Карпов Даниил Константинович, ИУ5-61Б
Вариант №10: номер задачи - 2; номер набора данных - 2.

In [10]:

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
import seaborn as sns
import matplotlib.pyplot as plt
from pylab import rcParams # для того, чтобы задавать размер диаграмм
%matplotlib inline
```

In [11]:

```
data = pd.read_csv('/Users/dankarpov/Desktop/archive/dc-wikia-data.csv', sep=',')
```

In [12]:

```
data.head()
```

Out[12]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVI
0	1422	Batman (Bruce Wayne)	\/wiki\/Batman_(Bruce_Wayne)	Secret Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Character
1	23387	Superman (Clark Kent)	\/wiki\/Superman_(Clark_Kent)	Secret Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Character
2	1458	Green Lantern (Hal Jordan)	\/wiki\/Green_Lantern_(Hal_Jordan)	Secret Identity	Good Characters	Brown Eyes	Brown Hair	Male Characters	NaN	Living Character
3	1659	James Gordon (New Earth)	\/wiki\/James_Gordon_(New_Earth)	Public Identity	Good Characters	Brown Eyes	White Hair	Male Characters	NaN	Living Character
4	1576	Richard Grayson (New Earth)	\/wiki\/Richard_Grayson_(New_Earth)	Secret Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Character

In [13]:

```
data.isnull().sum()
```

Out[13]:

```
page_id                0
name                   0
urlslug                0
ID                  2013
ALIGN                601
EYE                 3628
HAIR                2274
SEX                  125
GSM                 6832
ALIVE                  3
APPEARANCES          355
FIRST APPEARANCE      69
```

```
YEAR                 69
dtype: int64
```

In [14]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6896 entries, 0 to 6895
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   page_id           6896 non-null   int64  
 1   name              6896 non-null   object 
 2   urlslug           6896 non-null   object 
 3   ID                4883 non-null   object 
 4   ALIGN             6295 non-null   object 
 5   EYE               3268 non-null   object 
 6   HAIR              4622 non-null   object 
 7   SEX               6771 non-null   object 
 8   GSM               64 non-null     object 
 9   ALIVE             6893 non-null   object 
 10  APPEARANCES       6541 non-null   float64
 11  FIRST APPEARANCE  6827 non-null   object 
 12  YEAR              6827 non-null   float64
dtypes: float64(2), int64(1), object(10)
memory usage: 700.5+ KB
```

In [15]:

```
mising_count = data.isnull().sum()
all_count = data.isnull().count()
pd.concat([mising_count.sort_values(), (mising_count/all_count*100).sort_values()],
          axis=1, keys=['Количество пропусков', 'Процент пропусков']).tail(11)
```

Out[15]:

	Количество пропусков	Процент пропусков
urlslug	0	0.000000
ALIVE	3	0.043503
FIRST APPEARANCE	69	1.000580
YEAR	69	1.000580
SEX	125	1.812645
APPEARANCES	355	5.147912
ALIGN	601	8.715197
ID	2013	29.190835
HAIR	2274	32.975638
EYE	3628	52.610209
GSM	6832	99.071926

Обработка пропусков для категориального признака **"GSM"**

Выполним удаление данного признака так как отстутствуют **99%** данных

In [16]:

```
data.drop(['GSM'], axis=1, inplace=True)
```

Обработка пропусков для **"APPEARANCES"**

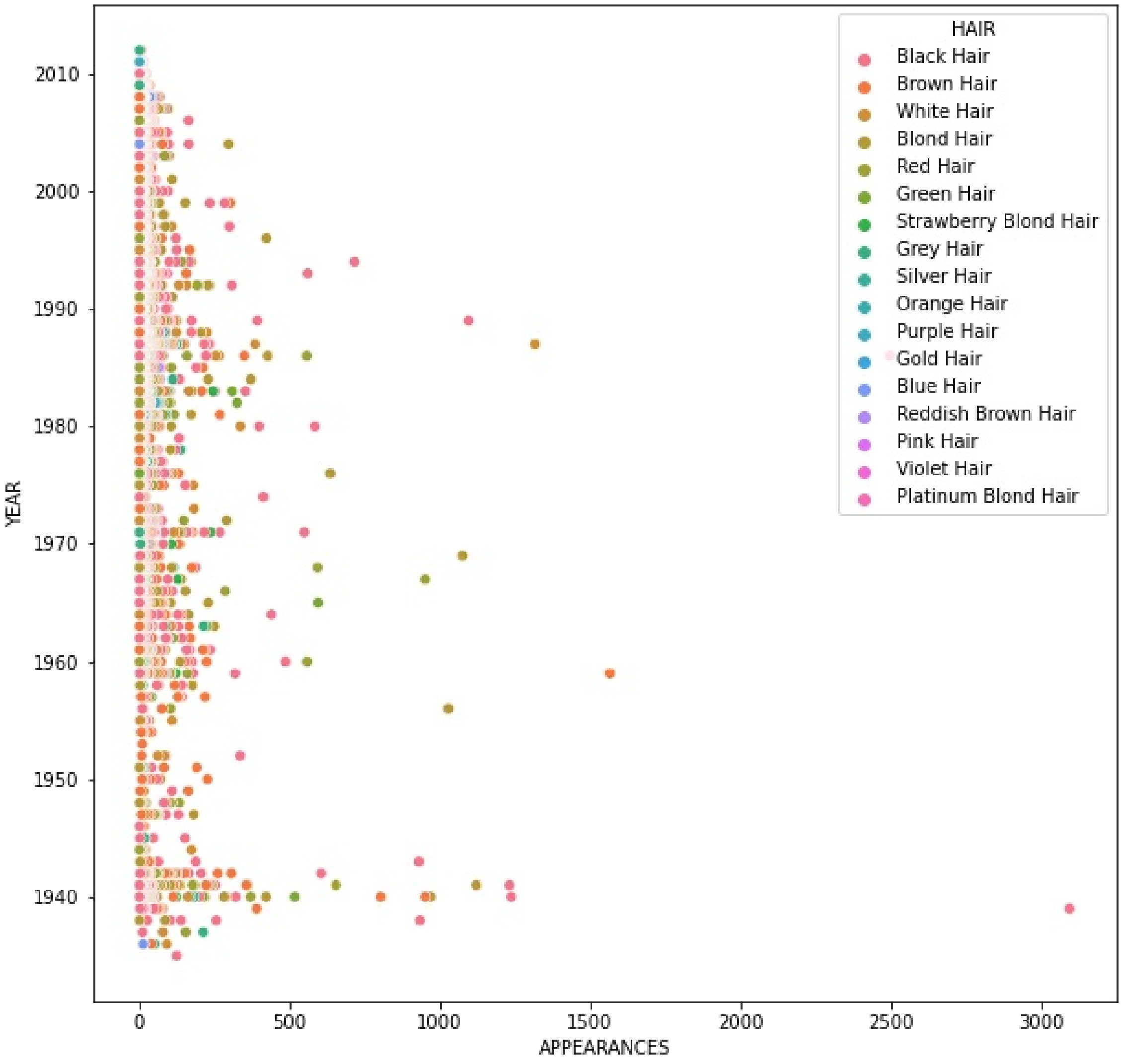
Заполненим этот признак так как пропуски незначительные (всего **5%)**

In [21]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x = "APPEARANCES",y = "YEAR", data=data, hue='HAIR')
```

Out[21]:

```
<AxesSubplot:xlabel='APPEARANCES', ylabel='YEAR'>
```



Для заполнения будем использовать моду "Наиболее вероятный":

In [20]:

```
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data[['APPEARANCES']])
imp_num = SimpleImputer(strategy='most_frequent')
data_num_imp = imp_num.fit_transform(data[['APPEARANCES']])
data['APPEARANCES'] = data_num_imp
filled_data = data_num_imp[mask_missing_values_only]
print('APPEARANCES', 'most_frequent', filled_data.size, filled_data[0], filled_data[fill
ed_data.size-1], sep='; ')
```

```
APPEARANCES; most_frequent; 355; 1.0; 1.0
```

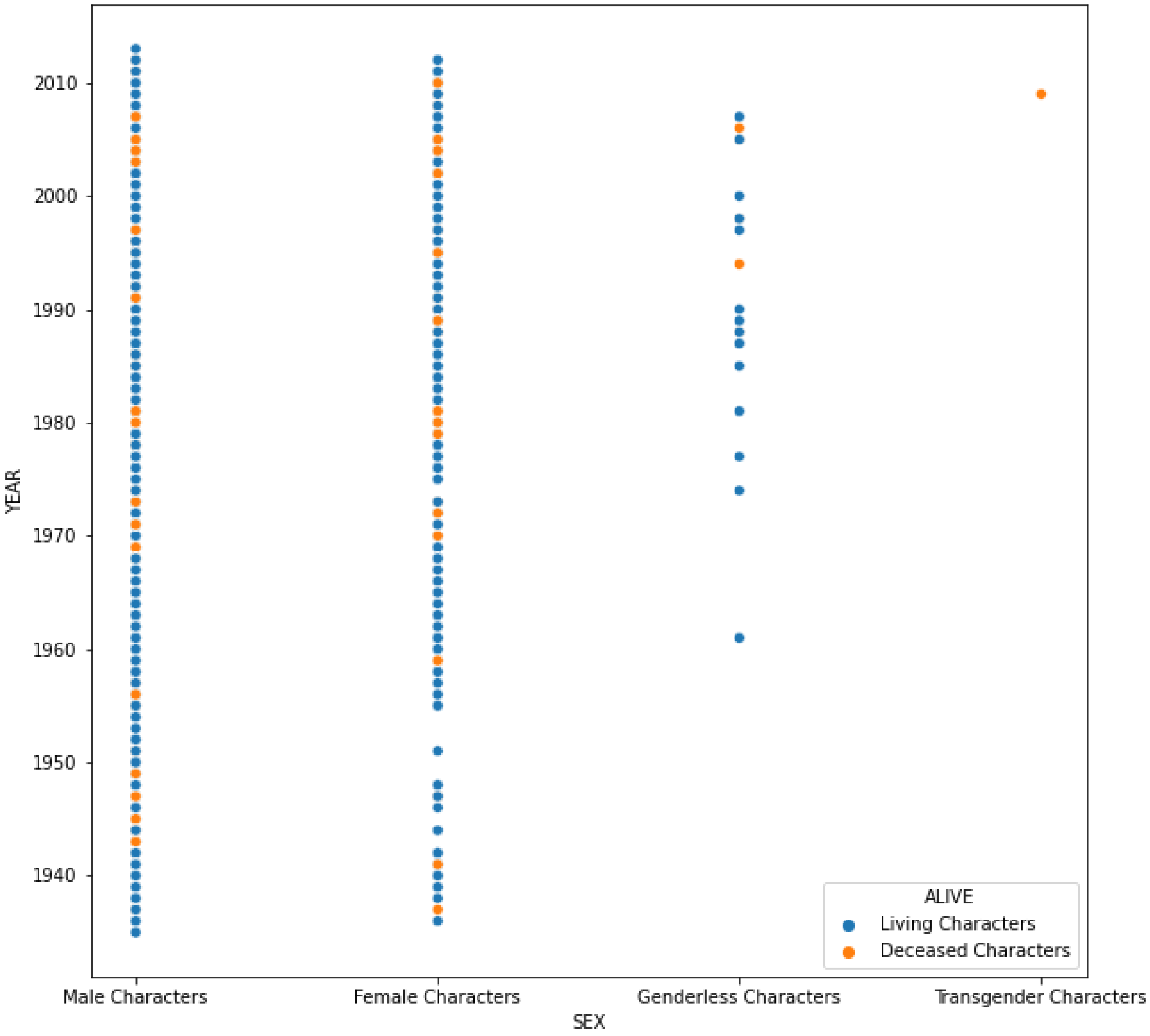
Еще один графичек чтобы был)))

In [26]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x = "SEX",y = "YEAR", data=data, hue='ALIVE')
```

Out[26]:

```
<AxesSubplot:xlabel='SEX', ylabel='YEAR'>
```



Итоговый вид датасета после обработки пропусков в двух признаках

In [27]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6896 entries, 0 to 6895
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   page_id           6896 non-null   int64  
 1   name              6896 non-null   object 
 2   urlslug           6896 non-null   object 
 3   ID                4883 non-null   object 
 4   ALIGN             6295 non-null   object 
 5   EYE               3268 non-null   object 
 6   HAIR              4622 non-null   object 
 7   SEX               6771 non-null   object 
 8   ALIVE             6893 non-null   object 
 9   APPEARANCES       6896 non-null   float64
 10  FIRST APPEARANCE  6827 non-null   object 
 11  YEAR              6827 non-null   float64
dtypes: float64(2), int64(1), object(9)
memory usage: 646.6+ KB
```

In [ ]: