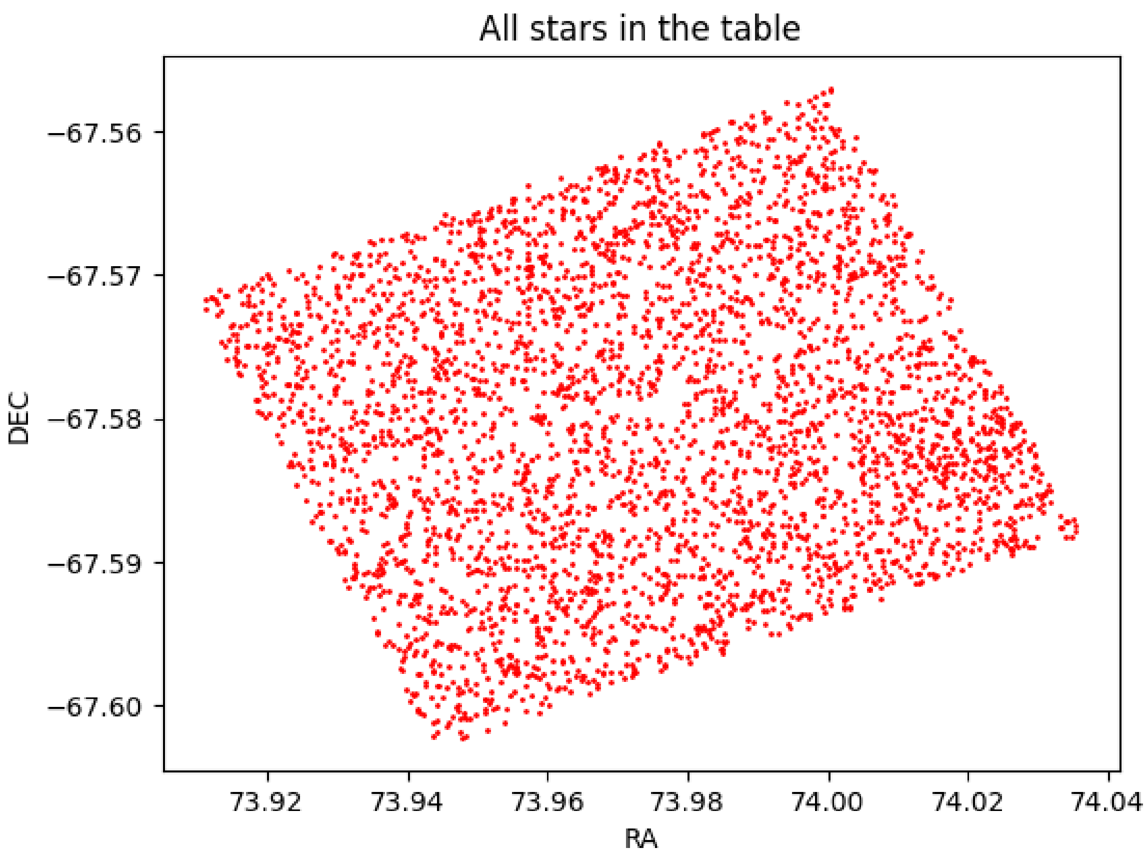
- The code reads a FITS file containing astronomical data.
- It extracts the RA and DEC of stars from the table and plots them in two ways:
  1. A default scatter plot.
  2. A scatter plot with smaller red markers for better distinction.
- The WCS object (`wcs`) is initialized but not used in this code. It could be used for coordinate transformations if needed.

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.wcs import WCS
from astropy.table import Table
from astropy.visualization import make_lupton_rgb
plt.ion()

hdul = fits.open('hst_results_nd.fits')


hdr = hdul[0].header
wcs = WCS(hdr)
data = hdul[0].data

t = Table.read('hst_results_nd.fits')

# check all stars in the table which are the first row of the table
ra = t[0]['RA']
dec = t[0]['DEC']
columns = t[0].colnames
print(columns)

# plot the table of all stars
plt.figure()
plt.plot(t['RA'], t['DEC'], 'o')
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table')
plt.show()

