MILESTONE#4

I have added the code for MySQL connection with Python and some analysis of existing data.

```
import mysql.connector
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [56]:

```
# Establish connection
conn = mysql.connector.connect(
  host="localhost",      # Host name
  user="root",   # Mysql user name
  password="*", # SQL password
  database="Explore_Secure_TravelInsurance" # Database name
)
```

In [58]:

```
# Create cursor object
cursor = conn.cursor()

# Execute a query
cursor.execute("SHOW TABLES;")


# Fetch and display results
tables = cursor.fetchall()
print("Tables in the database:", tables)
```

```
Tables in the database: [('Agent',), ('AgentSales',), ('Claim',), ('ClaimProcessing',), ('CountryOffers',), ('Custome
r',), ('CustomerPolicy',), ('HighRiskCoverage',), ('InsurancePolicy',), ('Partnership',), ('Payment',), ('Personaliz
edQuote',), ('PolicyType',), ('TravelPlatform',)]
```

In [60]:

```
cursor.execute("SELECT * FROM `Customer`;")
print(cursor.fetchall())
```

```
[(1, 'Alice Johnson', 'Female', datetime.date(1985, 6, 15), 'Engineer', 'Business', 'Medium'), (2, 'Bob Smith', 'Ma
le', datetime.date(1990, 9, 23), 'Doctor', 'Medical', 'High'), (3, 'Charlie Brown', 'Male', datetime.date(1987, 3, 1
0), 'Professor', 'Education', 'Low'), (4, 'Diana Prince', 'Female', datetime.date(1995, 12, 5), 'Journalist', 'Busine
ss', 'Medium'), (5, 'Evan Rogers', 'Male', datetime.date(1980, 7, 18), 'Athlete', 'Leisure', 'High'), (6, 'Fiona Davis
', 'Female', datetime.date(1992, 11, 30), 'Blogger', 'Leisure', 'Low'), (7, 'George Miller', 'Male', datetime.date(19
83, 4, 25), 'Lawyer', 'Business', 'Medium'), (8, 'Helen Carter', 'Female', datetime.date(1975, 5, 22), 'Retired', 'L
eisure', 'Low'), (9, 'Ian Thompson', 'Male', datetime.date(1999, 8, 14), 'Student', 'Education', 'Medium'), (10, 'J
essica Lee', 'Female', datetime.date(1988, 2, 17), 'Entrepreneur', 'Business', 'High'), (11, 'Kevin Malone', 'Male'
, datetime.date(1982, 11, 2), 'Accountant', 'Business', 'Low'), (12, 'Angela Martin', 'Female', datetime.date(198
0, 6, 25), 'Financial Analyst', 'Business', 'Medium'), (13, 'Ryan Howard', 'Male', datetime.date(1985, 5, 10), 'Ma
rketing Executive', 'Business', 'High'), (14, 'Kelly Kapoor', 'Female', datetime.date(1987, 2, 13), 'Social Media
Manager', 'Leisure', 'Medium'), (15, 'Toby Flenderson', 'Male', datetime.date(1975, 8, 11), 'HR Manager', 'Leisu
re', 'Low'), (16, 'Creed Bratton', 'Male', datetime.date(1960, 10, 14), 'Retired', 'Leisure', 'Low'), (17, 'Meredith
Palmer', 'Female', datetime.date(1978, 4, 22), 'Sales Representative', 'Business', 'Medium'), (18, 'Oscar Martin
```

ez', 'Male', datetime.date(1981, 9, 7), 'Tax Consultant', 'Business', 'Low'), (19, 'Jan Levinson', 'Female', datetim
e.date(1973, 12, 30), 'Senior Manager', 'Business', 'High'), (20, 'David Wallace', 'Male', datetime.date(1968, 5,
17), 'Executive', 'Business', 'High')]

In [62]:

```
cursor = conn.cursor()

# Fetch all table names
cursor.execute("SHOW TABLES;")
tables = [table[0] for table in cursor.fetchall()]

# Iterate over each table and fetch data
for table in tables:
    print(f"\nFetching data from table: {table}")

    try:
        query = f"SELECT * FROM `{table}`;"  # Use backticks for safety
        cursor.execute(query)

        # Get column names
        columns = [desc[0] for desc in cursor.description]

        # Fetch data
        rows = cursor.fetchall()

        # Convert to Pandas DataFrame
        df = pd.DataFrame(rows, columns=columns)

        # Display first 5 rows
        print(df.head())
    except Exception as e:
        print(f"Error fetching data from {table}: {e}")
```

```
Fetching data from table: Agent
   agent_id          name          contact_info
0         1  Michael Scott  michael@insurance.com
1         2     Pam Beesly     pam@insurance.com
2         3    Jim Halpert     jim@insurance.com
3         4  Dwight Schrute   dwight@insurance.com
4         5  Stanley Hudson  stanley@insurance.com

Fetching data from table: AgentSales
   agent_id  policy_id   sale_date commission
0         1        101  2025-01-05     50.00
1         2        102  2025-01-10     75.00
2         3        103  2025-02-01     30.00
3         4        104  2025-02-15     20.00
4         5        105  2025-03-01     90.00

Fetching data from table: Claim
   claim_id  customer_id  policy_id claim_status claim_amount submission_date
0         1            1        101      Pending     5000.00      2025-02-20
1         2            2        103     Approved     1500.00      2025-02-25
2         3            3        104     Rejected      800.00      2025-03-01
3         4            4        102   Processing    12000.00      2025-03-10
4         5            5        105      Pending     4500.00      2025-03-15
```

Fetching data from table: ClaimProcessing

	processing_id	claim_id	assigned_agent	processing_status
0	1	1	2	Under Review
1	2	2	3	Completed
2	3	3	1	Rejected
3	4	4	4	In Progress
4	5	5	5	Pending

Fetching data from table: CountryOffers

	offer_id	country_name	discount	\
0	1	USA	10.00	
1	2	Canada	5.00	
2	3	Germany	7.50	
3	4	Australia	12.00	
4	5	Japan	6.00	

	special_terms
0	10% discount on all travel policies
1	Applicable only for first-time travelers
2	Limited to business travel policies
3	Special discount for students and seniors
4	Coverage includes natural disaster protection

Fetching data from table: Customer

	customer_id	name	gender	date_of_birth	occupation	\
0	1	Alice Johnson	Female	1985-06-15	Engineer	
1	2	Bob Smith	Male	1990-09-23	Doctor	
2	3	Charlie Brown	Male	1987-03-10	Professor	
3	4	Diana Prince	Female	1995-12-05	Journalist	
4	5	Evan Rogers	Male	1980-07-18	Athlete	

	travel_purpose	risk_level
0	Business	Medium
1	Medical	High
2	Education	Low
3	Business	Medium
4	Leisure	High

Fetching data from table: CustomerPolicy

	customer_id	policy_id	purchase_date	expiry_date
0	1	101	2024-01-01	2025-01-01
1	2	102	2024-02-10	2025-02-10
2	2	103	2024-02-15	2025-02-15
3	3	103	2023-12-05	2024-12-05
4	3	104	2024-03-10	2025-03-10

Fetching data from table: HighRiskCoverage

	coverage_id	policy_id	risk_type	coverage_amount
0	1	101	Extreme Sports Injury	50000.00
1	2	102	War Zone Coverage	75000.00
2	3	103	Pandemic Insurance	60000.00
3	4	104	Terrorism Coverage	80000.00
4	5	105	Political Unrest	70000.00

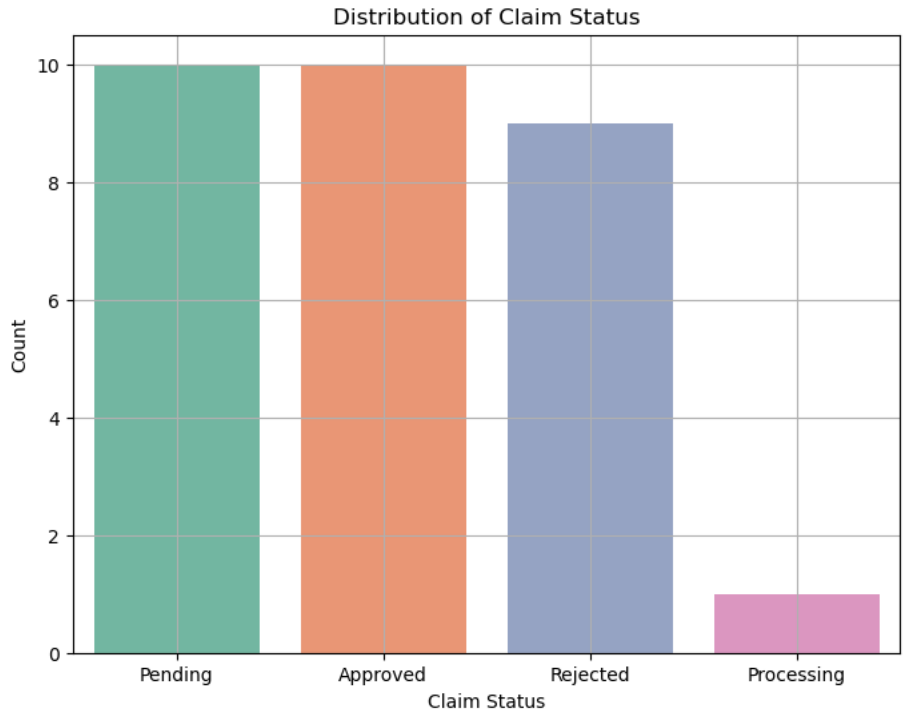
Fetching data from table: InsurancePolicy

```
   policy_id                 policy_name  policy_type_id  \
0        101           Basic Health Plan               1
1        102     Comprehensive Travel Plan             1
2        103  Flight Cancellation Protection             2
3        104       Baggage Protection Plan             3
4        105      Extreme Sports Insurance             4

                        coverage_details premium
0    Covers hospitalization and medical expenses  200.00
1  Includes health, baggage, and flight coverage  500.00
2      Covers full refund on flight cancellation  150.00
3              Covers baggage loss and theft  100.00
4       Covers injuries from sports activities  350.00

Fetching data from table: Partnership
   platform_id  policy_id commission partnership_date
0            1        101     20.00       2025-01-01
1            2        102     25.00       2025-01-05
2            3        103     15.00       2025-02-01
3            4        104     10.00       2025-02-15
4            5        105     30.00       2025-03-01

Fetching data from table: Payment
   payment_id  customer_id  policy_id  amount payment_date
0           1            1        101  200.00   2025-02-20
1           2            2        103  150.00   2025-02-25
2           3            3        104  100.00   2025-03-01
3           4            4        102  500.00   2025-03-10
4           5            5        105  350.00   2025-03-15

Fetching data from table: PersonalizedQuote
   quote_id  customer_id  policy_id custom_premium  \
0         1            1        101         180.00
1         2            2        103         140.00
2         3            3        104          90.00
3         4            4        102         480.00
4         5            5        105         330.00

             coverage_adjustments  quote_date
0  Extended hospitalization coverage  2025-03-01
1         Added trip delay coverage  2025-03-02
2                              None  2025-03-03
3    Added higher baggage protection  2025-03-04
4   Extreme skiing coverage included  2025-03-05

Fetching data from table: PolicyType
   policy_type_id              type_name  \
0               1         Health Insurance
1               2      Flight Cancellation
2               3            Baggage Loss
3               4   Extreme Sports Coverage
4               5   Senior Citizen Coverage

                              description
0            Covers medical expenses during travel
```

```
1  Covers cost of canceled flights due to emergen...
2                Covers lost or stolen baggage
3          Covers injuries from adventure sports
4          Special policies for elderly travelers

Fetching data from table: TravelPlatform
   platform_id       name
0            1     Expedia
1            2  Booking.com
2            3      Airbnb
3            4   Skyscanner
4            5       Kayak
```

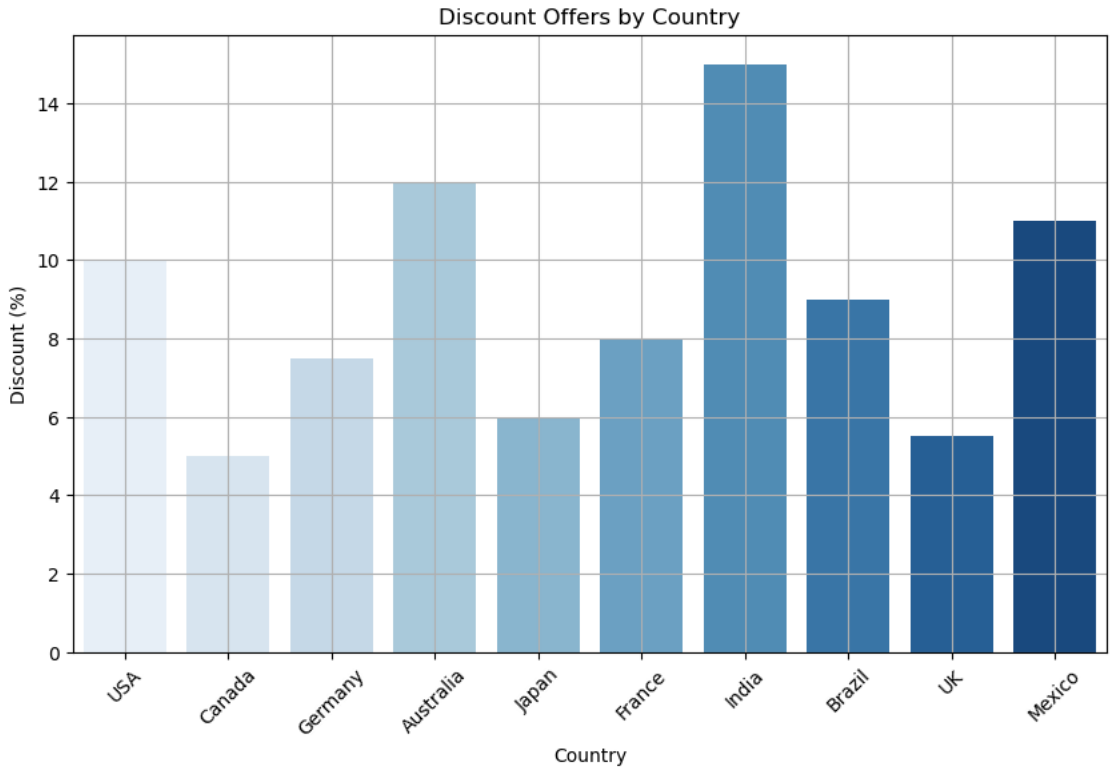
In [64]:

```
# Fetching the data from claim table create a plot to see the claim status of each one from the data.
# Fetch data from the Claim table
query = "SELECT * FROM Claim;"
cursor.execute(query)

# Get column names
columns = [desc[0] for desc in cursor.description]

# Fetch data
rows = cursor.fetchall()

# Convert to Pandas DataFrame
df_claim = pd.DataFrame(rows, columns=columns)
# Plotting the claim status distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df_claim, x='claim_status', palette='Set2')
plt.title('Distribution of Claim Status')
plt.xlabel('Claim Status')
plt.ylabel('Count')
plt.grid(True)
plt.show()
/var/folders/34/jqpjg_x508j0x978h6stq2j80000gn/T/ipykernel_5177/154940699.py:15: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df_claim, x='claim_status', palette='Set2')
```



In [70]:

```
#Fetching from country tables data to see countires which offers discounts.
# Fetch data from the CountryOffers table
query = "SELECT * FROM CountryOffers;"
cursor.execute(query)

