Here’s how you can connect all your datasets into one table (Product) in SQLite for your Django project **"URBAN CART"**. We'll use separate models for main\_category and sub\_category and relate them using foreign keys.

**Steps to Implement**

**1. Set Up the Models**

In your Django app (e.g., urban\_cart), define the models for MainCategory, SubCategory, and Product.

python

CopyEdit

from django.db import models

class MainCategory(models.Model):

name = models.CharField(max\_length=50, unique=True)

def \_\_str\_\_(self):

return self.name

class SubCategory(models.Model):

name = models.CharField(max\_length=50, unique=True)

main\_category = models.ForeignKey(MainCategory, on\_delete=models.CASCADE, related\_name="subcategories")

def \_\_str\_\_(self):

return f"{self.name} ({self.main\_category.name})"

class Product(models.Model):

name = models.CharField(max\_length=255)

main\_category = models.ForeignKey(MainCategory, on\_delete=models.CASCADE)

sub\_category = models.ForeignKey(SubCategory, on\_delete=models.CASCADE)

image\_url = models.URLField()

site\_link = models.URLField()

ratings = models.FloatField(null=True, blank=True)

no\_of\_ratings = models.PositiveIntegerField(null=True, blank=True)

discount\_price = models.DecimalField(max\_digits=10, decimal\_places=2)

actual\_price = models.DecimalField(max\_digits=10, decimal\_places=2)

def \_\_str\_\_(self):

return self.name

**2. Run Migrations**

After defining the models, make migrations and apply them to your database.

bash

CopyEdit

python manage.py makemigrations

python manage.py migrate

**3. Process Your Datasets**

You can use Python (with **Pandas**) to process and insert your CSV data into the database.

**Step 1: Install Pandas**

If not installed, install it using:

bash

CopyEdit

pip install pandas

**Step 2: Write a Script to Populate the Database**

Create a script (import\_data.py) in your Django app directory to read the datasets and populate the database.

python

CopyEdit

import os

import pandas as pd

from urban\_cart.models import MainCategory, SubCategory, Product

# Path to your datasets

DATASETS = [

"Casual Shoes.csv",

"Formal Shoes.csv",

"Sports Shoes.csv",

"Shoes.csv",

"Kids Shoes.csv",

]

def clean\_price(price):

"""Clean the price string and convert to float."""

return float(price.replace('â‚¹', '').replace(',', '').strip())

def import\_data():

for dataset in DATASETS:

# Load the dataset

data = pd.read\_csv(dataset)

for \_, row in data.iterrows():

# Get or create MainCategory

main\_category, \_ = MainCategory.objects.get\_or\_create(name=row['main\_category'])

# Get or create SubCategory

sub\_category, \_ = SubCategory.objects.get\_or\_create(name=row['sub\_category'], main\_category=main\_category)

# Create Product

Product.objects.create(

name=row['name'],

main\_category=main\_category,

sub\_category=sub\_category,

image\_url=row['image\_url'],

site\_link=row['site\_link'],

ratings=row['ratings'] if not pd.isna(row['ratings']) else None,

no\_of\_ratings=row['no\_of\_ratings'] if not pd.isna(row['no\_of\_ratings']) else None,

discount\_price=clean\_price(row['discount\_price']),

actual\_price=clean\_price(row['actual\_price']),

)

# Call the import function

if \_\_name\_\_ == "\_\_main\_\_":

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'urban\_cart.settings')

import django

django.setup()

import\_data()

print("Data imported successfully!")

Run the script:

bash

CopyEdit

python import\_data.py

**4. Create the Template**

Now, create a template to display the products.

**View Function**

In views.py, create a function to fetch and render products:

python

CopyEdit

from django.shortcuts import render

from .models import Product

def product\_list(request):

products = Product.objects.all()

return render(request, 'product\_list.html', {'products': products})

**Template (product\_list.html)**

Create a template to display the products:

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Product List</title>

</head>

<body>

<h1>Products</h1>

<ul>

{% for product in products %}

<li>

<h3>{{ product.name }}</h3>

<p>Main Category: {{ product.main\_category.name }}</p>

<p>Sub Category: {{ product.sub\_category.name }}</p>

<p>Price: ₹{{ product.discount\_price }} (Actual: ₹{{ product.actual\_price }})</p>

<p>Ratings: {{ product.ratings }} ({{ product.no\_of\_ratings }} reviews)</p>

<p><a href="{{ product.site\_link }}" target="\_blank">View on site</a></p>

<img src="{{ product.image\_url }}" alt="{{ product.name }}" width="150">

</li>

{% endfor %}

</ul>

</body>

</html>

**Add URL for the View**

In urls.py, add a route for the product list view:

python

CopyEdit

from django.urls import path

from . import views

urlpatterns = [

path('products/', views.product\_list, name='product\_list'),

]

**5. Test the Application**

1. Start your Django development server:

bash

CopyEdit

python manage.py runserver

1. Visit the http://127.0.0.1:8000/products/ URL to see your products displayed.

**Summary**

* The models MainCategory and SubCategory are related using a foreign key.
* All datasets are combined into a single Product table.
* External URLs for images and site links are used in the template for display.
* Prices are cleaned and stored as decimal values for proper formatting.

In your project, you can store the datasets (e.g., Casual Shoes.csv, Formal Shoes.csv, etc.) in a directory within your Django project for easy access. A common location is to create a folder called datasets in the root of your Django app or project.

Here’s the typical structure of your Django project with the datasets included:

bash

CopyEdit

urban\_cart/ # Django project directory

├── urban\_cart/ # Main project folder

│ ├── settings.py # Django settings file

│ ├── urls.py # Project-level URL configuration

│ └── ...

├── datasets/ # Directory to store your datasets

│ ├── Casual Shoes.csv

│ ├── Formal Shoes.csv

│ ├── Sports Shoes.csv

│ ├── Shoes.csv

│ └── Kids Shoes.csv

├── app\_name/ # Django app folder (replace `app\_name` with your actual app name)

│ ├── models.py # App models

│ ├── views.py # App views

│ ├── templates/ # HTML templates

│ └── ...

├── manage.py # Django management script

└── db.sqlite3 # SQLite database

**How to Use the Datasets in Your Script**

1. Save the datasets in the datasets/ directory.
2. Update the file paths in your import\_data.py script to reflect the location of the datasets.

**Updated Script Example:**

python

CopyEdit

import os

import pandas as pd

from app\_name.models import MainCategory, SubCategory, Product

# Define the path to the datasets directory

DATASETS\_DIR = os.path.join(os.path.dirname(\_\_file\_\_), '../datasets')

# List of dataset filenames

DATASETS = [

"Casual Shoes.csv",

"Formal Shoes.csv",

"Sports Shoes.csv",

"Shoes.csv",

"Kids Shoes.csv",

]

def clean\_price(price):

"""Clean the price string and convert to float."""

return float(price.replace('â‚¹', '').replace(',', '').strip())

def import\_data():

for dataset in DATASETS:

dataset\_path = os.path.join(DATASETS\_DIR, dataset) # Get the full path

data = pd.read\_csv(dataset\_path) # Load the dataset

for \_, row in data.iterrows():

# Get or create MainCategory

main\_category, \_ = MainCategory.objects.get\_or\_create(name=row['main\_category'])

# Get or create SubCategory

sub\_category, \_ = SubCategory.objects.get\_or\_create(name=row['sub\_category'], main\_category=main\_category)

# Create Product

Product.objects.create(

name=row['name'],

main\_category=main\_category,

sub\_category=sub\_category,

image\_url=row['image\_url'],

site\_link=row['site\_link'],

ratings=row['ratings'] if not pd.isna(row['ratings']) else None,

no\_of\_ratings=row['no\_of\_ratings'] if not pd.isna(row['no\_of\_ratings']) else None,

discount\_price=clean\_price(row['discount\_price']),

actual\_price=clean\_price(row['actual\_price']),

)

# Call the import function

if \_\_name\_\_ == "\_\_main\_\_":

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'urban\_cart.settings')

import django

django.setup()

import\_data()

print("Data imported successfully!")

**How to Use the Datasets**

1. Place all your .csv files in the datasets/ folder.
2. Run the import\_data.py script to populate the database:

bash

CopyEdit

python import\_data.py

1. The data from the CSV files will now be stored in your SQLite database.