# plot the start with smaller red spots so that they can be distinguished
plt.figure()
plt.plot(t['RA'], t['DEC'], 'ro', ls= 'None', markersize = 1)
plt.xlabel('RA')
plt.ylabel('DEC')
plt.title('All stars in the table')
plt.show()
```

['Name', 'RA', 'DEC', 'HST_WFC3_F225W', 'HST_WFC3_F275W', 'HST_WFC3_F336W', 'HST_
WFC3_F475W', 'HST_WFC3_F814W', 'HST_WFC3_F110W', 'HST_WFC3_F160W', 'Av_Best', 'Av
_Exp', 'Av_p16', 'Av_p50', 'Av_p84', 'Rv_Best', 'Rv_Exp', 'Rv_p16', 'Rv_p50', 'Rv
_p84', 'Rv_A_Best', 'Rv_A_Exp', 'Rv_A_p16', 'Rv_A_p50', 'Rv_A_p84', 'f_A_Best',
'f_A_Exp', 'f_A_p16', 'f_A_p50', 'f_A_p84', 'distance_Best', 'distance_Exp', 'dis
tance_p16', 'distance_p50', 'distance_p84', 'radius_Best', 'radius_Exp', 'radius_
p16', 'radius_p50', 'radius_p84', 'logL_Best', 'logL_Exp', 'logL_p16', 'logL_p5
0', 'logL_p84', 'logg_Best', 'logg_Exp', 'logg_p16', 'logg_p50', 'logg_p84', 'mbo
lmag_Best', 'mbolmag_Exp', 'mbolmag_p16', 'mbolmag_p50', 'mbolmag_p84', 'logA_Bes
t', 'logA_Exp', 'logA_p16', 'logA_p50', 'logA_p84', 'logT_Best', 'logT_Exp', 'log
T_p16', 'logT_p50', 'logT_p84', 'M_ini_Best', 'M_ini_Exp', 'M_ini_p16', 'M_ini_p5
0', 'M_ini_p84', 'M_act_Best', 'M_act_Exp', 'M_act_p16', 'M_act_p50', 'M_act_p8
4', 'Z_Best', 'Z_Exp', 'Z_p16', 'Z_p50', 'Z_p84', 'logHST_WFC3_F225W_nd_Best', 'l
ogHST_WFC3_F225W_nd_Exp', 'logHST_WFC3_F225W_nd_p16', 'logHST_WFC3_F225W_nd_p50',
'logHST_WFC3_F225W_nd_p84', 'logHST_WFC3_F275W_nd_Best', 'logHST_WFC3_F275W_nd_Ex
p', 'logHST_WFC3_F275W_nd_p16', 'logHST_WFC3_F275W_nd_p50', 'logHST_WFC3_F275W_nd
_p84', 'logHST_WFC3_F336W_nd_Best', 'logHST_WFC3_F336W_nd_Exp', 'logHST_WFC3_F336
W_nd_p16', 'logHST_WFC3_F336W_nd_p50', 'logHST_WFC3_F336W_nd_p84', 'logHST_WFC3_F
475W_nd_Best', 'logHST_WFC3_F475W_nd_Exp', 'logHST_WFC3_F475W_nd_p16', 'logHST_WF
C3_F475W_nd_p50', 'logHST_WFC3_F475W_nd_p84', 'logHST_WFC3_F814W_nd_Best', 'logHS
T_WFC3_F814W_nd_Exp', 'logHST_WFC3_F814W_nd_p16', 'logHST_WFC3_F814W_nd_p50', 'lo
gHST_WFC3_F814W_nd_p84', 'logHST_WFC3_F110W_nd_Best', 'logHST_WFC3_F110W_nd_Exp',
'logHST_WFC3_F110W_nd_p16', 'logHST_WFC3_F110W_nd_p50', 'logHST_WFC3_F110W_nd_p8
4', 'logHST_WFC3_F160W_nd_Best', 'logHST_WFC3_F160W_nd_Exp', 'logHST_WFC3_F160W_n
d_p16', 'logHST_WFC3_F160W_nd_p50', 'logHST_WFC3_F160W_nd_p84', 'logGALEX_FUV_nd_
Best', 'logGALEX_FUV_nd_Exp', 'logGALEX_FUV_nd_p16', 'logGALEX_FUV_nd_p50', 'logG
ALEX_FUV_nd_p84', 'logF_QION228_nd_Best', 'logF_QION228_nd_Exp', 'logF_QION228_nd
_p16', 'logF_QION228_nd_p50', 'logF_QION228_nd_p84', 'logHST_WFC3_F225W_wd_Best',
'logHST_WFC3_F225W_wd_Exp', 'logHST_WFC3_F225W_wd_p16', 'logHST_WFC3_F225W_wd_p5