# Get column names
columns = [desc[0] for desc in cursor.description]

# Fetch data
rows = cursor.fetchall()

# Convert to Pandas DataFrame
df_country_offers = pd.DataFrame(rows, columns=columns)

# Plot the distribution of discounts by country
plt.figure(figsize=(10, 6))
sns.barplot(data=df_country_offers, x='country_name', y='discount', palette='Blues')
plt.title('Discount Offers by Country')
plt.xlabel('Country')
plt.ylabel('Discount (%)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
/var/folders/34/jqpjg_x508j0x978h6stq2j80000gn/T/ipykernel_5177/2443226331.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=df_country_offers, x='country_name', y='discount', palette='Blues')
```



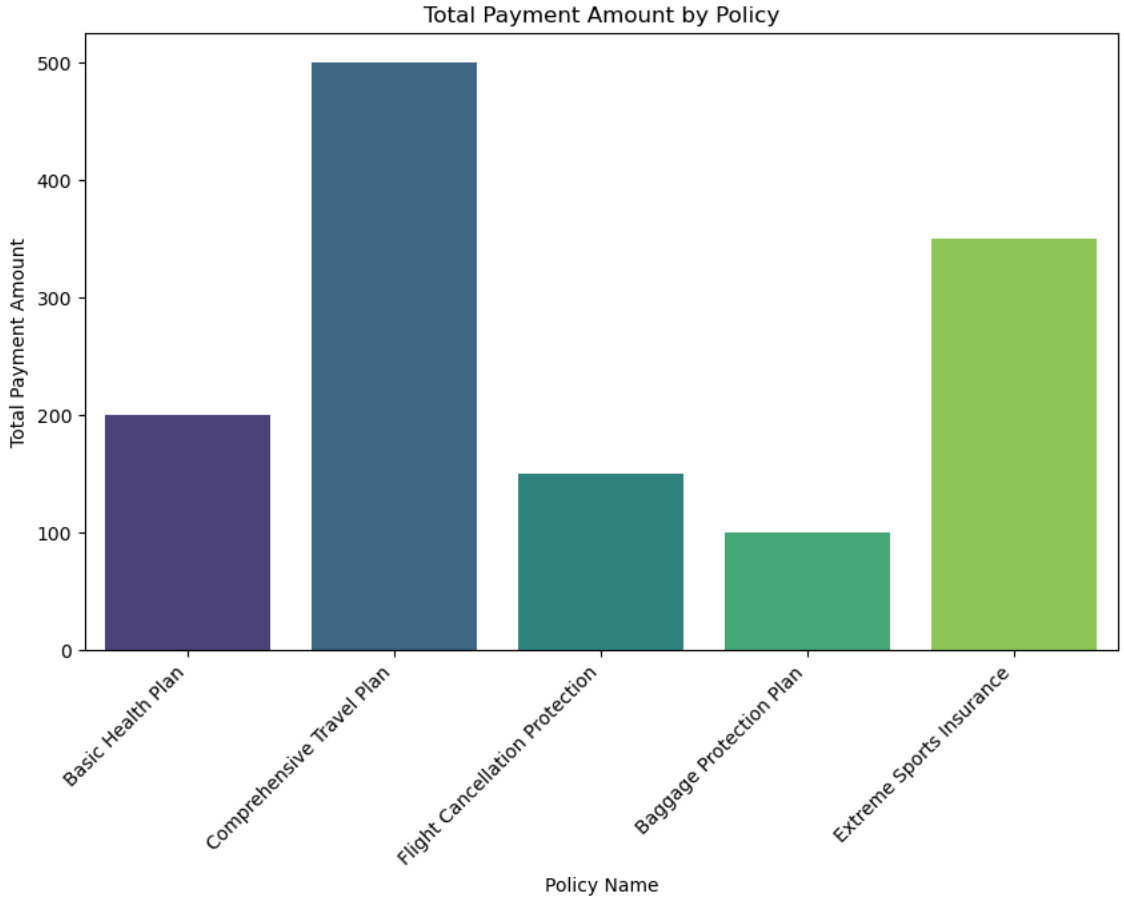
In [76]:

```
#Sum of amount customers paid for each policy .
cursor = conn.cursor()

# Fetch data from the Payment table
cursor.execute("SELECT * FROM Payment;")
columns = [desc[0] for desc in cursor.description]
rows = cursor.fetchall()
df_payment = pd.DataFrame(rows, columns=columns)

# Fetch data from the InsurancePolicy table
cursor.execute("SELECT * FROM InsurancePolicy;")
columns = [desc[0] for desc in cursor.description]
rows = cursor.fetchall()
df_insurance_policy = pd.DataFrame(rows, columns=columns)

# Group payments by policy_id and sum the amounts
df_payment_policy = df_payment.groupby('policy_id')['amount'].sum().reset_index()

# Merge payment data with insurance policy data to get policy names
df_payment_policy = pd.merge(df_payment_policy, df_insurance_policy, on='policy_id', how='inner')

# Plot total payment amount by policy
plt.figure(figsize=(10, 6))
sns.barplot(data=df_payment_policy, x='policy_name', y='amount', palette='viridis')
plt.title('Total Payment Amount by Policy')
plt.xlabel('Policy Name')
plt.ylabel('Total Payment Amount')
plt.xticks(rotation=45, ha='right')
plt.show()
```

/var/folders/34/jqpjg_x508j0x978h6stq2j80000gn/T/ipykernel_5177/779888698.py:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(data=df_payment_policy, x='policy_name', y='amount', palette='viridis')
```



In [ ]: