```
from google.colab import files

# Upload the file
uploaded = files.upload()
```

Choose Files Day_10_ba…ng_data.csv

- **Day_10_banking_data.csv**(text/csv) - 1285 bytes, last modified: 1/24/2025 - 100% done

```
Saving Day_10_banking_data.csv to Day_10_banking_data.csv
```

```
import pandas as pd
```

```
fp='Day_10_banking_data.csv'
banking_data=pd.read_csv(fp)
filtered_transactions = banking_data[banking_data['Transaction_Amount'] <= 2000]
print("Rows where Transaction_Amount is less than or equal to 2000:")
print(filtered_transactions)
```

```
Rows where Transaction_Amount is less than or equal to 2000:
          Date      Account_Type    Branch Transaction_Type  \
0   2023-01-19     Fixed Deposit   Central     Loan Payment
1   2023-01-16           Current    Uptown       Withdrawal
3   2023-01-18           Savings    Uptown     Loan Payment
6   2023-01-04 Recurring Deposit   Central   Service Charge
9   2023-01-11     Fixed Deposit  Downtown          Deposit
10  2023-01-13 Recurring Deposit   Central          Deposit
11  2023-01-08     Fixed Deposit  Suburban   Service Charge
12  2023-01-15     Fixed Deposit    Uptown   Service Charge
18  2023-01-12 Recurring Deposit  Suburban   Service Charge

    Transaction_Amount  Account_Balance
0               985.51          6839.59
1               641.43          8908.39
3              1914.60          5776.63
6              1621.82          6465.79
9              1529.59          2592.16
10              846.41          6443.14
```

```
11              1803.88          6560.58
12              1225.50          4224.47
18              1339.57          8666.74
```

```
loan_payment_filter = banking_data[
    (banking_data['Transaction_Type'] == 'Loan Payment') &
    (banking_data['Account_Balance'] > 5000)
]
print("\nRows where Transaction_Type is 'Loan Payment' and Account_Balance > 5000:")
print(loan_payment_filter)
```

```
Rows where Transaction_Type is 'Loan Payment' and Account_Balance > 5000:
          Date       Account_Type   Branch Transaction_Type  \
0   2023-01-19      Fixed Deposit  Central     Loan Payment
2   2023-01-10            Current   Uptown     Loan Payment
3   2023-01-18            Savings   Uptown     Loan Payment
7   2023-01-09            Current  Central     Loan Payment
13  2023-01-05  Recurring Deposit  Central     Loan Payment
17  2023-01-07            Current  Central     Loan Payment

    Transaction_Amount  Account_Balance
0               985.51          6839.59
2              3363.85         12428.67
3              1914.60          5776.63
7              2346.72         10708.85
13             4683.64          6762.43
17             4116.52          9785.64
```

```
uptown_transactions = banking_data[banking_data['Branch'] == 'Uptown']
print("\nTransactions made in the 'Uptown' branch:")
print(uptown_transactions)
```

```
Transactions made in the 'Uptown' branch:
          Date   Account_Type  Branch Transaction_Type  Transaction_Amount  \
1   2023-01-16        Current  Uptown       Withdrawal              641.43
2   2023-01-10        Current  Uptown     Loan Payment             3363.85
```

```
3   2023-01-18        Savings  Uptown     Loan Payment             1914.60
12  2023-01-15  Fixed Deposit  Uptown   Service Charge             1225.50

    Account_Balance
1           8908.39
2          12428.67
3           5776.63
12          4224.47
```

```
banking_data['Transaction_Fee'] = banking_data['Transaction_Amount'] * 0.02

# Create a new column Balance_Status
banking_data['Balance_Status'] = banking_data['Account_Balance'].apply(
    lambda x: 'High Balance' if x > 5000 else 'Low Balance'
)

print("\nData with new columns (Transaction_Fee and Balance_Status):")
print(banking_data[['Transaction_Amount', 'Transaction_Fee', 'Account_Balance', 'Balance_Status']])
```

```
Data with new columns (Transaction_Fee and Balance_Status):
    Transaction_Amount  Transaction_Fee  Account_Balance Balance_Status
0               985.51          19.7102          6839.59   High Balance
1               641.43          12.8286          8908.39   High Balance
2              3363.85          67.2770         12428.67   High Balance
3              1914.60          38.2920          5776.63   High Balance
4              2788.57          55.7714          4779.04    Low Balance
5              4584.05          91.6810          7635.47   High Balance
6              1621.82          32.4364          6465.79   High Balance
7              2346.72          46.9344         10708.85   High Balance
8              3899.98          77.9996         12646.56   High Balance
9              1529.59          30.5918          2592.16    Low Balance
10              846.41          16.9282          6443.14   High Balance
11             1803.88          36.0776          6560.58   High Balance
12             1225.50          24.5100          4224.47    Low Balance
13             4683.64          93.6728          6762.43   High Balance
14             4136.54          82.7308          8175.08   High Balance
15             3350.32          67.0064         12836.51   High Balance
16             4421.57          88.4314          8330.40   High Balance
```

```
        17              4116.52          82.3304          9785.64   High Balance
        18              1339.57          26.7914          8666.74   High Balance
        19              4516.52          90.3304          8789.19   High Balance
```

Start coding or generate with AI.