0', 'logHST_WFC3_F225W_wd_p84', 'logHST_WFC3_F275W_wd_Best', 'logHST_WFC3_F275W_w
d_Exp', 'logHST_WFC3_F275W_wd_p16', 'logHST_WFC3_F275W_wd_p50', 'logHST_WFC3_F275
W_wd_p84', 'logHST_WFC3_F336W_wd_Best', 'logHST_WFC3_F336W_wd_Exp', 'logHST_WFC3_
F336W_wd_p16', 'logHST_WFC3_F336W_wd_p50', 'logHST_WFC3_F336W_wd_p84', 'logHST_WF
C3_F475W_wd_Best', 'logHST_WFC3_F475W_wd_Exp', 'logHST_WFC3_F475W_wd_p16', 'logHS
T_WFC3_F475W_wd_p50', 'logHST_WFC3_F475W_wd_p84', 'logHST_WFC3_F814W_wd_Best', 'l
ogHST_WFC3_F814W_wd_Exp', 'logHST_WFC3_F814W_wd_p16', 'logHST_WFC3_F814W_wd_p50',
'logHST_WFC3_F814W_wd_p84', 'logHST_WFC3_F110W_wd_Best', 'logHST_WFC3_F110W_wd_Ex
p', 'logHST_WFC3_F110W_wd_p16', 'logHST_WFC3_F110W_wd_p50', 'logHST_WFC3_F110W_wd
_p84', 'logHST_WFC3_F160W_wd_Best', 'logHST_WFC3_F160W_wd_Exp', 'logHST_WFC3_F160
W_wd_p16', 'logHST_WFC3_F160W_wd_p50', 'logHST_WFC3_F160W_wd_p84', 'logGALEX_FUV_
wd_Best', 'logGALEX_FUV_wd_Exp', 'logGALEX_FUV_wd_p16', 'logGALEX_FUV_wd_p50', 'l
ogGALEX_FUV_wd_p84', 'logF_QION228_wd_Best', 'logF_QION228_wd_Exp', 'logF_QION228
_wd_p16', 'logF_QION228_wd_p50', 'logF_QION228_wd_p84', 'symlogHST_WFC3_F225W_wd_
bias_Best', 'symlogHST_WFC3_F225W_wd_bias_Exp', 'symlogHST_WFC3_F225W_wd_bias_p1
6', 'symlogHST_WFC3_F225W_wd_bias_p50', 'symlogHST_WFC3_F225W_wd_bias_p84', 'syml
ogHST_WFC3_F275W_wd_bias_Best', 'symlogHST_WFC3_F275W_wd_bias_Exp', 'symlogHST_WF
C3_F275W_wd_bias_p16', 'symlogHST_WFC3_F275W_wd_bias_p50', 'symlogHST_WFC3_F275W_
wd_bias_p84', 'symlogHST_WFC3_F336W_wd_bias_Best', 'symlogHST_WFC3_F336W_wd_bias_
Exp', 'symlogHST_WFC3_F336W_wd_bias_p16', 'symlogHST_WFC3_F336W_wd_bias_p50', 'sy
mlogHST_WFC3_F336W_wd_bias_p84', 'symlogHST_WFC3_F475W_wd_bias_Best', 'symlogHST_
WFC3_F475W_wd_bias_Exp', 'symlogHST_WFC3_F475W_wd_bias_p16', 'symlogHST_WFC3_F475
W_wd_bias_p50', 'symlogHST_WFC3_F475W_wd_bias_p84', 'symlogHST_WFC3_F814W_wd_bias
_Best', 'symlogHST_WFC3_F814W_wd_bias_Exp', 'symlogHST_WFC3_F814W_wd_bias_p16',
'symlogHST_WFC3_F814W_wd_bias_p50', 'symlogHST_WFC3_F814W_wd_bias_p84', 'symlogHS
T_WFC3_F110W_wd_bias_Best', 'symlogHST_WFC3_F110W_wd_bias_Exp', 'symlogHST_WFC3_F
110W_wd_bias_p16', 'symlogHST_WFC3_F110W_wd_bias_p50', 'symlogHST_WFC3_F110W_wd_b
ias_p84', 'symlogHST_WFC3_F160W_wd_bias_Best', 'symlogHST_WFC3_F160W_wd_bias_Ex
p', 'symlogHST_WFC3_F160W_wd_bias_p16', 'symlogHST_WFC3_F160W_wd_bias_p50', 'syml
ogHST_WFC3_F160W_wd_bias_p84', 'chi2min', 'Pmax', 'Pmax_indx', 'total_log_norm',
'best_gridsub_tag', 'reorder_tag', 'X', 'Y', 'F110W_RATE', 'F110W_RATERR', 'F110W

```
_VEGA', 'F110W_STD', 'F110W_ERR', 'F110W_CHI', 'F110W_SNR', 'F110W_SHARP', 'F110W
_ROUND', 'F110W_CROWD', 'F110W_FLAG', 'F160W_RATE', 'F160W_RATERR', 'F160W_VEGA',
'F160W_STD', 'F160W_ERR', 'F160W_CHI', 'F160W_SNR', 'F160W_SHARP', 'F160W_ROUND',
'F160W_CROWD', 'F160W_FLAG', 'F225W_RATE', 'F225W_RATERR', 'F225W_VEGA', 'F225W_S
TD', 'F225W_ERR', 'F225W_CHI', 'F225W_SNR', 'F225W_SHARP', 'F225W_ROUND', 'F225W_
CROWD', 'F225W_FLAG', 'F275W_RATE', 'F275W_RATERR', 'F275W_VEGA', 'F275W_STD', 'F
275W_ERR', 'F275W_CHI', 'F275W_SNR', 'F275W_SHARP', 'F275W_ROUND', 'F275W_CROWD',
'F275W_FLAG', 'F336W_RATE', 'F336W_RATERR', 'F336W_VEGA', 'F336W_STD', 'F336W_ER
R', 'F336W_CHI', 'F336W_SNR', 'F336W_SHARP', 'F336W_ROUND', 'F336W_CROWD', 'F336W
_FLAG', 'F475W_RATE', 'F475W_RATERR', 'F475W_VEGA', 'F475W_STD', 'F475W_ERR', 'F4
75W_CHI', 'F475W_SNR', 'F475W_SHARP', 'F475W_ROUND', 'F475W_CROWD', 'F475W_FLAG',
'F814W_RATE', 'F814W_RATERR', 'F814W_VEGA', 'F814W_STD', 'F814W_ERR', 'F814W_CH
I', 'F814W_SNR', 'F814W_SHARP', 'F814W_ROUND', 'F814W_CROWD', 'F814W_FLAG', 'HST_
WFC3_F336W_MD', 'reliable']
```





Explanation of the Code

1. **Extract Relevant Data**:
    - `age = t['logA_p50']` : Extracts the logarithmic age of stars (logarithm of age in years).
    - `mass = t['M_ini_p50']` : Extracts the initial mass of stars.
    - `f475w = t['F475W_VEGA']` : Extracts the F475W magnitude in the Vega system.
    - `f814w = t['F814W_VEGA']` : Extracts the F814W magnitude in the Vega system.
2. **Filter Stars by Age**:
    - `young_stars = t[age < 8.5]` : Filters stars with log(age) < 8.5, representing young stars.
    - `old_stars = t[age > 10]` : Filters stars with log(age) > 10, representing old stars.
3. **Create a Color-Magnitude Diagram (CMD)**:
    - `plt.figure()` : Creates a new figure for the plot.
    - `plt.plot(f475w - f814w, f814w, '.', markersize=1, label='All stars', color='gray')` :
        - Plots all stars in gray, using the color index (F475W - F814W) on the x-axis and the F814W magnitude on the y-axis.
    - `plt.ylim(26.5, 17.5)` : Sets the y-axis limits, inverted to show brighter stars at the top.

4. **Plot Young and Old Stars**:

   - `plt.plot(young_stars['F475W_VEGA'] - young_stars['F814W_VEGA'], young_stars['F814W_VEGA'], 'bo', markersize=1, label='Young stars')` :
     - Plots young stars in blue, using their color index and magnitude.
   - `plt.plot(old_stars['F475W_VEGA'] - old_stars['F814W_VEGA'], old_stars['F814W_VEGA'], 'ro', markersize=1, label='Old stars')` :
     - Plots old stars in red, using their color index and magnitude.

5. **Add Labels and Legend**:

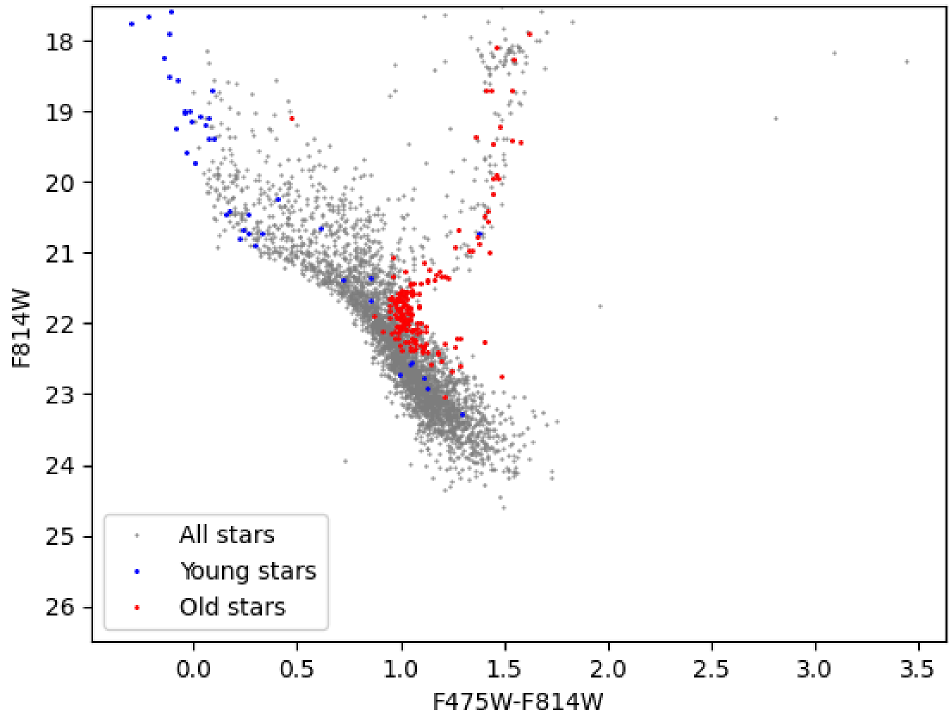
   - `plt.xlabel('F475W-F814W')` : Labels the x-axis as "F475W-F814W" (color index).
   - `plt.ylabel('F814W')` : Labels the y-axis as "F814W" (magnitude).
   - `plt.legend()` : Adds a legend to distinguish between all stars, young stars, and old stars.

6. **Display and Save the Plot**:

   - `plt.show()` : Displays the plot.
   - `plt.savefig('cmd_by_age.png')` : Saves the plot as an image file named `cmd_by_age.png` .

Output:

- A **color-magnitude diagram (CMD)** is generated:
  - **Gray points**: Represent all stars in the dataset.
  - **Blue points**: Represent young stars (log(age) < 8.5).
  - **Red points**: Represent old stars (log(age) > 10).
- The x-axis represents the color index (F475W - F814W), and the y-axis represents the F814W magnitude (inverted for brighter stars).
- The plot is saved as `cmd_by_age.png` .\



Use Case:

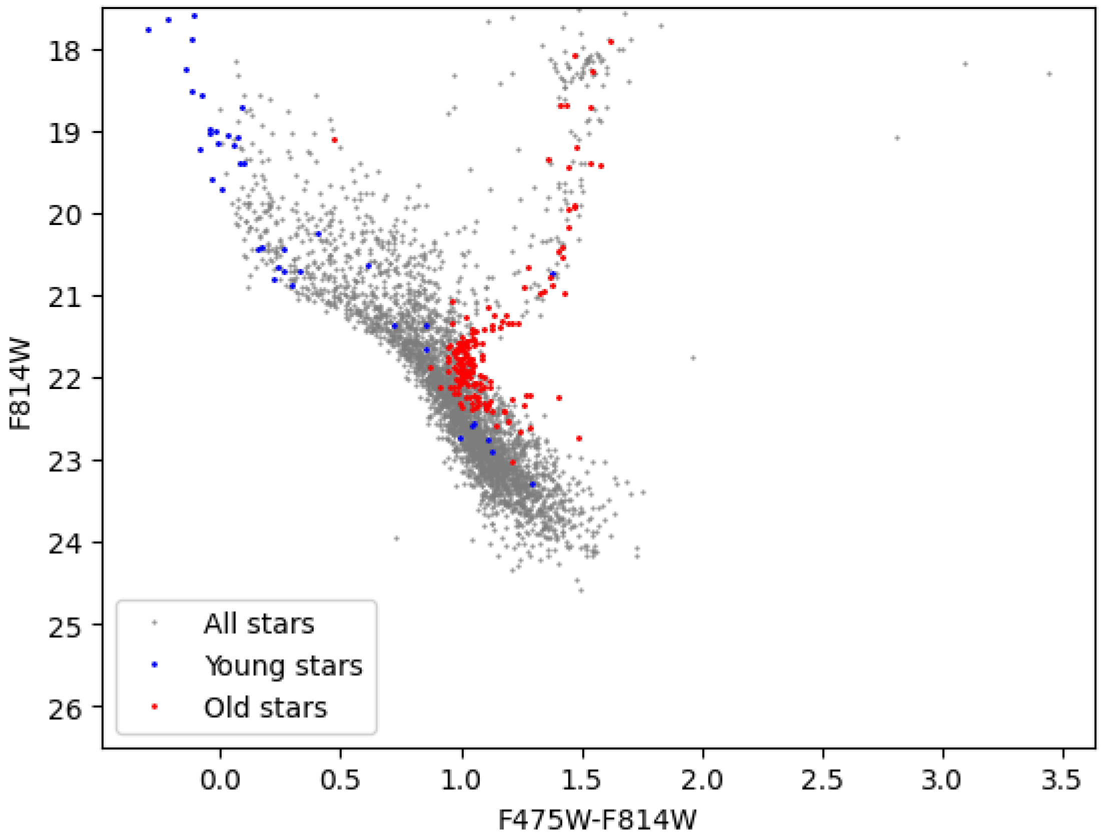
This CMD is useful for analyzing the stellar population based on age:

- Young stars (blue) and old stars (red) are highlighted for comparison.
- The diagram helps identify trends in color and brightness for different age groups.

In [9]:

```
# Extract relevant columns from the table
age = t['logA_p50']  # Logarithmic age of stars (logarithm of age in years)
mass = t['M_ini_p50']  # Initial mass of stars
f475w = t['F475W_VEGA']  # F475W magnitude in the Vega system
f814w = t['F814W_VEGA']  # F814W magnitude in the Vega system

# Filter stars based on age
young_stars = t[age < 8.5]   # Stars with log(age) < 8.5 (young stars)
old_stars = t[age > 10]  # Stars with log(age) > 10 (old stars)

# Create a color-magnitude diagram (CMD)
plt.figure()

# Plot all stars in gray
plt.plot(f475w - f814w, f814w, '.', markersize=1, label='All stars', color='gray

# Set y-axis limits (inverted for brighter stars)
plt.ylim(26.5, 17.5)

# Plot young stars in blue
plt.plot(
    young_stars['F475W_VEGA'] - young_stars['F814W_VEGA'],  # Color (F475W - F81
    young_stars['F814W_VEGA'],  # Magnitude (F814W)
    'bo', markersize=1, label='Young stars'
)
```

```python
# Plot old stars in red
plt.plot(
    old_stars['F475W_VEGA'] - old_stars['F814W_VEGA'],  # Color (F475W - F814W)
    old_stars['F814W_VEGA'],  # Magnitude (F814W)
    'ro', markersize=1, label='Old stars'
)

# Label the axes
plt.xlabel('F475W-F814W')  # X-axis: Color index
plt.ylabel('F814W')  # Y-axis: Magnitude

# Add a legend to distinguish between all stars, young stars, and old stars
plt.legend()
# Save the plot as an image file
plt.savefig('cmd_by_age.png')
# Display the plot
plt.show()
```



In [ ]: