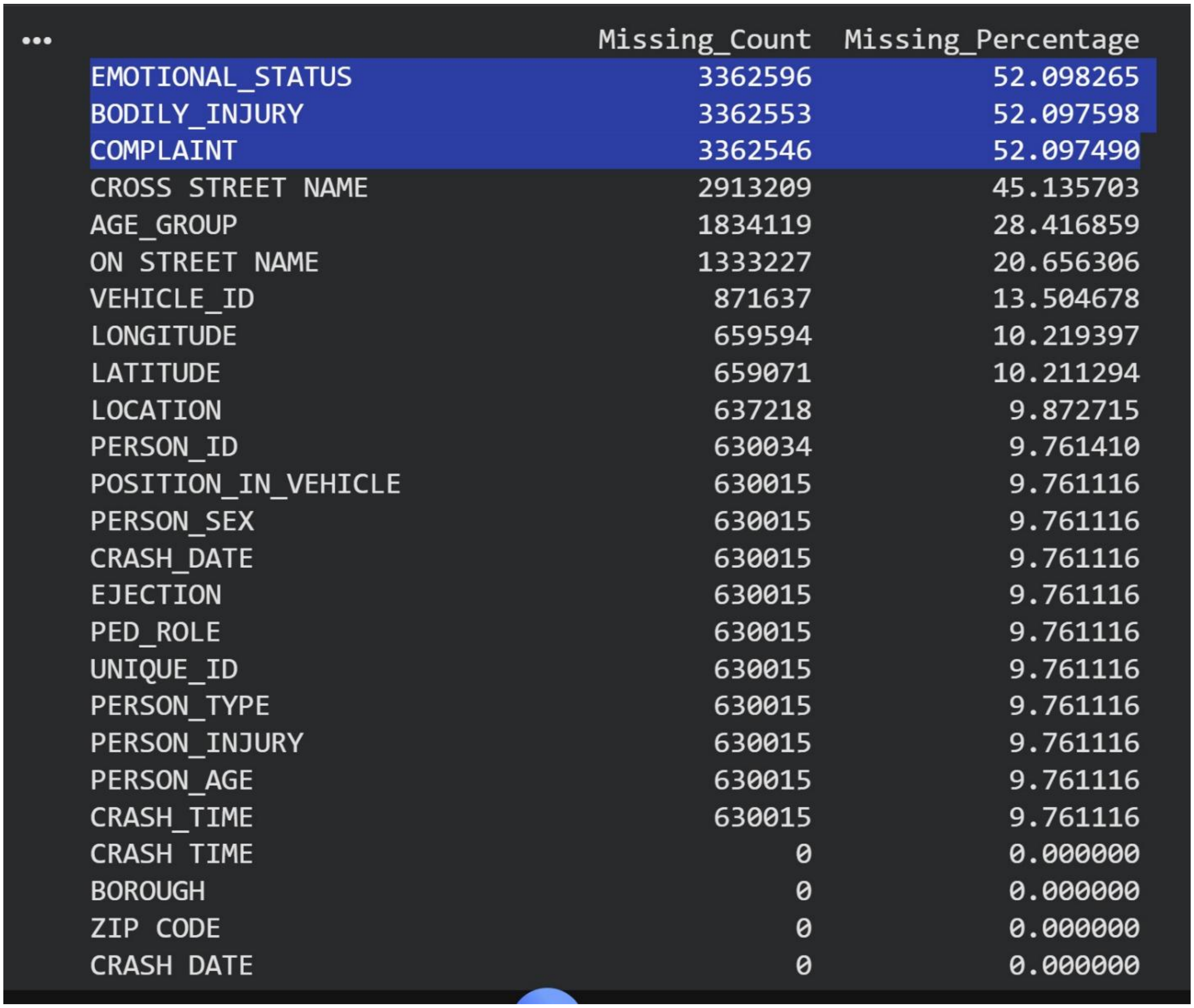
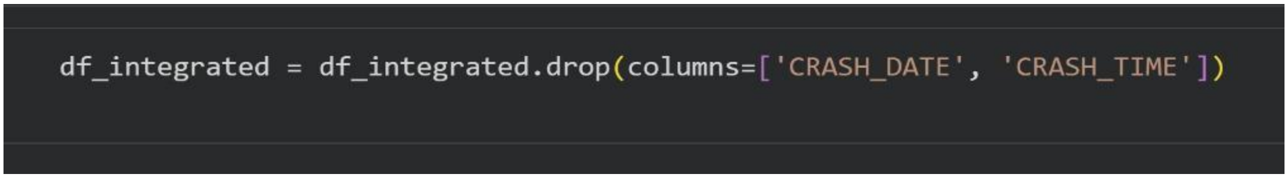
# Post-Integration cleaning



```
•••                          Missing_Count  Missing_Percentage
    EMOTIONAL_STATUS               3362596           52.098265
    BODILY_INJURY                  3362553           52.097598
    COMPLAINT                      3362546           52.097490
    CROSS STREET NAME              2913209           45.135703
    AGE_GROUP                      1834119           28.416859
    ON STREET NAME                 1333227           20.656306
    VEHICLE_ID                      871637           13.504678
    LONGITUDE                       659594           10.219397
    LATITUDE                        659071           10.211294
    LOCATION                        637218            9.872715
    PERSON_ID                       630034            9.761410
    POSITION_IN_VEHICLE             630015            9.761116
    PERSON_SEX                      630015            9.761116
    CRASH_DATE                      630015            9.761116
    EJECTION                        630015            9.761116
    PED_ROLE                        630015            9.761116
    UNIQUE_ID                       630015            9.761116
    PERSON_TYPE                     630015            9.761116
    PERSON_INJURY                   630015            9.761116
    PERSON_AGE                      630015            9.761116
    CRASH_TIME                      630015            9.761116
    CRASH TIME                           0            0.000000
    BOROUGH                              0            0.000000
    ZIP CODE                             0            0.000000
    CRASH DATE                           0            0.000000
```

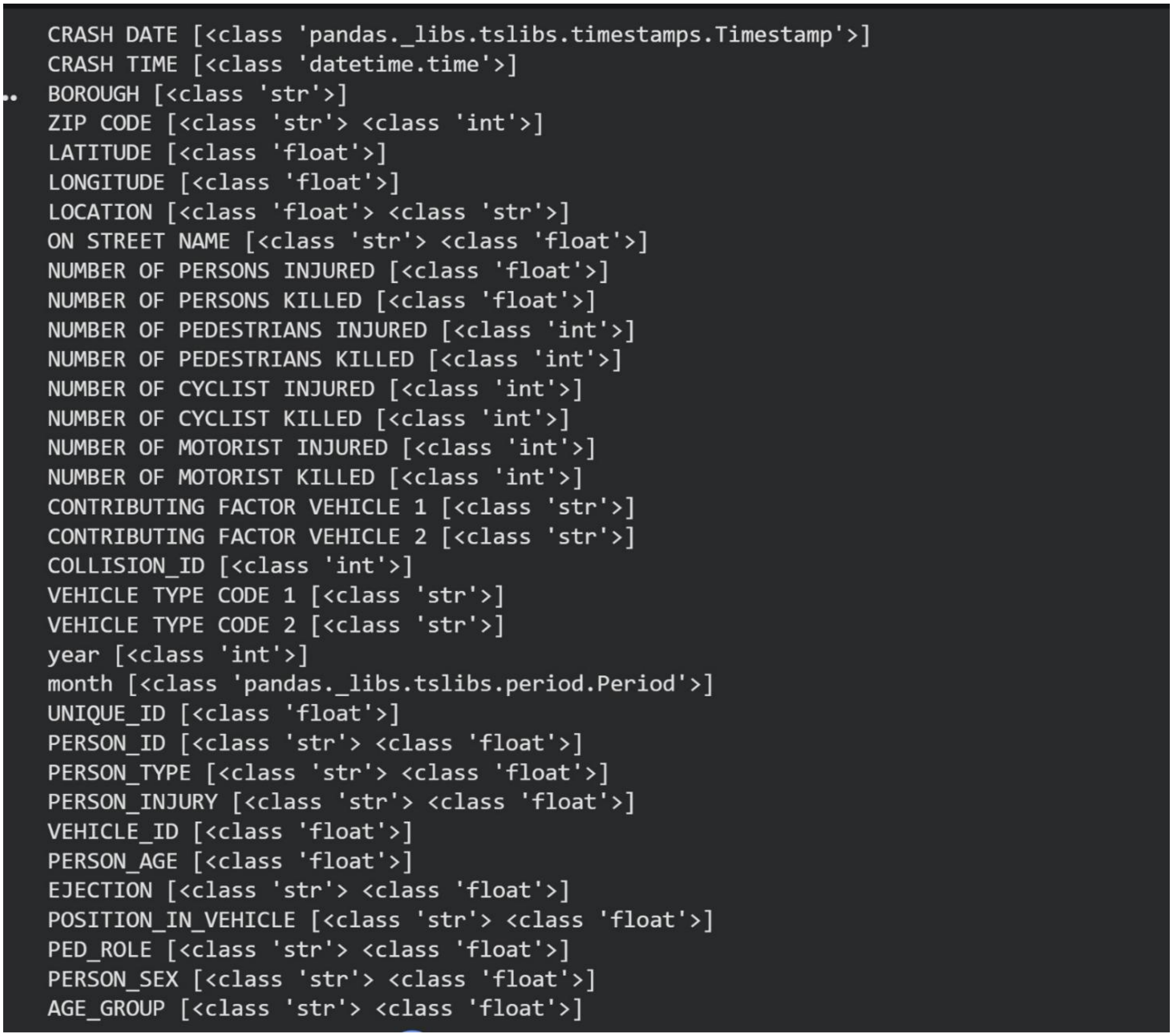
**We dropped EMOTIONAL_STATUS, BODILY_INJURY, and COMPLAINT because each had over 50% missing values. Retaining them would weaken analysis and introduce bias, with no**

reliable imputation strategy available. And we will drop the cross street name too as the percentage if missing values in it is high and this column will not provide informative insights.



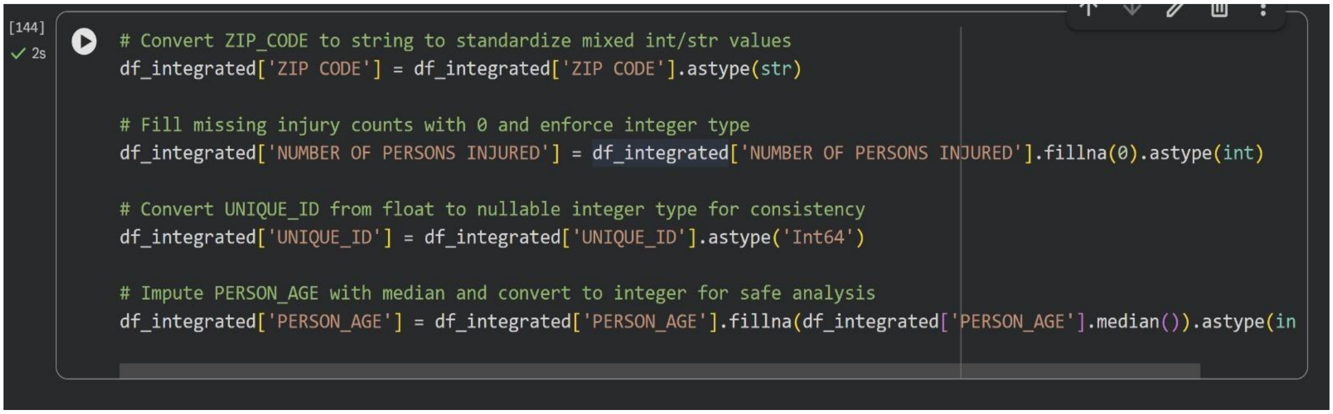
```
df_integrated = df_integrated.drop(columns=['CRASH_DATE', 'CRASH_TIME'])
```

We dropped these 2 columns from the integrated table (table resulted from joining the 2 datasets) as they were redundant and there are other 2 columns that represents that same data.



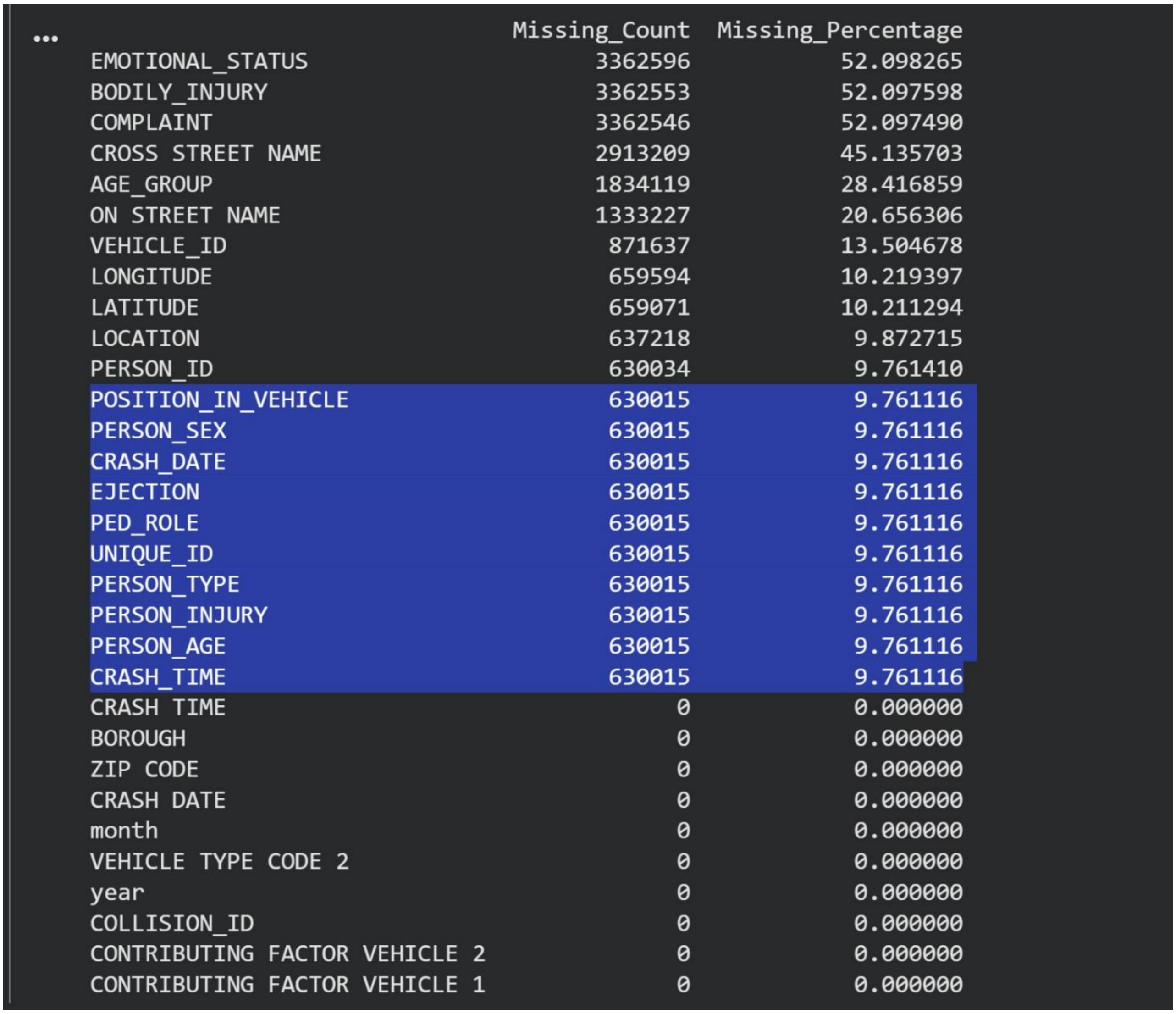
```
CRASH DATE [<class 'pandas._libs.tslibs.timestamps.Timestamp'>]
CRASH TIME [<class 'datetime.time'>]
BOROUGH [<class 'str'>]
ZIP CODE [<class 'str'> <class 'int'>]
LATITUDE [<class 'float'>]
LONGITUDE [<class 'float'>]
LOCATION [<class 'float'> <class 'str'>]
ON STREET NAME [<class 'str'> <class 'float'>]
NUMBER OF PERSONS INJURED [<class 'float'>]
NUMBER OF PERSONS KILLED [<class 'float'>]
NUMBER OF PEDESTRIANS INJURED [<class 'int'>]
NUMBER OF PEDESTRIANS KILLED [<class 'int'>]
NUMBER OF CYCLIST INJURED [<class 'int'>]
NUMBER OF CYCLIST KILLED [<class 'int'>]
NUMBER OF MOTORIST INJURED [<class 'int'>]
NUMBER OF MOTORIST KILLED [<class 'int'>]
CONTRIBUTING FACTOR VEHICLE 1 [<class 'str'>]
CONTRIBUTING FACTOR VEHICLE 2 [<class 'str'>]
COLLISION_ID [<class 'int'>]
VEHICLE TYPE CODE 1 [<class 'str'>]
VEHICLE TYPE CODE 2 [<class 'str'>]
year [<class 'int'>]
month [<class 'pandas._libs.tslibs.period.Period'>]
UNIQUE_ID [<class 'float'>]
PERSON_ID [<class 'str'> <class 'float'>]
PERSON_TYPE [<class 'str'> <class 'float'>]
PERSON_INJURY [<class 'str'> <class 'float'>]
VEHICLE_ID [<class 'float'>]
PERSON_AGE [<class 'float'>]
EJECTION [<class 'str'> <class 'float'>]
POSITION_IN_VEHICLE [<class 'str'> <class 'float'>]
PED_ROLE [<class 'str'> <class 'float'>]
PERSON_SEX [<class 'str'> <class 'float'>]
AGE_GROUP [<class 'str'> <class 'float'>]
```

# Identified mixed data types across key columns (e.g., ZIP_CODE, LOCATION, PERSON_ID) requiring standardization before analysis.



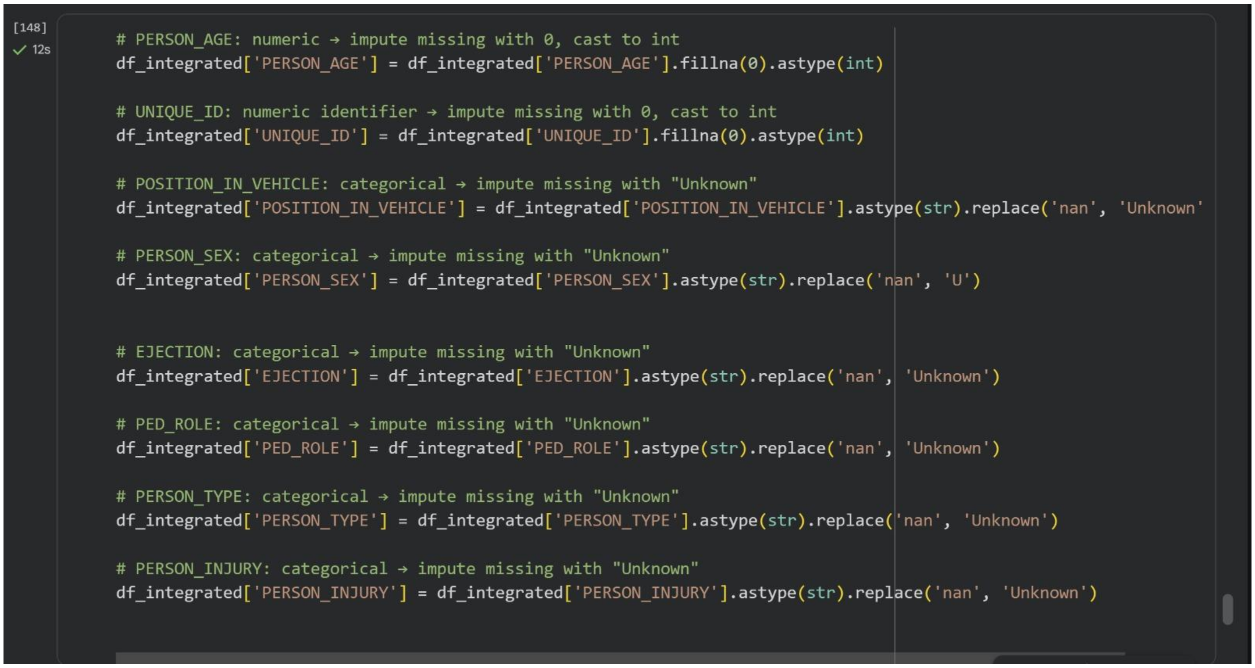
```
# Convert ZIP_CODE to string to standardize mixed int/str values
df_integrated['ZIP CODE'] = df_integrated['ZIP CODE'].astype(str)

# Fill missing injury counts with 0 and enforce integer type
df_integrated['NUMBER OF PERSONS INJURED'] = df_integrated['NUMBER OF PERSONS INJURED'].fillna(0).astype(int)

# Convert UNIQUE_ID from float to nullable integer type for consistency
df_integrated['UNIQUE_ID'] = df_integrated['UNIQUE_ID'].astype('Int64')

# Impute PERSON_AGE with median and convert to integer for safe analysis
df_integrated['PERSON_AGE'] = df_integrated['PERSON_AGE'].fillna(df_integrated['PERSON_AGE'].median()).astype(in
```

# The justification is in the comments.



```
...                               Missing_Count  Missing_Percentage
    EMOTIONAL_STATUS                    3362596           52.098265
    BODILY_INJURY                       3362553           52.097598
    COMPLAINT                           3362546           52.097490
    CROSS STREET NAME                   2913209           45.135703
    AGE_GROUP                           1834119           28.416859
    ON STREET NAME                      1333227           20.656306
    VEHICLE_ID                           871637           13.504678
    LONGITUDE                            659594           10.219397
    LATITUDE                             659071           10.211294
    LOCATION                             637218            9.872715
    PERSON_ID                            630034            9.761410
    POSITION_IN_VEHICLE                  630015            9.761116
    PERSON_SEX                           630015            9.761116
    CRASH_DATE                           630015            9.761116
    EJECTION                             630015            9.761116
    PED_ROLE                             630015            9.761116
    UNIQUE_ID                            630015            9.761116
    PERSON_TYPE                          630015            9.761116
    PERSON_INJURY                        630015            9.761116
    PERSON_AGE                           630015            9.761116
    CRASH_TIME                           630015            9.761116
    CRASH TIME                                0            0.000000
    BOROUGH                                   0            0.000000
    ZIP CODE                                  0            0.000000
    CRASH DATE                                0            0.000000
    month                                     0            0.000000
    VEHICLE TYPE CODE 2                       0            0.000000
    year                                      0            0.000000
    COLLISION_ID                              0            0.000000
    CONTRIBUTING FACTOR VEHICLE 2             0            0.000000
    CONTRIBUTING FACTOR VEHICLE 1             0            0.000000
```

**We observed that there are 6300015 crash records with no corresponding person records, so we filled these rows with ”Unknown” for categorical and 0 for numerical.**



```
# PERSON_AGE: numeric → impute missing with 0, cast to int
df_integrated['PERSON_AGE'] = df_integrated['PERSON_AGE'].fillna(0).astype(int)

# UNIQUE_ID: numeric identifier → impute missing with 0, cast to int
df_integrated['UNIQUE_ID'] = df_integrated['UNIQUE_ID'].fillna(0).astype(int)

# POSITION_IN_VEHICLE: categorical → impute missing with "Unknown"
df_integrated['POSITION_IN_VEHICLE'] = df_integrated['POSITION_IN_VEHICLE'].astype(str).replace('nan', 'Unknown'

# PERSON_SEX: categorical → impute missing with "Unknown"
df_integrated['PERSON_SEX'] = df_integrated['PERSON_SEX'].astype(str).replace('nan', 'U')


# EJECTION: categorical → impute missing with "Unknown"
df_integrated['EJECTION'] = df_integrated['EJECTION'].astype(str).replace('nan', 'Unknown')

# PED_ROLE: categorical → impute missing with "Unknown"
df_integrated['PED_ROLE'] = df_integrated['PED_ROLE'].astype(str).replace('nan', 'Unknown')

# PERSON_TYPE: categorical → impute missing with "Unknown"
df_integrated['PERSON_TYPE'] = df_integrated['PERSON_TYPE'].astype(str).replace('nan', 'Unknown')

# PERSON_INJURY: categorical → impute missing with "Unknown"
df_integrated['PERSON_INJURY'] = df_integrated['PERSON_INJURY'].astype(str).replace('nan', 'Unknown')
```