In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_excel("OnlineRetail (1).xlsx")
df.head()
```

Out[2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cou
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Un Kingd
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Un Kingd
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Un Kingd
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	Un Kingd
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	Un Kingd

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   InvoiceNo    541909 non-null  object        
 1   StockCode    541909 non-null  object        
 2   Description  540455 non-null  object        
 3   Quantity     541909 non-null  int64         
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64       
 6   CustomerID   406829 non-null  float64       
 7   Country      541909 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

In [4]:

```
df.describe()
```

Out[4]:

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	541909.000000	541909	541909.000000	406829.000000
mean	9.552250	2011-07-04 13:34:57.156386048	4.611114	15287.690570
min	-80995.000000	2010-12-01 08:26:00	-11062.060000	12346.000000
25%	1.000000	2011-03-28 11:34:00	1.250000	13953.000000
50%	3.000000	2011-07-19 17:17:00	2.080000	15152.000000
75%	10.000000	2011-10-19 11:27:00	4.130000	16791.000000
max	80995.000000	2011-12-09 12:50:00	38970.000000	18287.000000
std	218.081158	NaN	96.759853	1713.600303

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

In [6]:

```
df = df.dropna(subset=['Description'])
```

In [7]:

```
df = df[(df['Quantity'] > 0) & (df['UnitPrice'] > 0)]
```

In [8]:

```
df = df[~df['InvoiceNo'].astype(str).str.startswith('C')]
```

In [9]:

```
df = df.dropna(subset=['CustomerID'])
```

In [10]:

```
df = df.reset_index(drop=True)
```

In [11]:

```
df.info()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397884 entries, 0 to 397883
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   InvoiceNo    397884 non-null  object        
 1   StockCode    397884 non-null  object        
 2   Description  397884 non-null  object        
 3   Quantity     397884 non-null  int64         
 4   InvoiceDate  397884 non-null  datetime64[ns]
 5   UnitPrice    397884 non-null  float64       
 6   CustomerID   397884 non-null  float64       
 7   Country      397884 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 24.3+ MB
```

Out[11]:

```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

In [12]:

```
top_products = df.groupby('Description')['Quantity'].sum().sort_values(ascending=Fa
print(top_products)
```

```
Description
PAPER CRAFT , LITTLE BIRDIE           80995
MEDIUM CERAMIC TOP STORAGE JAR        77916
WORLD WAR 2 GLIDERS ASSTD DESIGNS     54415
JUMBO BAG RED RETROSPOT               46181
WHITE HANGING HEART T-LIGHT HOLDER    36725
ASSORTED COLOUR BIRD ORNAMENT         35362
PACK OF 72 RETROSPOT CAKE CASES       33693
POPCORN HOLDER                        30931
RABBIT NIGHT LIGHT                    27202
MINI PAINT SET VINTAGE                26076
Name: Quantity, dtype: int64
```

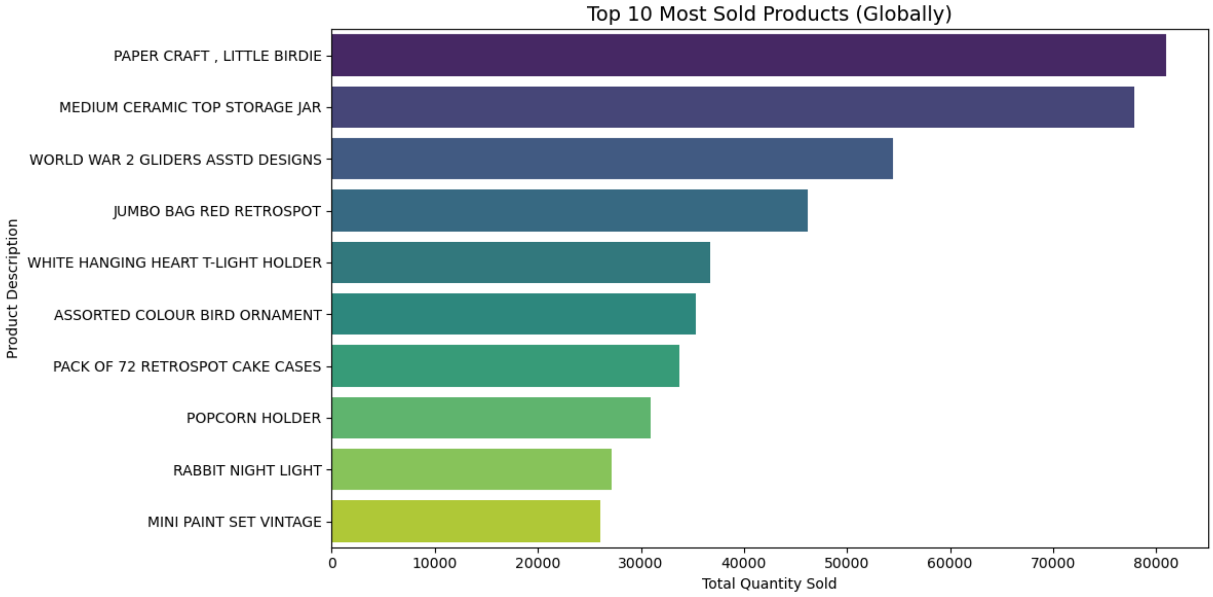
In [13]:

```
plt.figure(figsize=(12,6))
sns.barplot(x=top_products.values, y=top_products.index, palette='viridis')
plt.title('Top 10 Most Sold Products (Globally)', fontsize=14)
plt.xlabel('Total Quantity Sold')
plt.ylabel('Product Description')
plt.tight_layout()
plt.show()
```

```
C:\Users\akhil\AppData\Local\Temp\ipykernel_14508\2932963806.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_products.values, y=top_products.index, palette='viridis')
```



In [14]:
```python
country_top = df.groupby(['Country', 'Description'])['Quantity'].sum().reset_index(
country_top = country_top.sort_values(['Country', 'Quantity'], ascending=[True, Fal

# Example: Show top 5 products in France
france_top5 = country_top[country_top['Country'] == 'France'].head(5)
print(france_top5)
```

```
     Country                     Description  Quantity
6574  France               RABBIT NIGHT LIGHT      4000
6351  France           MINI PAINT SET VINTAGE      2196
6648  France   RED TOADSTOOL LED NIGHT LIGHT       1291
6857  France     SET/6 RED SPOTTY PAPER CUPS       1272
5684  France   ASSORTED COLOUR BIRD ORNAMENT       1204
```

In [15]:
```python
# Month column
df['Month'] = df['InvoiceDate'].dt.month

# Group by Month and Description
month_top = df.groupby(['Month', 'Description'])['Quantity'].sum().reset_index()
month_top = month_top.sort_values(['Month', 'Quantity'], ascending=[True, False])

# Top 5 products in December (month = 12)
december_top5 = month_top[month_top['Month'] == 12].head(5)
print(december_top5)
```

```
       Month                         Description  Quantity
27872     12        PAPER CRAFT , LITTLE BIRDIE     80995
29071     12   WORLD WAR 2 GLIDERS ASSTD DESIGNS      6502
28082     12                      POPCORN HOLDER      5362
29002     12  WHITE HANGING HEART T-LIGHT HOLDER      4356
27827     12     PACK OF 72 RETROSPOT CAKE CASES      3856
```

In [16]:

```
pivot_table = df.pivot_table(
    index='Description',
    columns='CustomerID',
    values='Quantity',
    aggfunc='sum',
    fill_value=0
)
```

In [17]:

```
pivot_table.head()
```

Out[17]:

CustomerID	12346.0	12347.0	12348.0	12349.0	12350.0	12352.0	12353.0	12354.0	12
Description									
10 COLOUR SPACEBOY PEN	0	0	0	0	0	0	0	0	
12 COLOURED PARTY BALLOONS	0	0	0	0	0	0	0	0	
12 DAISY PEGS IN WOOD BOX	0	0	0	0	0	0	0	0	
12 EGG HOUSE PAINTED WOOD	0	0	0	0	0	0	0	0	
12 HANGING EGGS HAND PAINTED	0	0	0	0	0	0	0	0	

5 rows × 4338 columns

In [18]:

```
# Choose a product
product_name = "WHITE HANGING HEART T-LIGHT HOLDER"

pivot_transposed = pivot_table.T

# Compute correlation with all other products
similar_items = pivot_transposed.corrwith(pivot_transposed[product_name])

# Drop missing values and sort by similarity
similar_items = similar_items.dropna().drop(product_name).sort_values(ascending=Fal
```

In [19]:

```
print(similar_items.head(5))
```

```
Description
GIN + TONIC DIET METAL SIGN         0.750247
RED HANGING HEART T-LIGHT HOLDER    0.657817
WASHROOM METAL SIGN                 0.642895
LAUNDRY 15C METAL SIGN              0.641969
GREEN VINTAGE SPOT BEAKER           0.631999
dtype: float64
```

In [20]:

```
def recommend_products(product_name, pivot_table, top_n=5):
    # Transpose the pivot table
    pivot_transposed = pivot_table.T

    # Check if the product exists
    if product_name not in pivot_transposed.columns:
        return f"'{product_name}' not found in product list."

    # Compute correlations
    similarity = pivot_transposed.corrwith(pivot_transposed[product_name])
    similarity = similarity.dropna().drop(product_name).sort_values(ascending=False

    return similarity.head(top_n)
```

In [21]:

```
recommendations = recommend_products("WHITE HANGING HEART T-LIGHT HOLDER", pivot_ta
print(recommendations)
```

```
Description
GIN + TONIC DIET METAL SIGN         0.750247
RED HANGING HEART T-LIGHT HOLDER    0.657817
WASHROOM METAL SIGN                 0.642895
LAUNDRY 15C METAL SIGN              0.641969
GREEN VINTAGE SPOT BEAKER           0.631999
dtype: float64
```

In [22]:

```
product_list = pivot_table.index.tolist()
for p in product_list[:20]:  # See first 20 products
    print(p)
```

```
10 COLOUR SPACEBOY PEN
12 COLOURED PARTY BALLOONS
12 DAISY PEGS IN WOOD BOX
12 EGG HOUSE PAINTED WOOD
12 HANGING EGGS HAND PAINTED
12 IVORY ROSE PEG PLACE SETTINGS
12 MESSAGE CARDS WITH ENVELOPES
12 PENCIL SMALL TUBE WOODLAND
12 PENCILS SMALL TUBE RED RETROSPOT
12 PENCILS SMALL TUBE SKULL
12 PENCILS TALL TUBE POSY
12 PENCILS TALL TUBE RED RETROSPOT
12 PENCILS TALL TUBE SKULLS
12 PENCILS TALL TUBE WOODLAND
12 PINK HEN+CHICKS IN BASKET
12 PINK ROSE PEG PLACE SETTINGS
12 RED ROSE PEG PLACE SETTINGS
15 PINK FLUFFY CHICKS IN BOX
15CM CHRISTMAS GLASS BALL 20 LIGHTS
16 PC CUTLERY SET PANTRY DESIGN
```

In [23]:

```
recommendations = recommend_products("12 MESSAGE CARDS WITH ENVELOPES", pivot_table
print(recommendations)
```

```
Description
6PC WOOD PLATE SET DISPOSABLE          0.798365
UNION FLAG WINDSOCK                    0.795746
SET/12 FUNKY FELT FLOWER PEG IN BAG    0.783474
IVORY PILLAR CANDLE SILVER FLOCK       0.775551
3 RAFFIA RIBBONS VINTAGE CHRISTMAS     0.756977
dtype: float64
```

In [24]:

```
recommendations = recommend_products("16 PC CUTLERY SET PANTRY DESIGN", pivot_table
print(recommendations)
```

```
Description
CLASSIC GLASS SWEET JAR         0.458971
REGENCY MIRROR WITH SHUTTERS    0.249632
METAL HERB GERDEN CONTAINER     0.223303
PICTURE DOMINOES                0.158141
UNION STRIPE CUSHION COVER      0.150927
dtype: float64
```

In [ ]: