Задание

Провести разведочный анализ данных, придумать продуктовые и технические гипотезы — какую ценность м
извлечь из данных для организации, которая предоставила данные.

Импорты и загрузка данных

In [4]:
```
# Импорты библиотек
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings('ignore')

# Настройки для отображения
plt.style.use('seaborn-v0_8')
sns.set_palette("husl")
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', None)

print("Библиотеки импортированы успешно!")

# Загрузка данных
print("Загружаем данные...")

# Основной датасет с транзакциями
df_transactions = pd.read_parquet('./data/transaction_fraud_data.parquet'

# Данные обменных курсов
df_exchange = pd.read_parquet('./data/historical_currency_exchange.parque

print(f"Транзакции загружены: {df_transactions.shape}")
print(f"Курсы валют загружены: {df_exchange.shape}")
```

```
Библиотеки импортированы успешно!
Загружаем данные...
Транзакции загружены: (7483766, 23)
Курсы валют загружены: (31, 12)
```

Основная информация о данных

In [5]:
```
print("=== ОСНОВНАЯ ИНФОРМАЦИЯ О ДАННЫХ ===")
print("\n1. Информация о транзакциях:")
print(df_transactions.info())

print("\n2. Первые 5 строк транзакций:")
print(df_transactions.head())

print("\n3. Статистика числовых признаков:")
print(df_transactions.describe())

print("\n4. Информация о курсах валют:")
print(df_exchange.info())
print(df_exchange.head())
```

```
=== ОСНОВНАЯ ИНФОРМАЦИЯ О ДАННЫХ ===

1. Информация о транзакциях:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7483766 entries, 0 to 7483765
Data columns (total 23 columns):
 #   Column                   Dtype
---  ------                   -----
 0   transaction_id           object
 1   customer_id              object
 2   card_number              int64
 3   timestamp                datetime64[us]
 4   vendor_category          object
 5   vendor_type              object
 6   vendor                   object
 7   amount                   float64
 8   currency                 object
 9   country                  object
 10  city                     object
 11  city_size                object
 12  card_type                object
 13  is_card_present          bool
 14  device                   object
 15  channel                  object
 16  device_fingerprint       object
 17  ip_address               object
 18  is_outside_home_country  bool
 19  is_high_risk_vendor      bool
 20  is_weekend               bool
 21  last_hour_activity       object
 22  is_fraud                 bool
dtypes: bool(5), datetime64[us](1), float64(1), int64(1), object(15)
memory usage: 1.0+ GB
None

2. Первые 5 строк транзакций:
  transaction_id customer_id       card_number                  timestamp  \
0    TX_a0ad2a2a  CUST_72886  6646734767813109 2024-09-30 00:00:01.034820
1    TX_3599c101  CUST_70474   376800864692727 2024-09-30 00:00:01.764464
2    TX_a9461c6d  CUST_10715  5251909460951913 2024-09-30 00:00:02.273762
3    TX_7be21fc4  CUST_16193   376079286931183 2024-09-30 00:00:02.297466
4    TX_150f490b  CUST_87572  6172948052178810 2024-09-30 00:00:02.544063

  vendor_category vendor_type          vendor     amount currency    country
0      Restaurant   fast_food       Taco Bell     294.87      GBP         UK
1   Entertainment      gaming           Steam    3368.97      BRL     Brazil
2         Grocery    physical     Whole Foods  102582.38      JPY      Japan
3             Gas       major           Exxon     630.60      AUD  Australia
4      Healthcare     medical  Medical Center  724949.27      NGN    Nigeria

           city city_size        card_type  is_card_present   device channel
0  Unknown City    medium  Platinum Credit            False  iOS App  mobile
1  Unknown City    medium  Platinum Credit            False     Edge     web
2  Unknown City    medium  Platinum Credit            False  Firefox     web
3  Unknown City    medium    Premium Debit            False  iOS App  mobile
4  Unknown City    medium      Basic Debit            False   Chrome     web

                 device_fingerprint       ip_address  is_outside_home_country
0  e8e6160445c935fd0001501e4cbac8bc   197.153.60.199                    False
1  a73043a57091e775af37f252b3a32af9  208.123.221.203                     True
```

```
2  218864e94ceaa41577d216b149722261   10.194.159.204                    False
3  70423fa3a1e74d01203cf93b51b9631d   17.230.177.225                    False
4  9880776c7b6038f2af86bd4e18a1b1a4  136.241.219.151                     True

   is_high_risk_vendor  is_weekend  \
0                False       False
1                 True       False
2                False       False
3                False       False
4                False       False


                                                    last_hour_activity  \
0  {'num_transactions': 1197, 'total_amount': 33498556.080464985,
'unique_merchants': 105, 'unique_countries': 12, 'max_single_amount':
1925480.6324148502}
1    {'num_transactions': 509, 'total_amount': 20114759.055250417,
'unique_merchants': 100, 'unique_countries': 12, 'max_single_amount':
5149117.011434267}
2     {'num_transactions': 332, 'total_amount': 39163854.72992601,
'unique_merchants': 97, 'unique_countries': 12, 'max_single_amount':
1852242.1831665323}
3     {'num_transactions': 764, 'total_amount': 22012599.81898404,
'unique_merchants': 105, 'unique_countries': 12, 'max_single_amount':
2055798.460682913}
4      {'num_transactions': 218, 'total_amount': 4827636.199648165,
'unique_merchants': 88, 'unique_countries': 12, 'max_single_amount':
1157231.252130005}

   is_fraud
0     False
1      True
2     False
3     False
4      True

3. Статистика числовых признаков:
        card_number                      timestamp        amount
count  7.483766e+06                        7483766  7.483766e+06
mean   4.222100e+15  2024-10-15 12:36:38.052469  4.792468e+04
min    3.700086e+14  2024-09-30 00:00:01.034820  1.000000e-02
25%    4.004400e+15  2024-10-07 18:08:27.325326  3.635300e+02
50%    5.010745e+15  2024-10-15 12:46:31.739295  1.177450e+03
75%    5.999914e+15  2024-10-23 07:37:00.969509  2.242953e+04
max    6.999728e+15  2024-10-30 23:59:59.101885  6.253153e+06
std    2.341170e+15                          NaN  1.775562e+05

4. Информация о курсах валют:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    31 non-null     object
 1   AUD     31 non-null     float64
 2   BRL     31 non-null     float64
 3   CAD     31 non-null     float64
 4   EUR     31 non-null     float64
 5   GBP     31 non-null     float64
 6   JPY     31 non-null     float64
```

```
 7   MXN     31 non-null     float64
 8   NGN     31 non-null     float64
 9   RUB     31 non-null     float64
 10  SGD     31 non-null     float64
 11  USD     31 non-null     int64
dtypes: float64(10), int64(1), object(1)
memory usage: 3.0+ KB
None
```

	date	AUD	BRL	CAD	EUR	GBP	JPY
0	2024-09-30	1.443654	5.434649	1.351196	0.895591	0.747153	142.573268
1	2024-10-01	1.442917	5.444170	1.352168	0.897557	0.746956	143.831429
2	2024-10-02	1.449505	5.425444	1.348063	0.903056	0.752241	143.806861
3	2024-10-03	1.456279	5.442044	1.351451	0.906018	0.754584	146.916773
4	2024-10-04	1.460930	5.477788	1.355260	0.906452	0.761891	146.592323

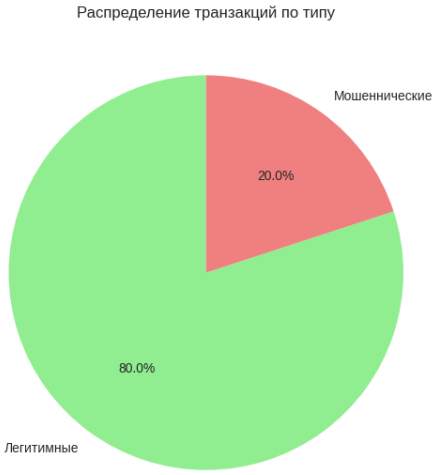
	MXN	NGN	RUB	SGD	USD
0	19.694724	1668.736400	94.133735	1.280156	1
1	19.667561	1670.694524	92.898519	1.284352	1
2	19.606748	1669.653006	94.583198	1.286983	1
3	19.457701	1670.097873	95.655442	1.294391	1
4	19.363467	1649.763738	94.755337	1.296800	1

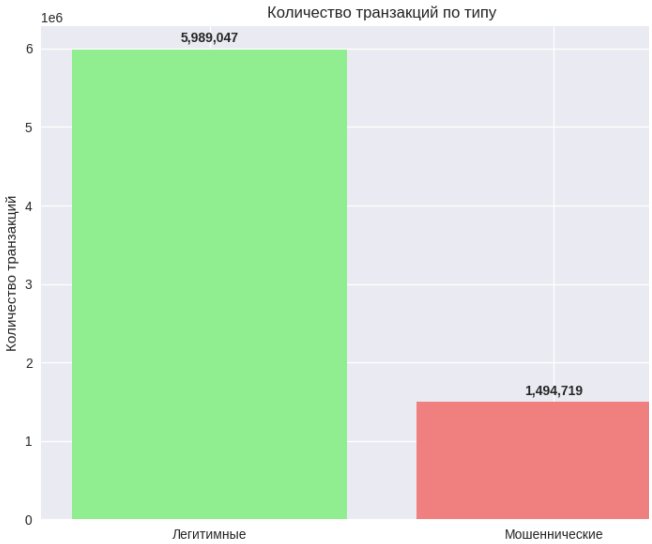
Анализ целевой переменной (is_fraud)

In [6]:

```python
fraud_stats = df_transactions['is_fraud'].value_counts()
fraud_percentage = (fraud_stats[True] / len(df_transactions)) * 100

print(f"Всего транзакций: {len(df_transactions):,}")
print(f"Мошеннических транзакций: {fraud_stats[True]:,}")
print(f"Легитимных транзакций: {fraud_stats[False]:,}")
print(f"Процент мошенничества: {fraud_percentage:.2f}%")

# Визуализация распределения
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Круговая диаграмма
ax1.pie(fraud_stats.values, labels=['Легитимные', 'Мошеннические'],
        autopct='%1.1f%%', startangle=90, colors=['lightgreen', 'lightcor
ax1.set_title('Распределение транзакций по типу')

# Столбчатая диаграмма
ax2.bar(['Легитимные', 'Мошеннические'], fraud_stats.values,
        color=['lightgreen', 'lightcoral'])
ax2.set_title('Количество транзакций по типу')
ax2.set_ylabel('Количество транзакций')

for i, v in enumerate(fraud_stats.values):
    ax2.text(i, v + max(fraud_stats.values) * 0.01, f'{v:,}',
             ha='center', va='bottom', fontweight='bold')

plt.tight_layout()
plt.show()
```

```
Всего транзакций: 7,483,766
Мошеннических транзакций: 1,494,719
Легитимных транзакций: 5,989,047
Процент мошенничества: 19.97%
```





Анализ временных паттернов

In [7]:

```
# Преобразование timestamp
df_transactions['timestamp'] = pd.to_datetime(df_transactions['timestamp'
df_transactions['date'] = df_transactions['timestamp'].dt.date
df_transactions['hour'] = df_transactions['timestamp'].dt.hour
df_transactions['day_of_week'] = df_transactions['timestamp'].dt.day_name
df_transactions['month'] = df_transactions['timestamp'].dt.month

# Анализ по часам
hourly_fraud = df_transactions.groupby(['hour', 'is_fraud']).size().unsta
hourly_fraud_rate = hourly_fraud[True] / (hourly_fraud[True] + hourly_fra

# Анализ по дням недели
daily_fraud = df_transactions.groupby(['day_of_week', 'is_fraud']).size()
daily_fraud_rate = daily_fraud[True] / (daily_fraud[True] + daily_fraud[F

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# По часам
ax1.plot(hourly_fraud_rate.index, hourly_fraud_rate.values, marker='o', l
ax1.set_title('Процент мошеннических транзакций по часам')
ax1.set_xlabel('Час')
ax1.set_ylabel('Процент мошенничества')
ax1.grid(True, alpha=0.3)

# По дням недели
days_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Sa
daily_fraud_rate_ordered = daily_fraud_rate.reindex(days_order)
ax2.bar(daily_fraud_rate_ordered.index, daily_fraud_rate_ordered.values,
        color='lightcoral', alpha=0.7)
ax2.set_title('Процент мошеннических транзакций по дням недели')
ax2.set_xlabel('День недели')
ax2.set_ylabel('Процент мошенничества')
ax2.tick_params(axis='x', rotation=45)

# Общее количество транзакций по часам
ax3.plot(hourly_fraud.index, hourly_fraud[False], label='Легитимные', mar
ax3.plot(hourly_fraud.index, hourly_fraud[True], label='Мошеннические', m
ax3.set_title('Количество транзакций по часам')
```

```
ax3.set_xlabel('Час')
ax3.set_ylabel('Количество транзакций')
ax3.legend()
ax3.grid(True, alpha=0.3)

# Общее количество транзакций по дням недели
daily_total_ordered = daily_fraud.reindex(days_order)
ax4.bar(daily_total_ordered.index, daily_total_ordered[False], label='Лег
ax4.bar(daily_total_ordered.index, daily_total_ordered[True], bottom=dail
        label='Мошеннические', alpha=0.7)
ax4.set_title('Количество транзакций по дням недели')
ax4.set_xlabel('День недели')
ax4.set_ylabel('Количество транзакций')
ax4.legend()
ax4.tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()

print(f"Период данных: с {df_transactions['date'].min()} по {df_transacti
print(f"Всего дней: {(df_transactions['date'].max() - df_transactions['da
```


```
Период данных: с 2024-09-30 по 2024-10-30
Всего дней: 31
```

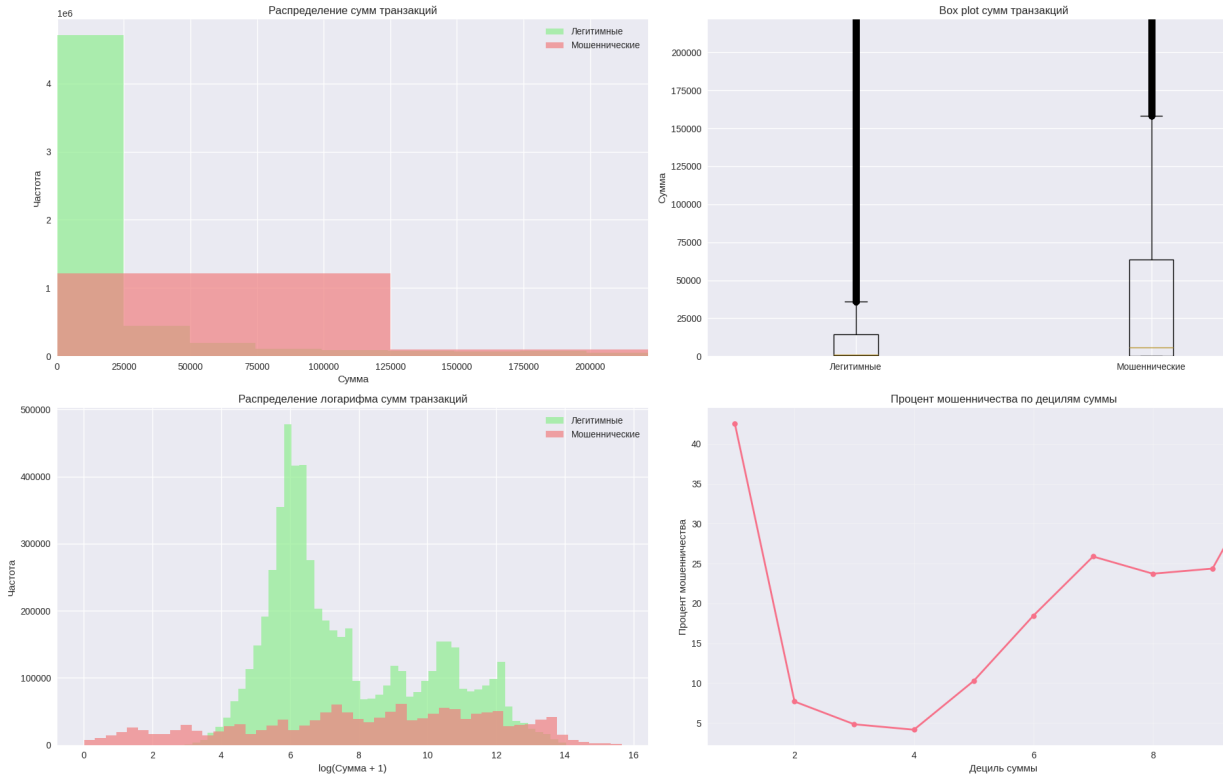
Анализ сумм транзакций

In [8]:

```
# Статистика по суммам
amount_stats = df_transactions.groupby('is_fraud')['amount'].describe()
print("Статистика сумм транзакций:")
print(amount_stats)

# Визуализация распределения сумм
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Гистограммы сумм
```

```
ax1.hist(df_transactions[df_transactions['is_fraud'] == False]['amount'],
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax1.hist(df_transactions[df_transactions['is_fraud'] == True]['amount'],
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax1.set_title('Распределение сумм транзакций')
ax1.set_xlabel('Сумма')
ax1.set_ylabel('Частота')
ax1.legend()
ax1.set_xlim(0, df_transactions['amount'].quantile(0.95))

# Box plot
ax2.boxplot([df_transactions[df_transactions['is_fraud'] == False]['amour
             df_transactions[df_transactions['is_fraud'] == True]['amount
            labels=['Легитимные', 'Мошеннические'])
ax2.set_title('Box plot сумм транзакций')
ax2.set_ylabel('Сумма')
ax2.set_ylim(0, df_transactions['amount'].quantile(0.95))

# Логарифмированное распределение
ax3.hist(np.log1p(df_transactions[df_transactions['is_fraud'] == False]['
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax3.hist(np.log1p(df_transactions[df_transactions['is_fraud'] == True]['a
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax3.set_title('Распределение логарифма сумм транзакций')
ax3.set_xlabel('log(Сумма + 1)')
ax3.set_ylabel('Частота')
ax3.legend()

# Процент мошенничества по квантилям суммы
amount_quantiles = pd.qcut(df_transactions['amount'], q=10, labels=False)
fraud_by_amount = df_transactions.groupby(amount_quantiles)['is_fraud'].m
ax4.plot(range(1, 11), fraud_by_amount.values, marker='o', linewidth=2, m
ax4.set_title('Процент мошенничества по децилям суммы')
ax4.set_xlabel('Дециль суммы')
ax4.set_ylabel('Процент мошенничества')
ax4.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Анализ экстремальных сумм
print(f"\nТоп-10 самых крупных транзакций:")
print(df_transactions.nlargest(10, 'amount')[['amount', 'currency', 'venc
```

```
Статистика сумм транзакций:
              count           mean            std    min      25%      50%  \
is_fraud
False     5989047.0   30242.538284   87656.818774  17.95  368.220   903.51
True      1494719.0  118773.589871  347542.933086   0.01  295.585  5626.06

               75%         max
is_fraud
False     14535.93  1240629.47
True      63556.20  6253152.62
```



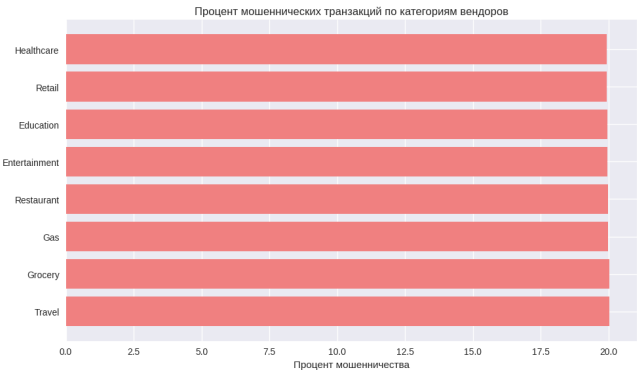

```
Топ-10 самых крупных транзакций:
             amount currency vendor_category  is_fraud
3170744  6253152.62      NGN          Travel      True
510694   6243513.37      NGN          Travel      True
3295531  6243212.03      NGN          Travel      True
1973827  6227400.98      NGN          Travel      True
2949588  6222909.52      NGN          Travel      True
5283061  6222174.26      NGN          Travel      True
5071905  6207032.30      NGN          Travel      True
6902797  6202390.22      NGN          Travel      True
4624170  6198732.75      NGN          Travel      True
744379   6195898.12      NGN          Travel      True
```

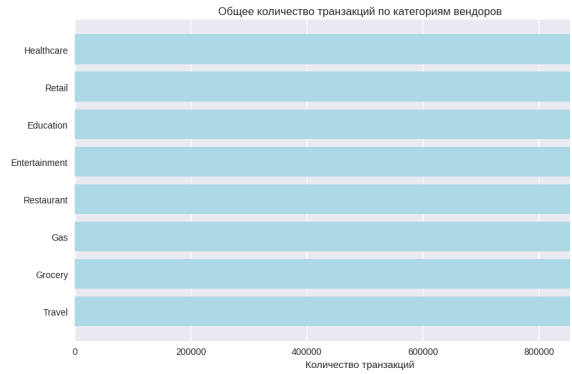
Анализ категорий вендоров

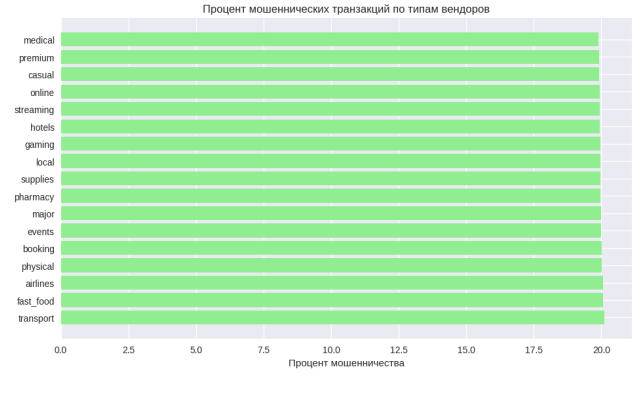
In [9]:
```
# Статистика по категориям
vendor_fraud = df_transactions.groupby('vendor_category')['is_fraud'].agg
vendor_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мош
vendor_fraud['Процент_мошенничества'] = vendor_fraud['Процент_мошенничест
vendor_fraud = vendor_fraud.sort_values('Процент_мошенничества', ascendir

print("Статистика мошенничества по категориям вендоров:")
print(vendor_fraud)

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Процент мошенничества по категориям
ax1.barh(vendor_fraud.index, vendor_fraud['Процент_мошенничества'], color
ax1.set_title('Процент мошеннических транзакций по категориям вендоров')
ax1.set_xlabel('Процент мошенничества')

# Количество транзакций по категориям
ax2.barh(vendor_fraud.index, vendor_fraud['Всего_транзакций'], color='lig
ax2.set_title('Общее количество транзакций по категориям вендоров')
ax2.set_xlabel('Количество транзакций')
```

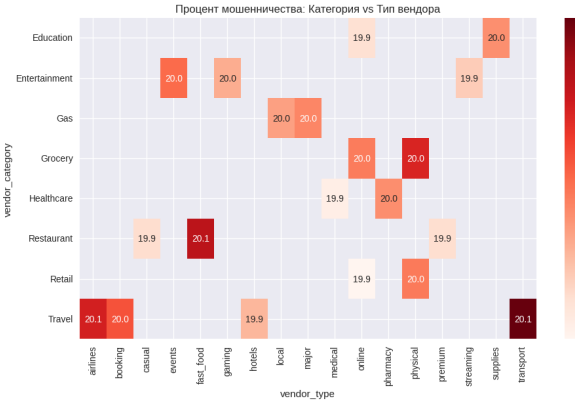
```
# Анализ типов вендоров
vendor_type_fraud = df_transactions.groupby('vendor_type')['is_fraud'].ag
vendor_type_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процен
vendor_type_fraud['Процент_мошенничества'] = vendor_type_fraud['Процент_м
vendor_type_fraud = vendor_type_fraud.sort_values('Процент_мошенничества'

ax3.barh(vendor_type_fraud.index, vendor_type_fraud['Процент_мошенничеств
ax3.set_title('Процент мошеннических транзакций по типам вендоров')
ax3.set_xlabel('Процент мошенничества')

# Тепловая карта категория vs тип
pivot_table = df_transactions.pivot_table(
    values='is_fraud',
    index='vendor_category',
    columns='vendor_type',
    aggfunc='mean'
) * 100

sns.heatmap(pivot_table, annot=True, fmt='.1f', cmap='Reds', ax=ax4)
ax4.set_title('Процент мошенничества: Категория vs Тип вендора')

plt.tight_layout()
plt.show()
```

```
Статистика мошенничества по категориям вендоров:
                Всего_транзакций  Мошеннических  Процент_мошенничества
vendor_category
Travel                    935790         187477              20.034089
Grocery                   934029         186987              20.019400
Gas                       935401         186829              19.973145
Restaurant                936178         186951              19.969600
Entertainment             936173         186890              19.963191
Education                 933542         186203              19.945862
Retail                    935883         186613              19.939779
Healthcare                936770         186769              19.937551
```





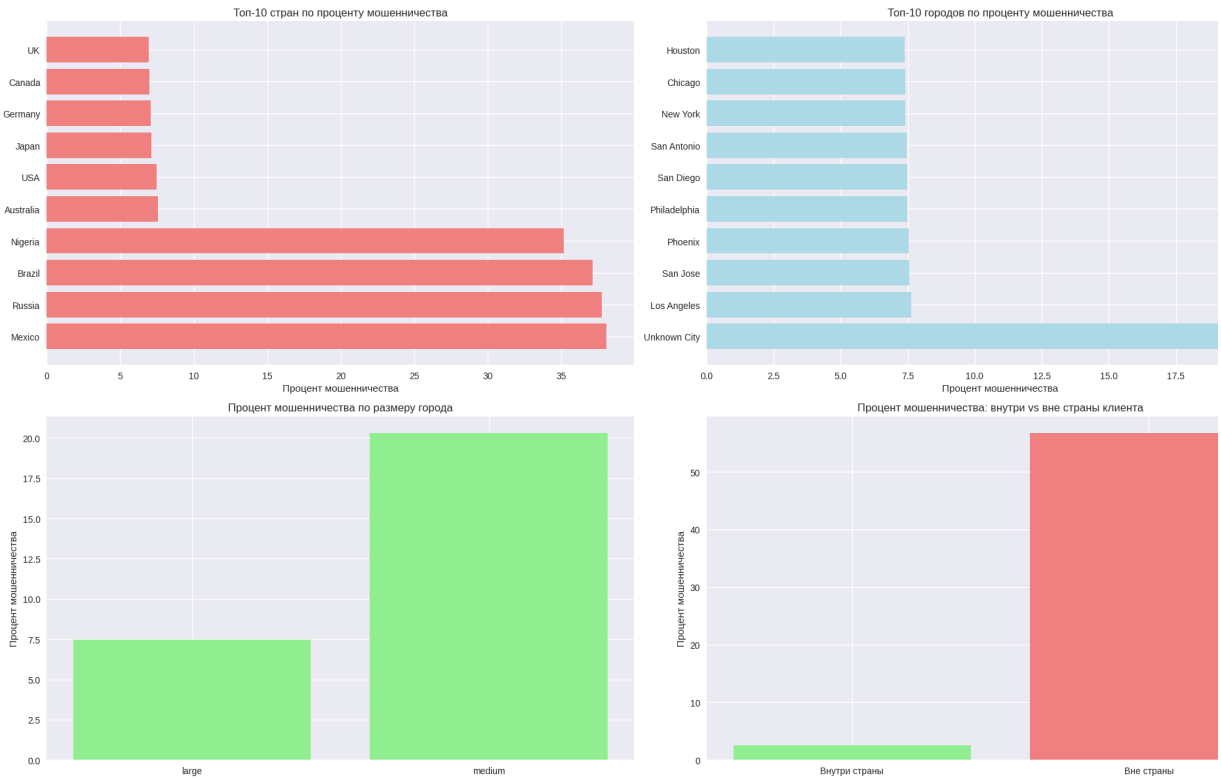




Анализ географических данных

In [10]:
```
# Статистика по странам
country_fraud = df_transactions.groupby('country')['is_fraud'].agg(['cour
country_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мо
country_fraud['Процент_мошенничества'] = country_fraud['Процент_мошенниче
country_fraud = country_fraud.sort_values('Процент_мошенничества', ascend

print("Топ-15 стран по проценту мошенничества:")
print(country_fraud.head(15))

# Статистика по городам
city_fraud = df_transactions.groupby('city')['is_fraud'].agg(['count', 's
city_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мошен
city_fraud['Процент_мошенничества'] = city_fraud['Процент_мошенничества']
city_fraud = city_fraud[city_fraud['Всего_транзакций'] >= 100]  # Минимум
city_fraud = city_fraud.sort_values('Процент_мошенничества', ascending=Fa

print("\nТоп-15 городов по проценту мошенничества (минимум 100 транзакций
print(city_fraud.head(15))

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Топ-10 стран по проценту мошенничества
top_countries = country_fraud.head(10)
ax1.barh(top_countries.index, top_countries['Процент_мошенничества'], col
ax1.set_title('Топ-10 стран по проценту мошенничества')
ax1.set_xlabel('Процент мошенничества')

# Топ-10 городов по проценту мошенничества
top_cities = city_fraud.head(10)
ax2.barh(top_cities.index, top_cities['Процент_мошенничества'], color='li
ax2.set_title('Топ-10 городов по проценту мошенничества')
ax2.set_xlabel('Процент мошенничества')

# Анализ размеров городов
city_size_fraud = df_transactions.groupby('city_size')['is_fraud'].agg(['
city_size_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_
city_size_fraud['Процент_мошенничества'] = city_size_fraud['Процент_мошен

ax3.bar(city_size_fraud.index, city_size_fraud['Процент_мошенничества'],
ax3.set_title('Процент мошенничества по размеру города')
ax3.set_ylabel('Процент мошенничества')

# Анализ операций вне страны клиента
outside_country_fraud = df_transactions.groupby('is_outside_home_country'
outside_country_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Пр
outside_country_fraud['Процент_мошенничества'] = outside_country_fraud['П

ax4.bar(['Внутри страны', 'Вне страны'], outside_country_fraud['Процент_м
        color=['lightgreen', 'lightcoral'])
ax4.set_title('Процент мошенничества: внутри vs вне страны клиента')
ax4.set_ylabel('Процент мошенничества')

plt.tight_layout()
plt.show()
```

```
Топ-15 стран по проценту мошенничества:
         Всего_транзакций  Мошеннических  Процент_мошенничества
country
Mexico             785704         298841              38.034807
Russia             793730         299425              37.723785
Brazil             804800         298629              37.105989
Nigeria            849840         298600              35.136026
Australia          496695          37652               7.580507
USA                500060          37312               7.461505
Japan              527393          37592               7.127891
Germany            524464          37205               7.093909
Canada             532632          37278               6.998828
UK                 538493          37345               6.935095
France             541287          37426               6.914262
Singapore          588668          37414               6.355705

Топ-15 городов по проценту мошенничества (минимум 100 транзакций):
             Всего_транзакций  Мошеннических  Процент_мошенничества
city
Unknown City          6983706        1457407              20.868676
Los Angeles             49494           3771               7.619105
San Jose                50015           3777               7.551734
Phoenix                 50333           3786               7.521904
Philadelphia            49914           3739               7.490884
San Diego               50425           3771               7.478433
San Antonio             50079           3736               7.460213
New York                49805           3696               7.420942
Chicago                 49912           3701               7.415050
Houston                 49957           3687               7.380347
Dallas                  50126           3648               7.277660
```

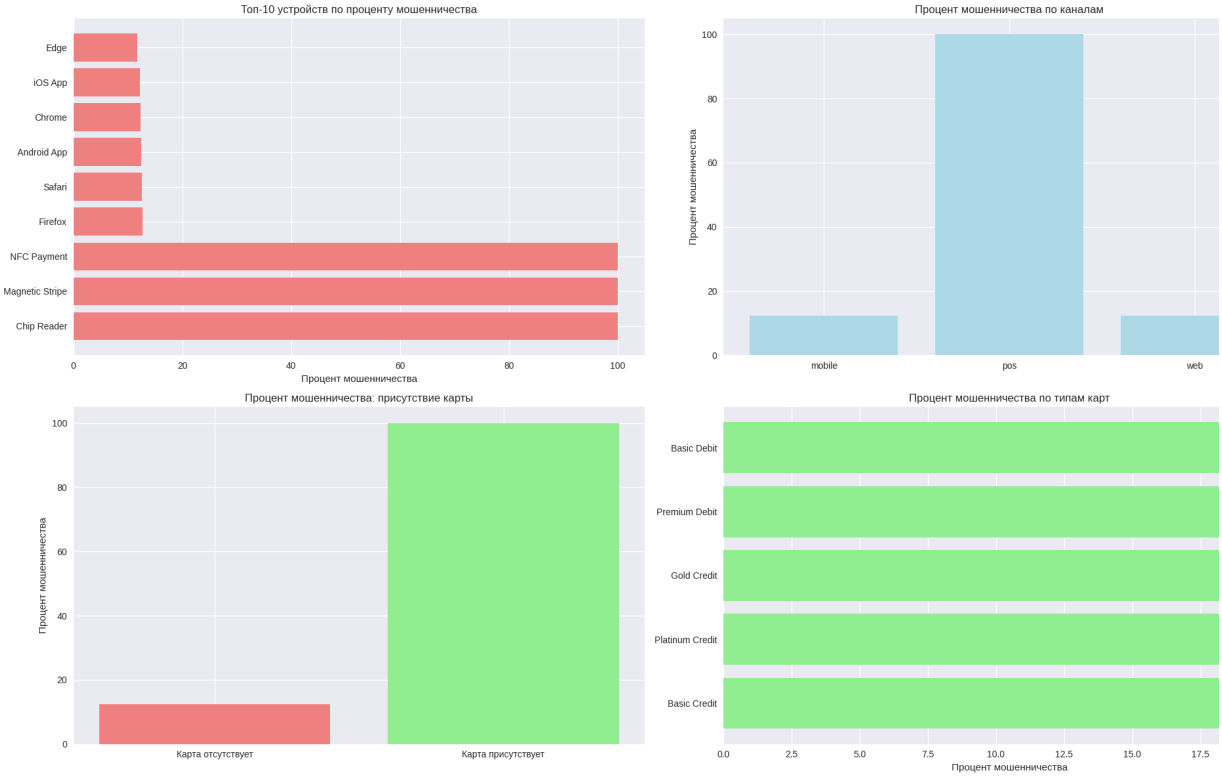


Анализ устройств и каналов

In [11]:
```
# Статистика по устройствам
device_fraud = df_transactions.groupby('device')['is_fraud'].agg(['count'
device_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мош
```

```
device_fraud['Процент_мошенничества'] = device_fraud['Процент_мошенничест
device_fraud = device_fraud[device_fraud['Всего_транзакций'] >= 100]
device_fraud = device_fraud.sort_values('Процент_мошенничества', ascendir

print("Статистика мошенничества по устройствам (минимум 100 транзакций):"
print(device_fraud)

# Статистика по каналам
channel_fraud = df_transactions.groupby('channel')['is_fraud'].agg(['cour
channel_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мо
channel_fraud['Процент_мошенничества'] = channel_fraud['Процент_мошенниче

print("\nСтатистика мошенничества по каналам:")
print(channel_fraud)

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Топ-10 устройств по проценту мошенничества
top_devices = device_fraud.head(10)
ax1.barh(top_devices.index, top_devices['Процент_мошенничества'], color='
ax1.set_title('Топ-10 устройств по проценту мошенничества')
ax1.set_xlabel('Процент мошенничества')

# Процент мошенничества по каналам
ax2.bar(channel_fraud.index, channel_fraud['Процент_мошенничества'], colo
ax2.set_title('Процент мошенничества по каналам')
ax2.set_ylabel('Процент мошенничества')

# Анализ присутствия карты
card_present_fraud = df_transactions.groupby('is_card_present')['is_fraud
card_present_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Проце
card_present_fraud['Процент_мошенничества'] = card_present_fraud['Процент

ax3.bar(['Карта отсутствует', 'Карта присутствует'], card_present_fraud['
        color=['lightcoral', 'lightgreen'])
ax3.set_title('Процент мошенничества: присутствие карты')
ax3.set_ylabel('Процент мошенничества')

# Анализ типов карт
card_type_fraud = df_transactions.groupby('card_type')['is_fraud'].agg(['
card_type_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_
card_type_fraud['Процент_мошенничества'] = card_type_fraud['Процент_мошен
card_type_fraud = card_type_fraud.sort_values('Процент_мошенничества', as

ax4.barh(card_type_fraud.index, card_type_fraud['Процент_мошенничества'],
ax4.set_title('Процент мошенничества по типам карт')
ax4.set_xlabel('Процент мошенничества')

plt.tight_layout()
plt.show()
```

```
Статистика мошенничества по устройствам (минимум 100 транзакций):
                Всего_транзакций  Мошеннических  Процент_мошенничества
device
Chip Reader               217324         217324             100.000000
Magnetic Stripe           217204         217204             100.000000
NFC Payment               216519         216519             100.000000
Firefox                  1120952         142171              12.683059
Safari                   1120245         141379              12.620364
Android App              1126117         140844              12.507049
Chrome                   1132384         140087              12.370980
iOS App                  1143461         140306              12.270292
Edge                     1189560         138885              11.675325

Статистика мошенничества по каналам:
         Всего_транзакций  Мошеннических  Процент_мошенничества
channel
mobile            2269578         281150              12.387765
pos                651047         651047             100.000000
web               4563141         562522              12.327517
```



Анализ активности за последний час

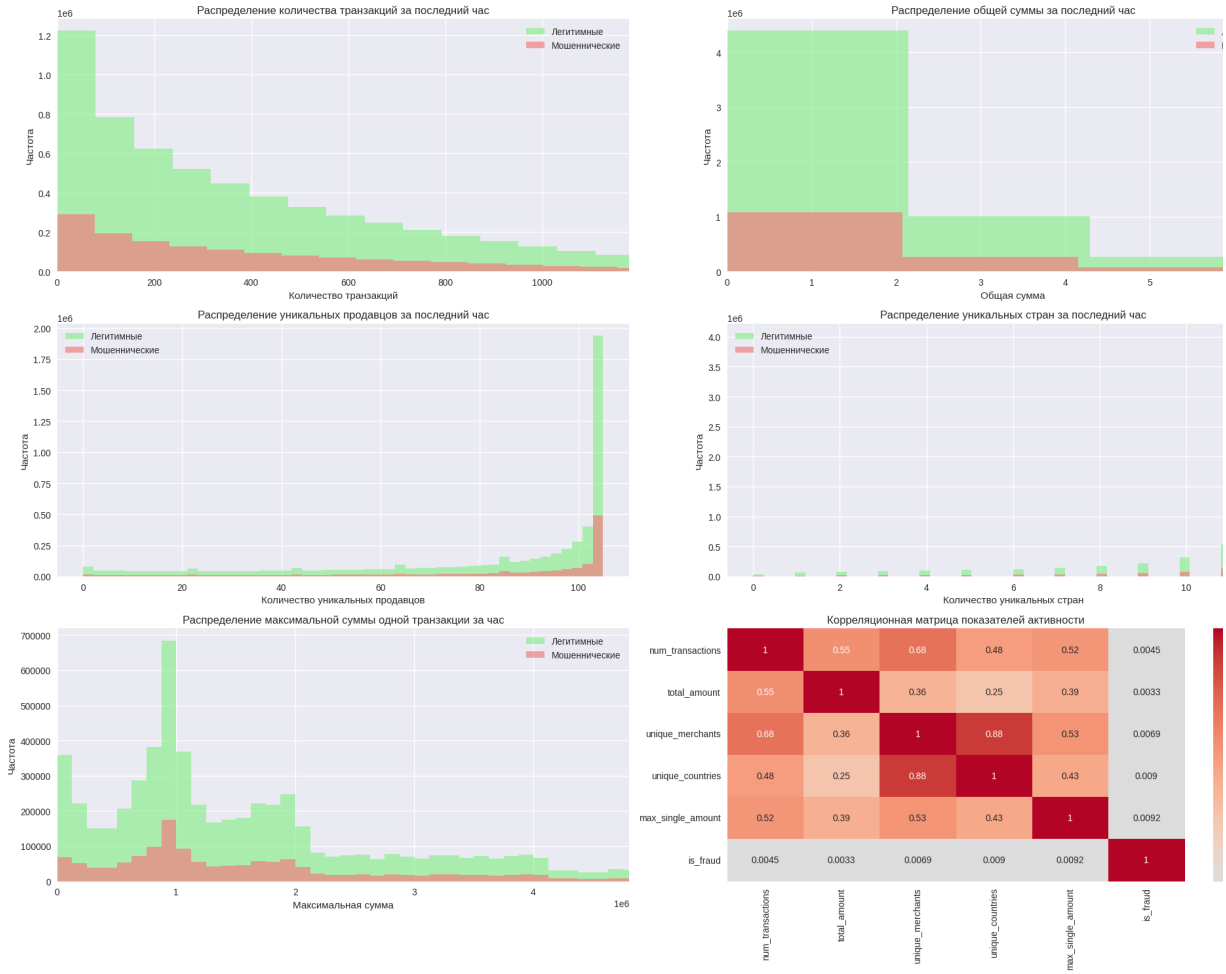
In [15]:

```
# Извлечение данных из структурированного поля
last_hour_data = df_transactions['last_hour_activity'].apply(pd.Series)
df_transactions = pd.concat([df_transactions, last_hour_data], axis=1)

# Статистика по показателям активности
activity_features = ['num_transactions', 'total_amount', 'unique_merchant

print("Статистика показателей активности за последний час:")
print(df_transactions[activity_features].describe())

# Корреляция с мошенничеством
correlations = df_transactions[activity_features + ['is_fraud']].corr()['
print("\nКорреляция показателей активности с мошенничеством:")
print(correlations)
```

```
# Визуализация
fig, ((ax1, ax2), (ax3, ax4), (ax5, ax6)) = plt.subplots(3, 2, figsize=(2

# Распределение количества транзакций за час
ax1.hist(df_transactions[df_transactions['is_fraud'] == False]['num_trans
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax1.hist(df_transactions[df_transactions['is_fraud'] == True]['num_transa
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax1.set_title('Распределение количества транзакций за последний час')
ax1.set_xlabel('Количество транзакций')
ax1.set_ylabel('Частота')
ax1.legend()
ax1.set_xlim(0, df_transactions['num_transactions'].quantile(0.95))

# Распределение общей суммы за час
ax2.hist(df_transactions[df_transactions['is_fraud'] == False]['total_amo
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax2.hist(df_transactions[df_transactions['is_fraud'] == True]['total_amou
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax2.set_title('Распределение общей суммы за последний час')
ax2.set_xlabel('Общая сумма')
ax2.set_ylabel('Частота')
ax2.legend()
ax2.set_xlim(0, df_transactions['total_amount'].quantile(0.95))

# Распределение уникальных продавцов
ax3.hist(df_transactions[df_transactions['is_fraud'] == False]['unique_me
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax3.hist(df_transactions[df_transactions['is_fraud'] == True]['unique_mer
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax3.set_title('Распределение уникальных продавцов за последний час')
ax3.set_xlabel('Количество уникальных продавцов')
ax3.set_ylabel('Частота')
ax3.legend()

# Распределение уникальных стран
ax4.hist(df_transactions[df_transactions['is_fraud'] == False]['unique_co
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax4.hist(df_transactions[df_transactions['is_fraud'] == True]['unique_cou
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax4.set_title('Распределение уникальных стран за последний час')
ax4.set_xlabel('Количество уникальных стран')
ax4.set_ylabel('Частота')
ax4.legend()

# Распределение максимальной суммы одной транзакции
ax5.hist(df_transactions[df_transactions['is_fraud'] == False]['max_singl
         bins=50, alpha=0.7, label='Легитимные', color='lightgreen')
ax5.hist(df_transactions[df_transactions['is_fraud'] == True]['max_single
         bins=50, alpha=0.7, label='Мошеннические', color='lightcoral')
ax5.set_title('Распределение максимальной суммы одной транзакции за час')
ax5.set_xlabel('Максимальная сумма')
ax5.set_ylabel('Частота')
ax5.legend()
ax5.set_xlim(0, df_transactions['max_single_amount'].quantile(0.95))

# Корреляционная матрица
corr_matrix = df_transactions[activity_features + ['is_fraud']].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0, ax=ax6)
```

```
ax6.set_title('Корреляционная матрица показателей активности')

plt.tight_layout()
plt.show()
```

```
Статистика показателей активности за последний час:
       num_transactions  total_amount  unique_merchants  unique_countries  \
count      7.483766e+06  7.483766e+06      7.483766e+06      7.483766e+06
mean       4.091429e+02  1.991719e+07      8.002226e+01      1.066260e+01
std        3.910964e+02  3.565890e+07      3.047287e+01      2.617777e+00
min        0.000000e+00  1.454232e-02      0.000000e+00      0.000000e+00
25%        1.050000e+02  3.367823e+06      6.300000e+01      1.100000e+01
50%        2.920000e+02  1.017851e+07      9.500000e+01      1.200000e+01
75%        6.060000e+02  2.273432e+07      1.040000e+02      1.200000e+01
max        3.962000e+03  1.072915e+09      1.050000e+02      1.200000e+01

       max_single_amount
count       7.483766e+06
mean        1.726432e+06
std         1.398325e+06
min         1.454232e-02
25%         8.015712e+05
50%         1.235738e+06
75%         2.290742e+06
max         6.253153e+06

Корреляция показателей активности с мошенничеством:
is_fraud             1.000000
max_single_amount    0.009226
unique_countries     0.009046
unique_merchants     0.006932
num_transactions     0.004506
total_amount         0.003332
Name: is_fraud, dtype: float64
```



Анализ валют

In [12]:
```
# Статистика по валютам
currency_fraud = df_transactions.groupby('currency')['is_fraud'].agg(['co
currency_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_м
currency_fraud['Процент_мошенничества'] = currency_fraud['Процент_мошенни
currency_fraud = currency_fraud.sort_values('Процент_мошенничества', asce

print("Статистика мошенничества по валютам:")
print(currency_fraud)

# Анализ обменных курсов
print("\nСтатистика обменных курсов:")
print(df_exchange.describe())

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Процент мошенничества по валютам
ax1.barh(currency_fraud.index, currency_fraud['Процент_мошенничества'], c
ax1.set_title('Процент мошеннических транзакций по валютам')
ax1.set_xlabel('Процент мошенничества')

# Количество транзакций по валютам
ax2.barh(currency_fraud.index, currency_fraud['Всего_транзакций'], color=
ax2.set_title('Общее количество транзакций по валютам')
ax2.set_xlabel('Количество транзакций')

# Динамика обменных курсов (выбираем несколько валют)
```

```
currencies_to_plot = ['EUR', 'GBP', 'JPY', 'RUB']
for currency in currencies_to_plot:
    if currency in df_exchange.columns:
        ax3.plot(df_exchange['date'], df_exchange[currency], label=currer
ax3.set_title('Динамика обменных курсов относительно USD')
ax3.set_xlabel('Дата')
ax3.set_ylabel('Курс к USD')
ax3.legend()
ax3.grid(True, alpha=0.3)

# Корреляция между валютами
currency_corr = df_exchange[['EUR', 'GBP', 'JPY', 'RUB', 'CAD', 'AUD']].c
sns.heatmap(currency_corr, annot=True, cmap='coolwarm', center=0, ax=ax4)
ax4.set_title('Корреляция между валютами')

plt.tight_layout()
plt.show()
```

```
Статистика мошенничества по валютам:
          Всего_транзакций  Мошеннических  Процент_мошенничества
currency
MXN                 785704         298841              38.034807
RUB                 793730         299425              37.723785
BRL                 804800         298629              37.105989
NGN                 849840         298600              35.136026
AUD                 496695          37652               7.580507
USD                 500060          37312               7.461505
JPY                 527393          37592               7.127891
EUR                1065751          74631               7.002668
CAD                 532632          37278               6.998828
GBP                 538493          37345               6.935095
SGD                 588668          37414               6.355705

Статистика обменных курсов:
             AUD        BRL        CAD        EUR        GBP         JPY  \
count  31.000000  31.000000  31.000000  31.000000  31.000000   31.000000
mean    1.486451   5.599606   1.373282   0.916183   0.764806  149.313721
std     0.021309   0.104725   0.013313   0.008551   0.006681    2.686216
min     1.442917   5.425444   1.348063   0.895591   0.746956  142.573268
25%     1.475624   5.486880   1.360292   0.910902   0.762839  148.544161
50%     1.487595   5.625377   1.377943   0.916792   0.765915  149.168474
75%     1.499793   5.689843   1.382771   0.923870   0.769698  151.214272
max     1.522229   5.761654   1.390965   0.927316   0.773928  153.800613

             MXN          NGN        RUB        SGD   USD
count  31.000000    31.000000  31.000000  31.000000  31.0
mean   19.650422  1640.017070  96.003479   1.308041   1.0
std     0.283454    15.192761   1.073514   0.011290   0.0
min    19.263497  1619.450022  92.898519   1.280156   1.0
25%    19.360603  1630.708907  95.450875   1.303973   1.0
50%    19.694724  1639.315783  95.922769   1.308048   1.0
75%    19.894018  1643.959480  96.868840   1.316452   1.0
max    20.048756  1670.694524  97.501463   1.324596   1.0
```

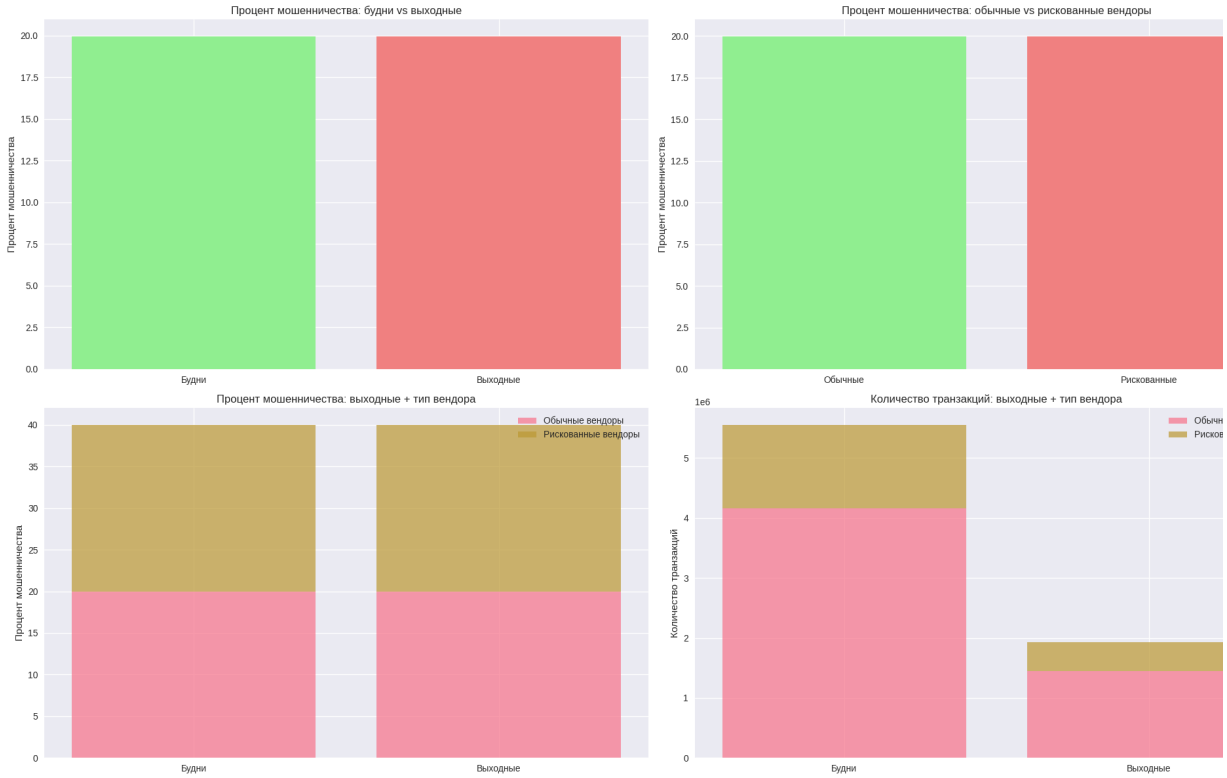
Анализ выходных дней

In [13]:

```
weekend_fraud = df_transactions.groupby('is_weekend')['is_fraud'].agg(['c
weekend_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процент_мс
weekend_fraud['Процент_мошенничества'] = weekend_fraud['Процент_мошенниче

print("Статистика мошенничества по выходным дням:")
print(weekend_fraud)

# Анализ рискованных вендоров
risk_vendor_fraud = df_transactions.groupby('is_high_risk_vendor')['is_fr
risk_vendor_fraud.columns = ['Всего_транзакций', 'Мошеннических', 'Процен
risk_vendor_fraud['Процент_мошенничества'] = risk_vendor_fraud['Процент_м

print("\nСтатистика мошенничества по рискованным вендорам:")
print(risk_vendor_fraud)

# Визуализация
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(20, 12))

# Выходные дни
ax1.bar(['Будни', 'Выходные'], weekend_fraud['Процент_мошенничества'],
        color=['lightgreen', 'lightcoral'])
ax1.set_title('Процент мошенничества: будни vs выходные')
ax1.set_ylabel('Процент мошенничества')

# Рискованные вендоры
ax2.bar(['Обычные', 'Рискованные'], risk_vendor_fraud['Процент_мошенничес
        color=['lightgreen', 'lightcoral'])
ax2.set_title('Процент мошенничества: обычные vs рискованные вендоры')
ax2.set_ylabel('Процент мошенничества')

# Комбинация выходных и рискованных вендоров
weekend_risk_fraud = df_transactions.groupby(['is_weekend', 'is_high_risk
weekend_risk_fraud = weekend_risk_fraud.unstack()
```

```
ax3.bar(['Будни', 'Выходные'], weekend_risk_fraud[False], label='Обычные
ax3.bar(['Будни', 'Выходные'], weekend_risk_fraud[True], bottom=weekend_r
        label='Рискованные вендоры', alpha=0.7)
ax3.set_title('Процент мошенничества: выходные + тип вендора')
ax3.set_ylabel('Процент мошенничества')
ax3.legend()

# Количество транзакций по комбинации
weekend_risk_count = df_transactions.groupby(['is_weekend', 'is_high_risk
ax4.bar(['Будни', 'Выходные'], weekend_risk_count[False], label='Обычные
ax4.bar(['Будни', 'Выходные'], weekend_risk_count[True], bottom=weekend_r
        label='Рискованные вендоры', alpha=0.7)
ax4.set_title('Количество транзакций: выходные + тип вендора')
ax4.set_ylabel('Количество транзакций')
ax4.legend()

plt.tight_layout()
plt.show()
```

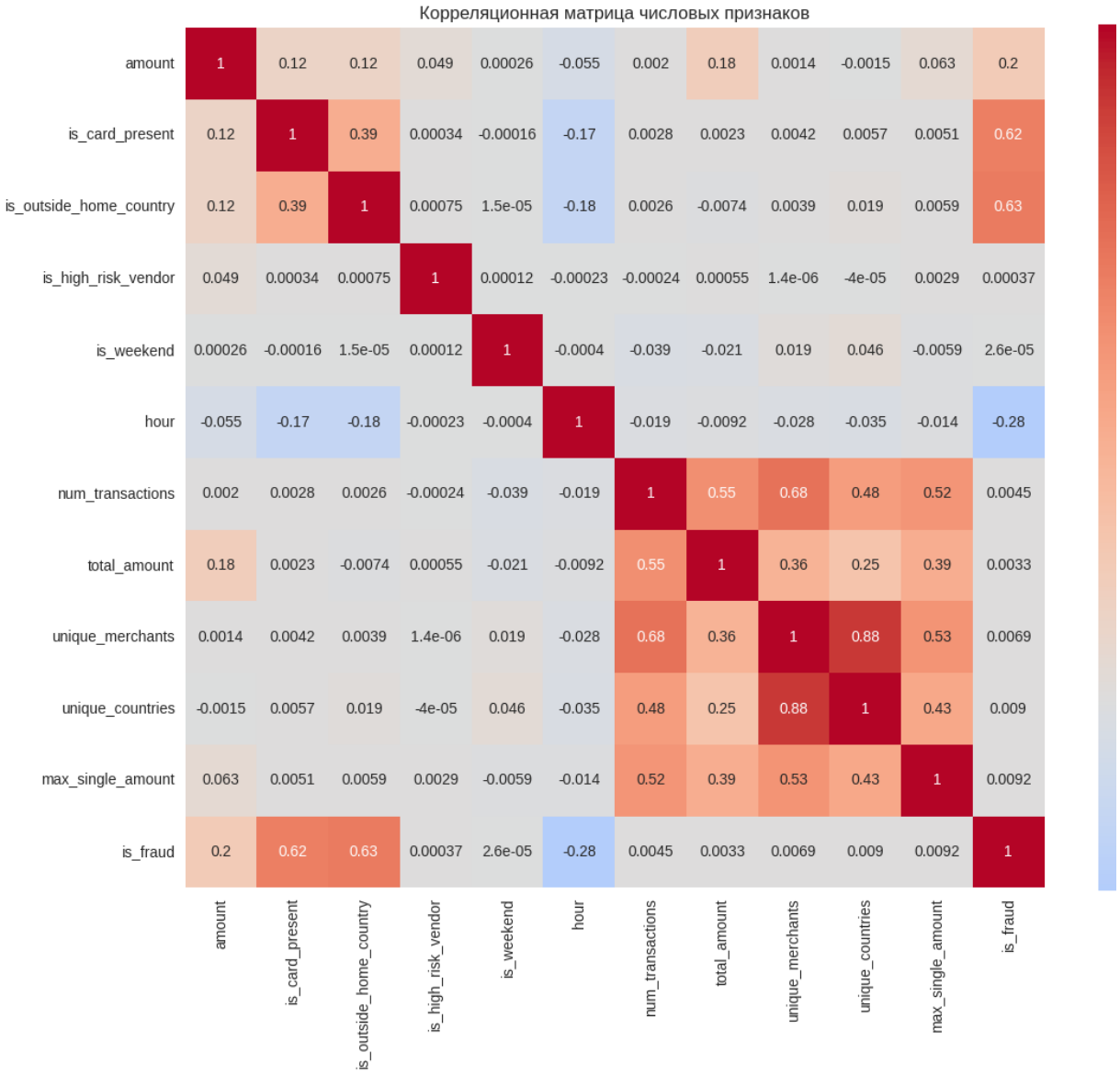
```
Статистика мошенничества по выходным дням:
            Всего_транзакций  Мошеннических  Процент_мошенничества
is_weekend
False                5554103        1109277              19.972208
True                 1929663         385442              19.974576

Статистика мошенничества по рискованным вендорам:
                     Всего_транзакций  Мошеннических  Процент_мошенничества
is_high_risk_vendor
False                         5611803        1120352              19.964208
True                          1871963         374367              19.998632
```



Корреляционный анализ

In [16]:
```
# Подготовка данных для корреляционного анализа
correlation_features = [
```

```
    'amount', 'is_card_present', 'is_outside_home_country',
    'is_high_risk_vendor', 'is_weekend', 'hour',
    'num_transactions', 'total_amount', 'unique_merchants',
    'unique_countries', 'max_single_amount', 'is_fraud'
]

# Создание корреляционной матрицы
corr_matrix = df_transactions[correlation_features].corr()

# Визуализация
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0, square=Tr
plt.title('Корреляционная матрица числовых признаков')
plt.tight_layout()
plt.show()

# Топ корреляций с целевой переменной
correlations_with_fraud = corr_matrix['is_fraud'].abs().sort_values(ascer
print("Топ-10 признаков по корреляции с мошенничеством:")
print(correlations_with_fraud.head(10))
```



```
Топ-10 признаков по корреляции с мошенничеством:
is_fraud                  1.000000
is_outside_home_country   0.634459
is_card_present           0.617886
hour                      0.279624
amount                    0.199342
max_single_amount         0.009226
unique_countries          0.009046
unique_merchants          0.006932
num_transactions          0.004506
total_amount              0.003332
Name: is_fraud, dtype: float64
```

Продуктовые и технические гипотезы

Продуктовые гипотезы

1. Географические паттерны мошенничества
    - Гипотеза: Мошенничество чаще происходит в определенных странах и городах
    - Ценность: Фокус на высокорисковых регионах, персонализация проверок
2. Временные паттерны активности
    - Гипотеза: Мошеннические транзакции имеют специфические временные паттерны
    - Ценность: Улучшение алгоритмов в реальном времени, оптимизация ресурсов
3. Категории вендоров и риски
    - Гипотеза: Определенные категории вендоров более подвержены мошенничеству
    - Ценность: Дифференцированный подход к проверкам, снижение false positive
4. Поведенческие паттерны клиентов
    - Гипотеза: Аномальная активность за короткий период указывает на мошенничество
    - Ценность: Раннее выявление, улучшение пользовательского опыта
5. Устройства и каналы
    - Гипотеза: Определенные устройства и каналы более уязвимы для мошенничества
    - Ценность: Улучшение безопасности, фокус на проблемных каналах

Технические гипотезы

1. Feature Engineering
    - Создание новых признаков на основе временных паттернов
    - Агрегация транзакций по клиентам и временным окнам
    - Нормализация сумм по валютам
2. Моделирование
    - Использование ансамблевых методов (Random Forest, XGBoost)
    - Применение методов для несбалансированных данных
    - Временные модели (LSTM, GRU) для последовательностей транзакций

3. Оптимизация
    - Балансировка классов (SMOTE, undersampling)
    - Подбор порогов классификации
    - Кросс-валидация с временными разбиениями
4. Мониторинг
    - A/B тестирование новых алгоритмов
    - Мониторинг drift'a данных
    - Автоматическое обновление моделей

Бизнес ценность

1. Снижение потерь: Уменьшение финансовых потерь от мошенничества
2. Улучшение UX: Снижение количества ложных срабатываний
3. Оптимизация ресурсов: Фокус на высокорисковых транзакциях
4. Конкурентное преимущество: Более безопасная платформа
5. Соответствие регуляторным требованиям: Улучшение compliance