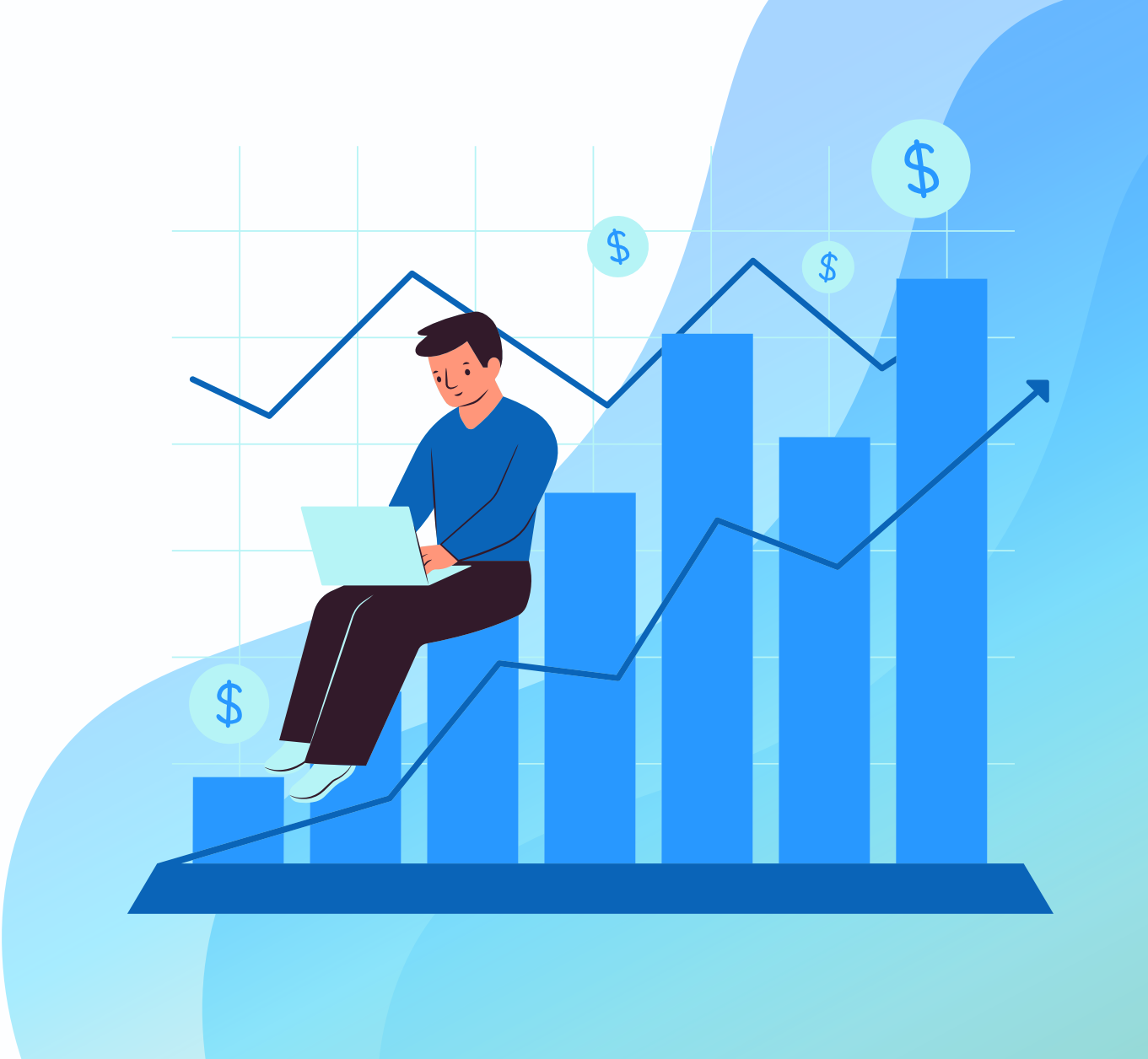
@LINKEDLN @GITHUB

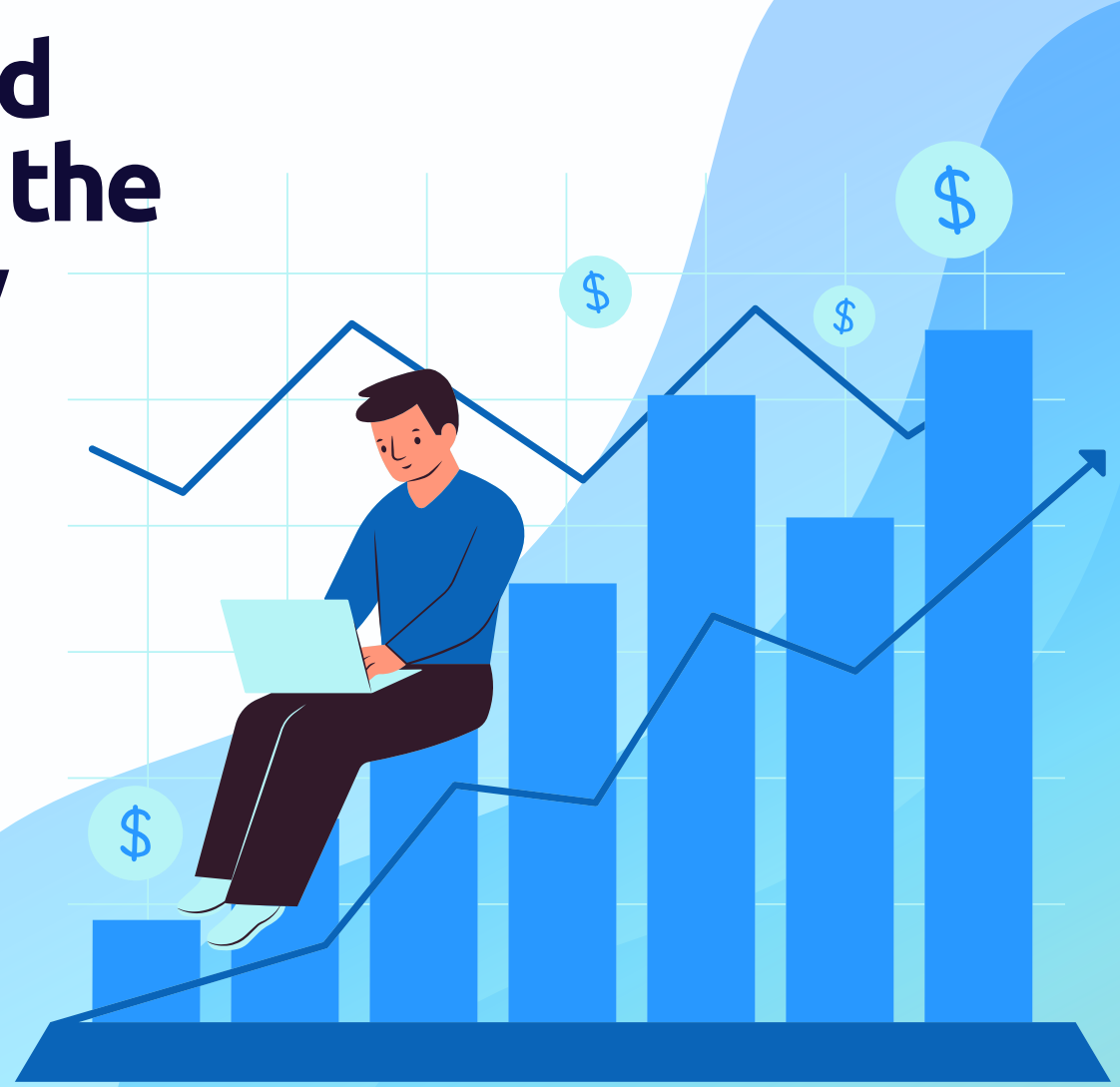
AMAZON STOCKS ANALYSIS

BY SQL



Hi, My name is Harshal and i've standarised / Analyze the data of Amazon Stocks by SQL.

- **Title: Amazon Stock Analysis**
- **Subtitle: A Comprehensive SQL-Based Analysis**
- **Presented By:** Harshal Kate



Data Overview

- Columns: Date, Open, High, Low, Close, Adj Close, Volume
- Data Source: Kaggle

Objectives

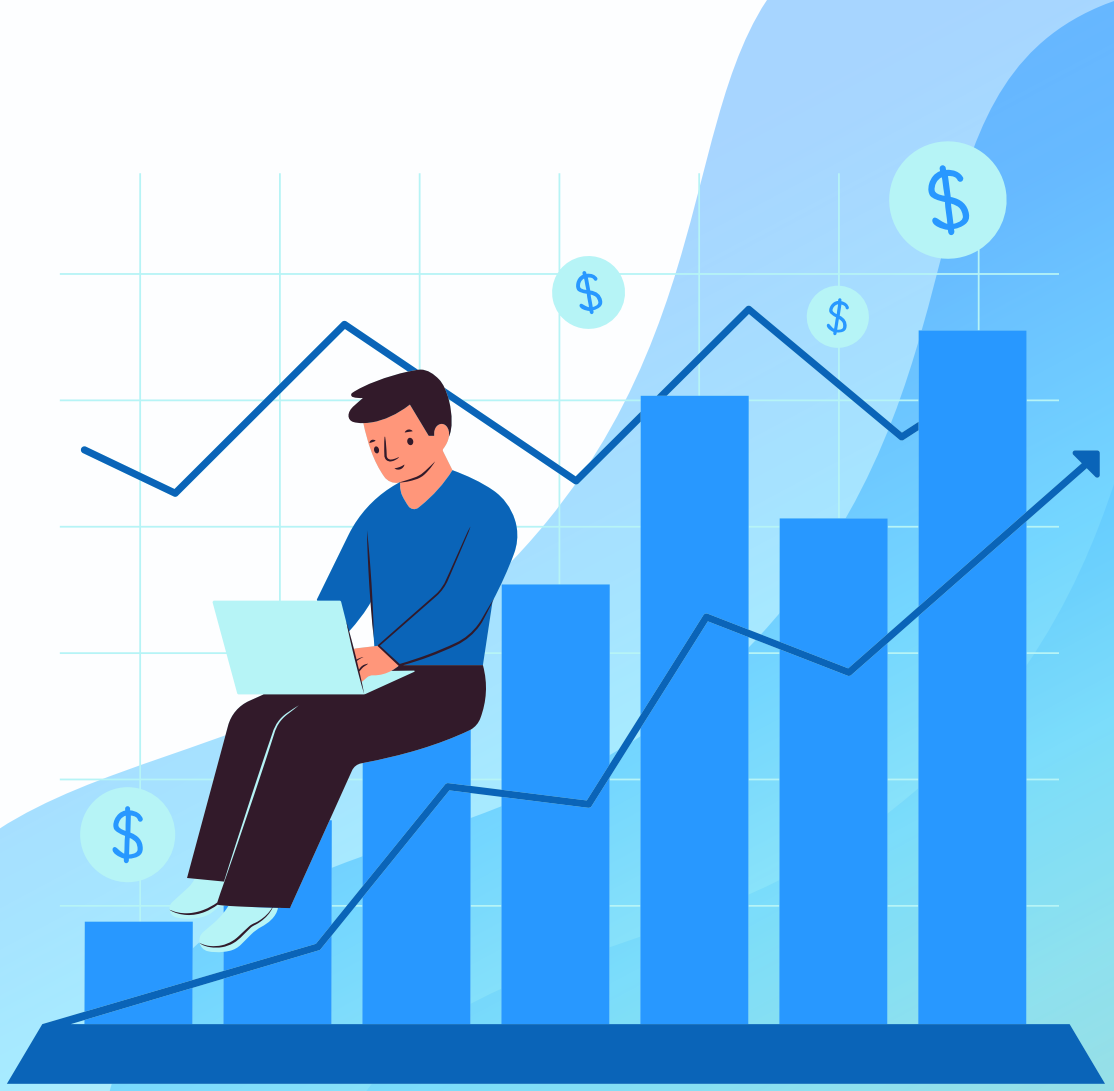
- Analyze historical stock performance
- Identify trends and patterns
- Provide insights for better investment decisions

Data Preparation

- Data Cleaning: Ensured data consistency and removed duplicates
- Data Transformation: Calculated additional metrics like daily returns

Key Metrics

- Open & Close Prices: Tracked opening and closing prices over time
- High & Low Prices: Identified daily price ranges
- Volume: Analyzed trading volume trends



Insights & Visualizations

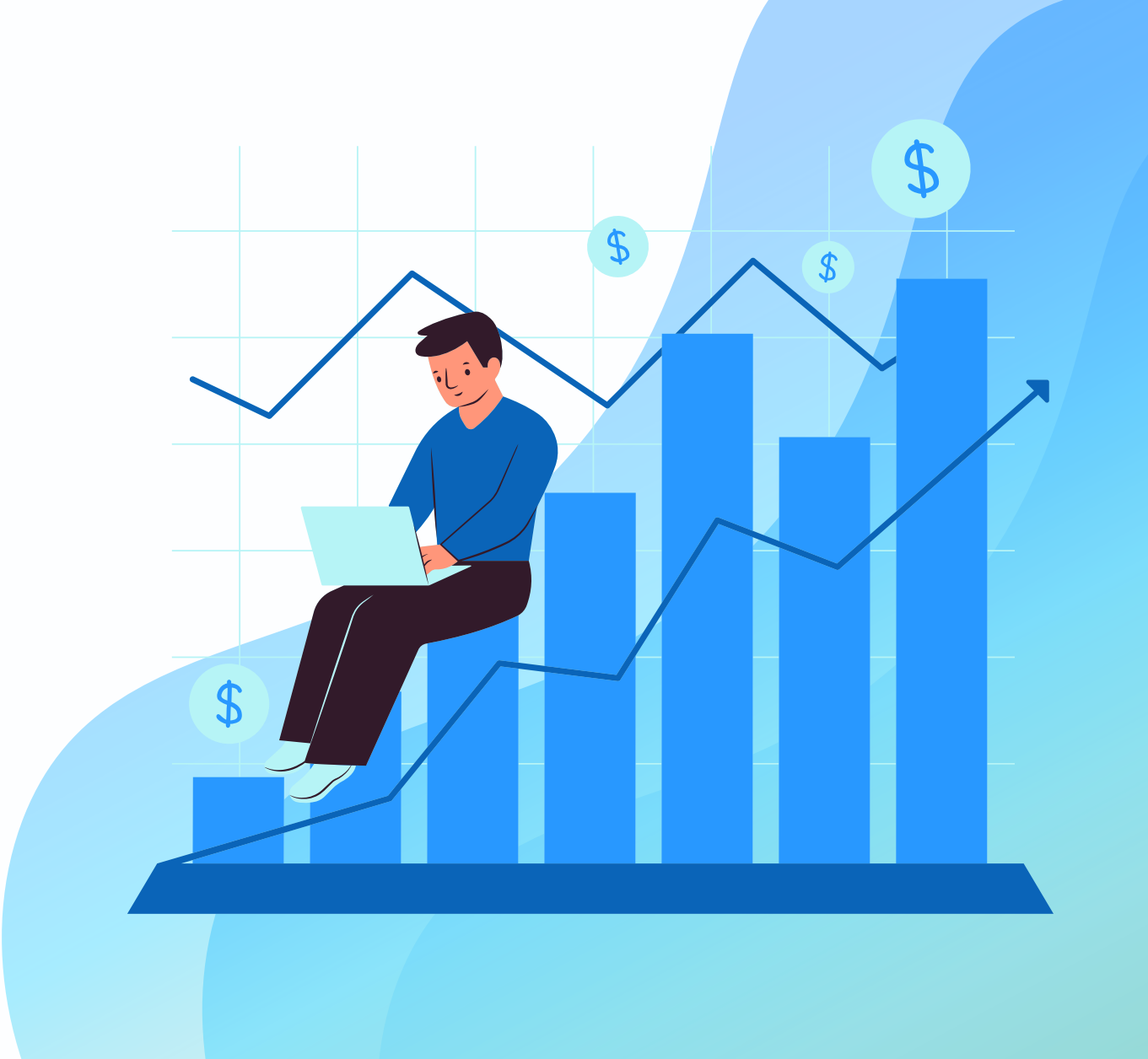
- Price Trends: Graph showing Open, High, Low, Close prices over time
- Volume Trends: Chart of trading volumes over time
- Daily Returns: Histogram of daily stock returns

Key Findings

- Identified significant price fluctuations on [specific dates]
- Noted highest trading volumes in [specific months/years]
- Observed consistent growth/decline trends over

Conclusion

- Summary: Key insights from the analysis
- Recommendations: Investment strategies based on Past/Present data performance

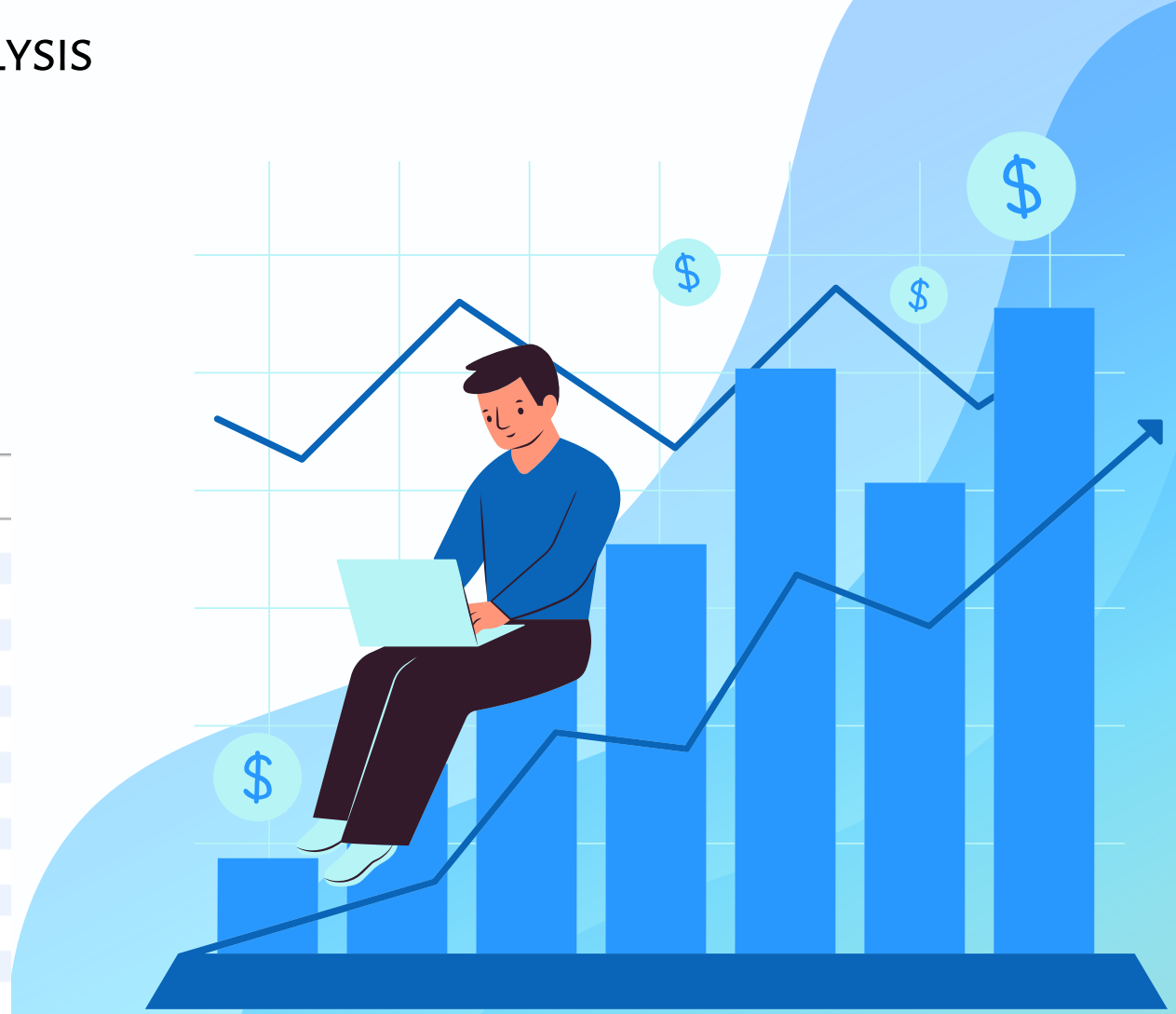


- CREATED THE DATABASE FOR THE ANALYSIS

```
Use Amzn;

select *
from amzn;
```

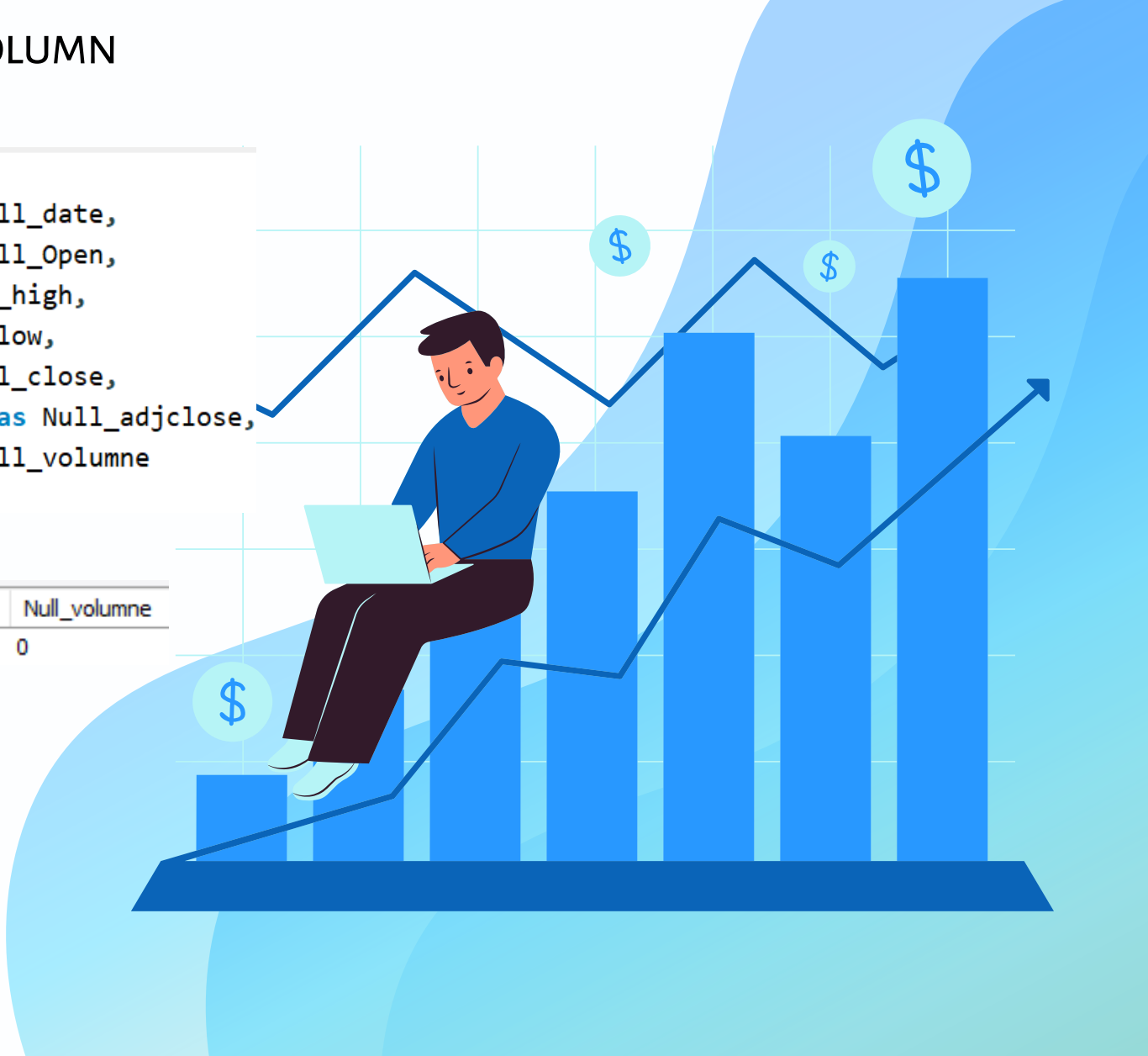
Date	Open	High	Low	Close	Adj Close	Volume
1997-05-15	0.121875	0.125	0.096354	0.097917	0.097917	1443120000
1997-05-16	0.098438	0.098958	0.085417	0.086458	0.086458	294000000
1997-05-19	0.088021	0.088542	0.08125	0.085417	0.085417	122136000
1997-05-20	0.086458	0.0875	0.081771	0.081771	0.081771	109344000
1997-05-21	0.081771	0.082292	0.06875	0.071354	0.071354	377064000
1997-05-22	0.071875	0.072396	0.065625	0.069792	0.069792	235536000
1997-05-23	0.070313	0.076042	0.066667	0.075	0.075	318744000
1997-05-27	0.075521	0.082292	0.072917	0.079167	0.079167	173952000
1997-05-28	0.08125	0.081771	0.076563	0.076563	0.076563	91488000
1997-05-29	0.077083	0.077083	0.073958	0.07526	0.07526	69456000
1997-05-30	0.075	0.075521	0.073958	0.075	0.075	51888000
1997-06-02	0.075521	0.076563	0.075	0.075521	0.075521	11832000
1997-06-03	0.076563	0.076563	0.073958	0.073958	0.073958	23664000
1997-06-04	0.073958	0.074479	0.069792	0.070833	0.070833	61608000
1997-06-05	0.070833	0.077083	0.06875	0.077083	0.077083	113448000
1997-06-06	0.075781	0.085417	0.075521	0.082813	0.082813	156144000



- CHECKING THE NULL VALUESS IN DATA COLUMN

```sql
select
sum(case when `Date` is null then 1 else 0 end) as Null_date,
sum(case when `Open` is null then 1 else 0 end) as Null_Open,
sum(case when High is null then 1 else 0 end) as Null_high,
sum(case when Low is null then 1 else 0 end) as Null_low,
sum(case when Close is null then 1 else 0 end) as Null_close,
sum(case when `Adj Close` is null then 1 else 0 end) as Null_adjclose,
sum(case when Volume is null then 1 else 0 end) as Null_volumne
from amzn;
```

	Null_date	Null_Open	Null_high	Null_low	Null_close	Null_adjclose	Null_volumne
▶	0	0	0	0	0	0	0

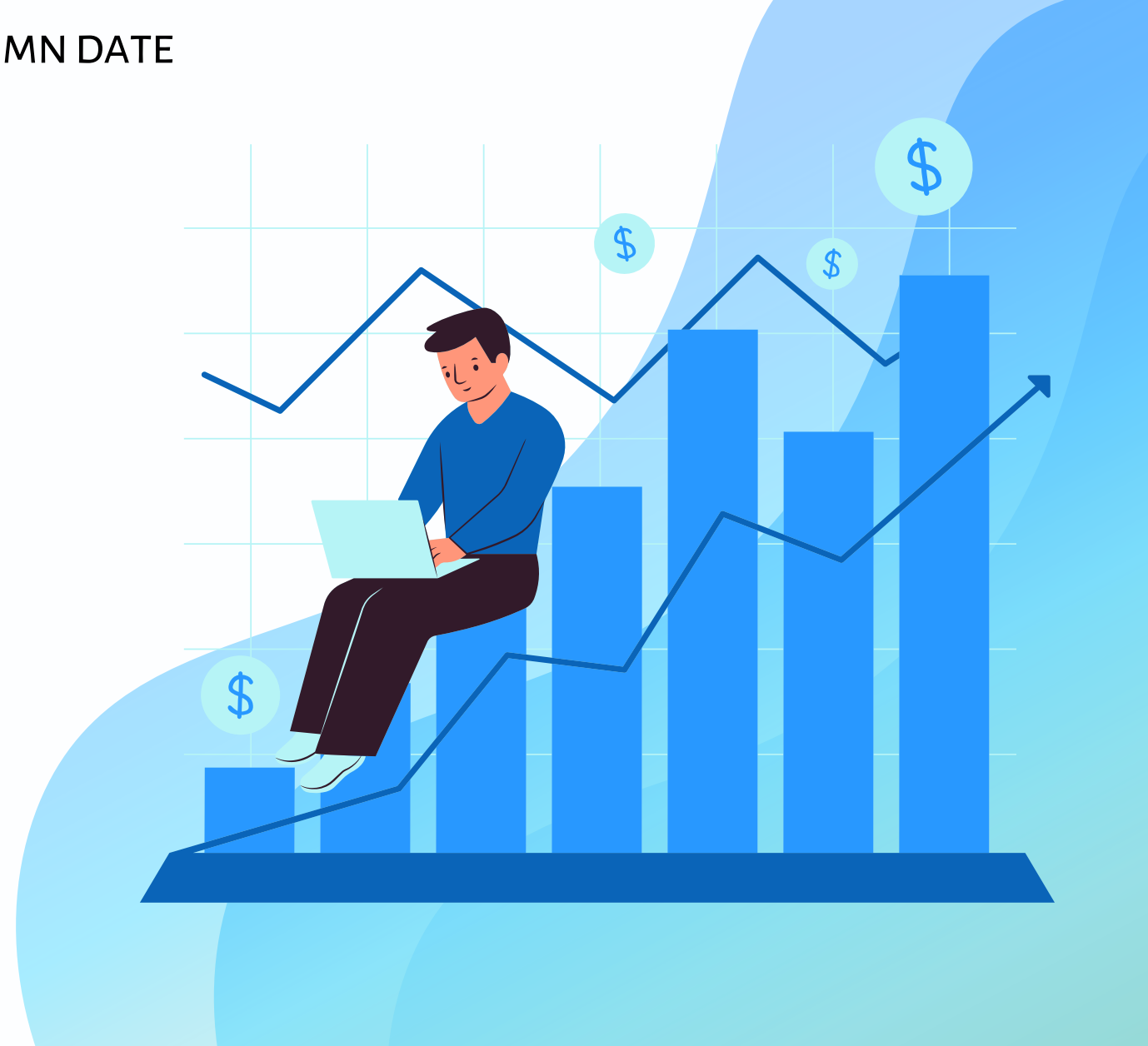


- # DATA STANDARDIZATION DATE CLOUMN DATE FORMAT

```
update amzn
set date = str_to_date(`Date`, '%Y-%m-%d');

select *
from amzn;
```

Date	Open	High	Low	Close	Adj Close	Volume
1997-05-15	0.121875	0.125	0.096354	0.097917	0.097917	1443120000
1997-05-16	0.098438	0.098958	0.085417	0.086458	0.086458	294000000
1997-05-19	0.088021	0.088542	0.08125	0.085417	0.085417	122136000
1997-05-20	0.086458	0.0875	0.081771	0.081771	0.081771	109344000
1997-05-21	0.081771	0.082292	0.06875	0.071354	0.071354	377064000
1997-05-22	0.071875	0.072396	0.065625	0.069792	0.069792	235536000
1997-05-23	0.070313	0.076042	0.066667	0.075	0.075	318744000
1997-05-27	0.075521	0.082292	0.072917	0.079167	0.079167	173952000
1997-05-28	0.08125	0.081771	0.076563	0.076563	0.076563	91488000
1997-05-29	0.077083	0.077083	0.073958	0.07526	0.07526	69456000
1997-05-30	0.075	0.075521	0.073958	0.075	0.075	51888000
1997-06-02	0.075521	0.076563	0.075	0.075521	0.075521	11832000
1997-06-03	0.076563	0.076563	0.073958	0.073958	0.073958	23664000
1997-06-04	0.073958	0.074479	0.069792	0.070833	0.070833	61608000
1997-06-05	0.070833	0.077083	0.06875	0.077083	0.077083	113448000
1997-06-06	0.075781	0.085417	0.075521	0.082813	0.082813	156144000

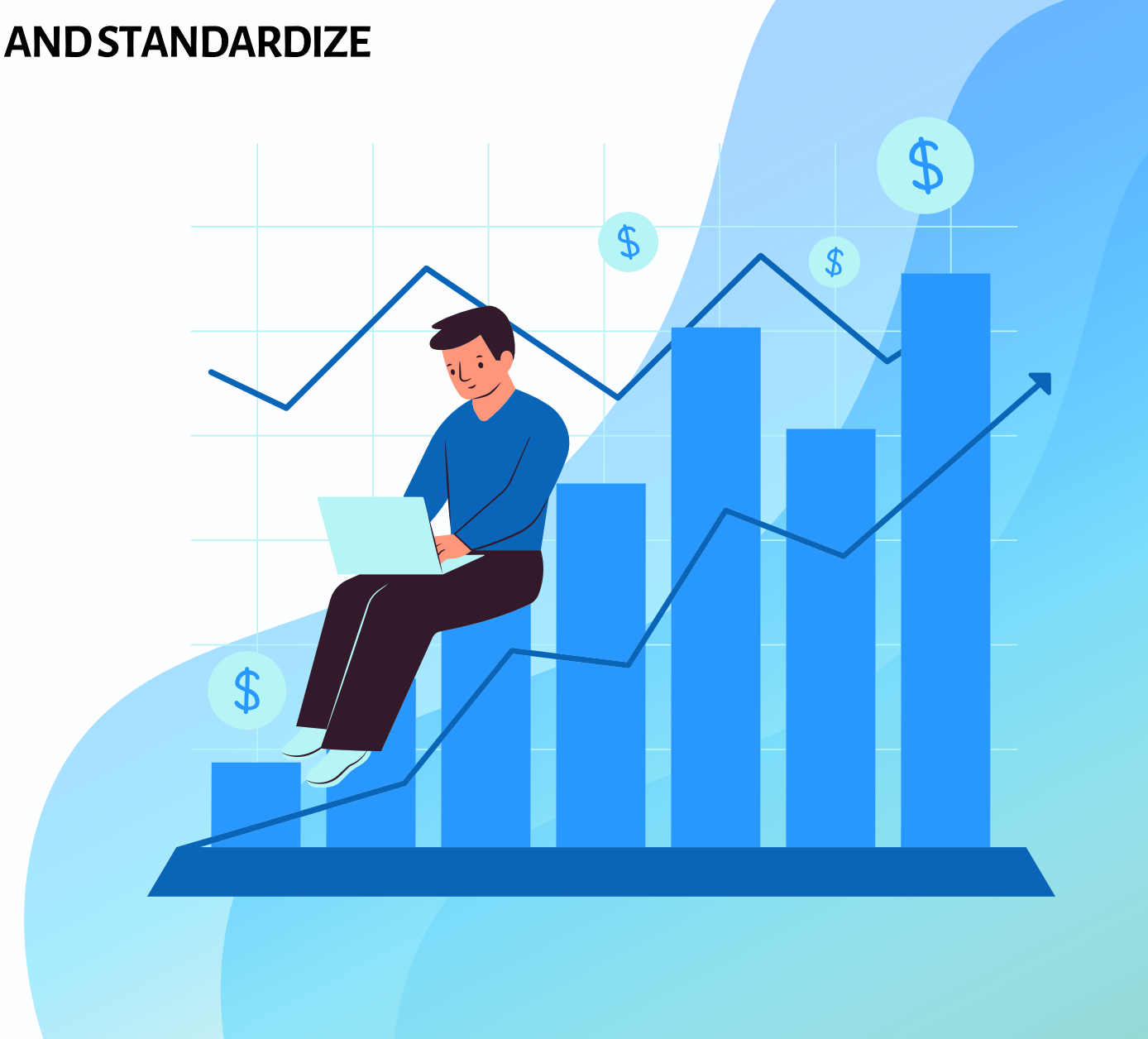


- **EDA OF THE ABOVE DAT AFTER CLEANING AND STANDARDIZE**

1. Summary statistics of the open high low

```
SELECT
    MIN(`Open`) AS min_open,
    MAX(High) AS max_high,
    AVG(`Close`) AS avg_Close
FROM
    amzn;
```

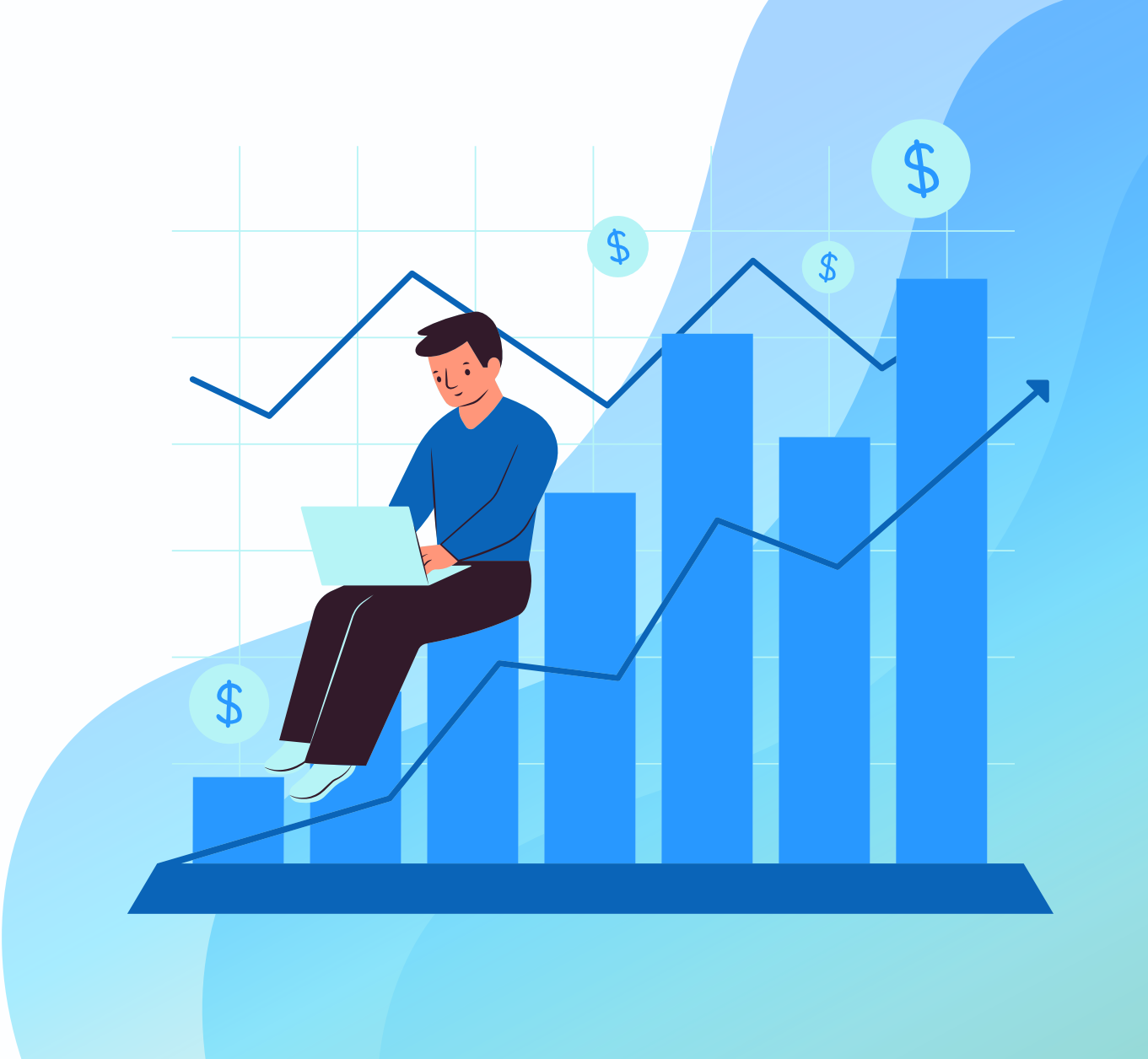
	min_open	max_high	avg_Close
▶	0.070313	188.654007	31.599739606200124



- Count of trading days of the stocks

```
SELECT
    COUNT(*) AS total_days
FROM
    amzn;
```

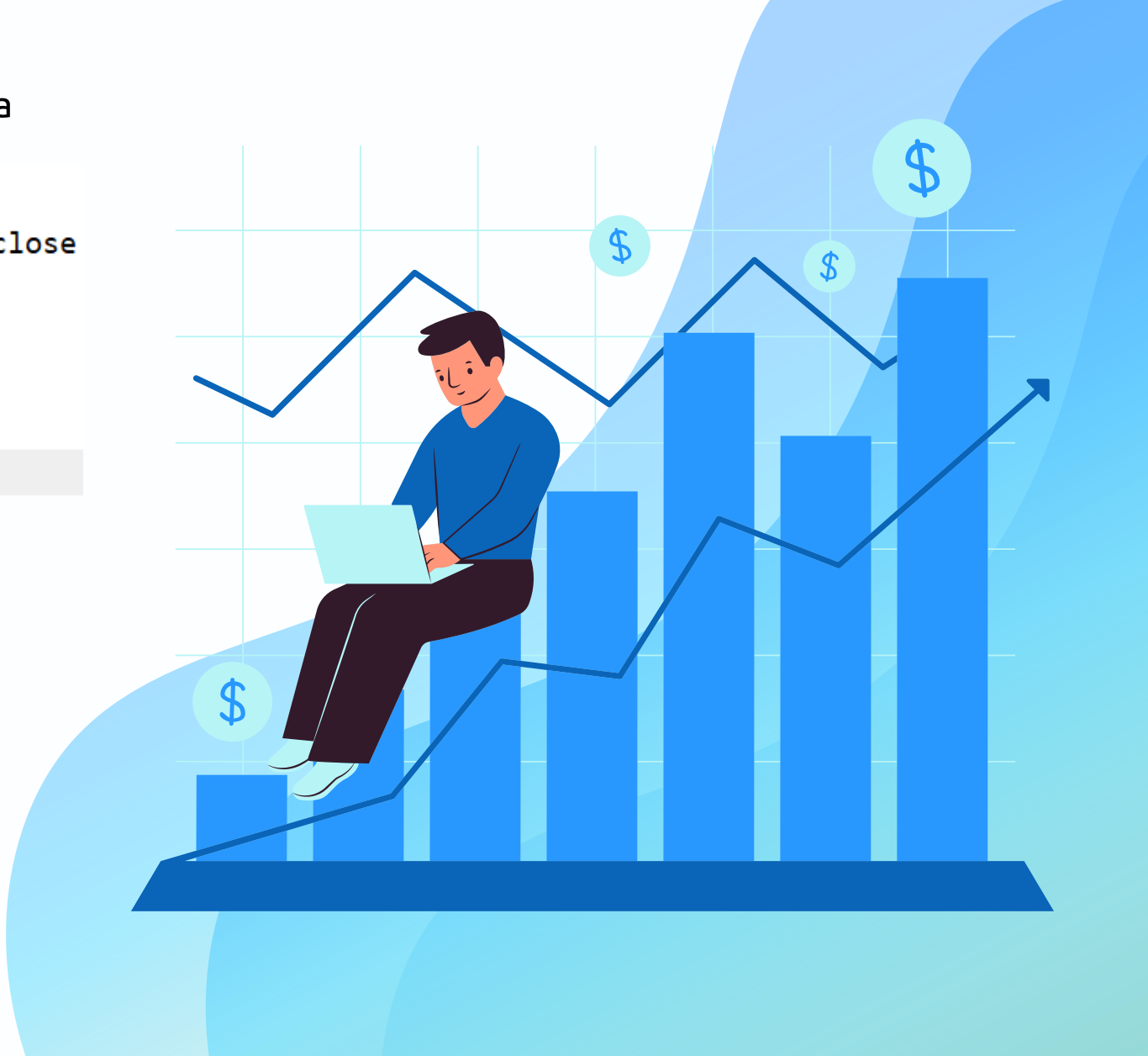
	total_days
▶	6516



2.Trend Analysis Yearly average closing price of the data

```sql
SELECT
    YEAR(`DATE`) AS `Year`, AVG(`Close`) AS avg_close
FROM
    amzn
GROUP BY YEAR(`Date`)
ORDER BY YEAR(`Date`) DESC
LIMIT 10;
```

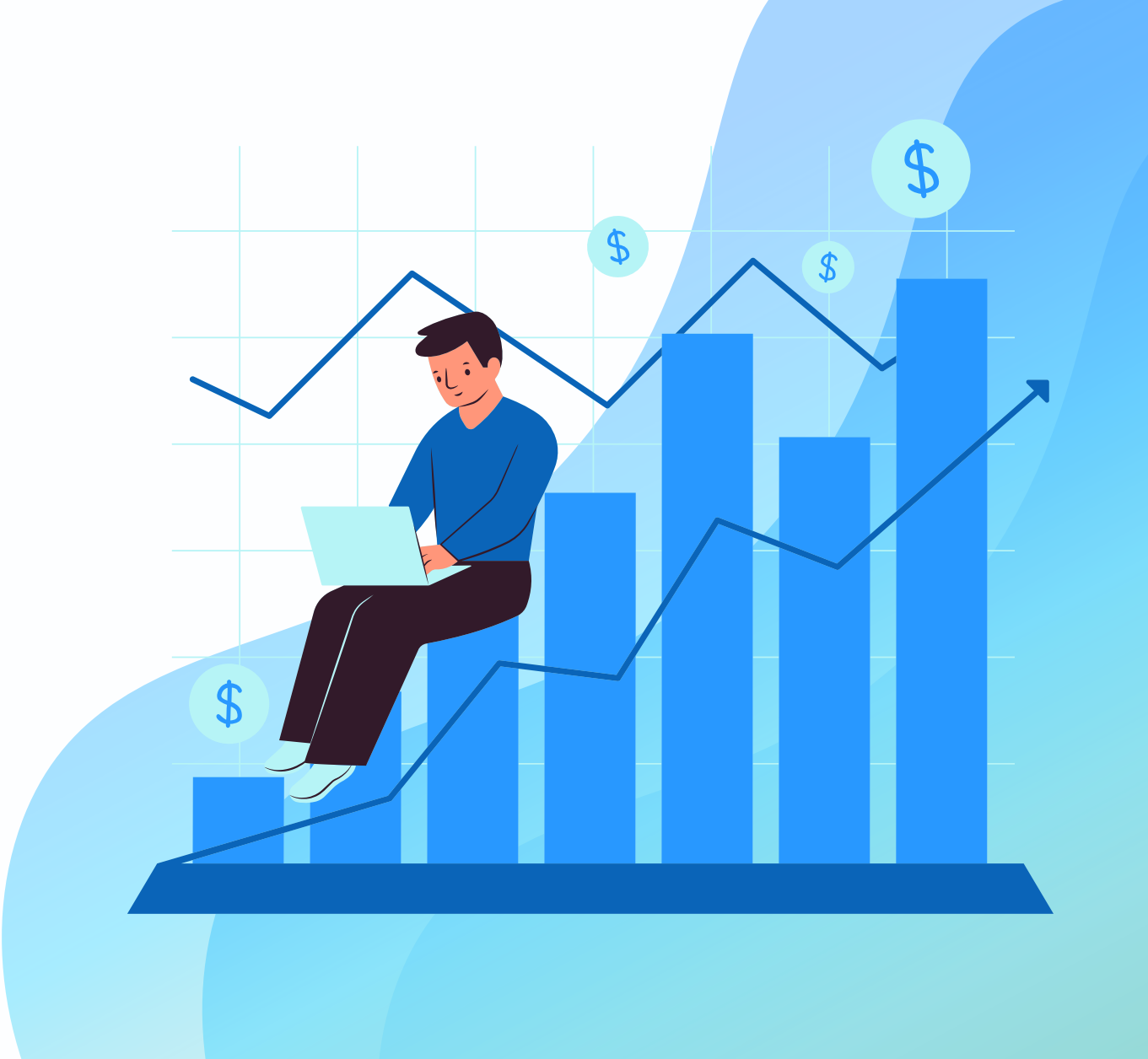
Year	avg_close
2023	96.8858462153846
2022	126.09881882470115
2021	167.19334932142868
2020	134.04275452569178
2019	89.45946026190477
2018	82.08630851792826
2017	48.408350561753004
2016	34.97615664285715
2015	23.906914666666676
2014	16.627551603174613



3.Monthly average closing price of the stocks

```sql
SELECT
    DATE_FORMAT(`Date`, '%Y-%m') AS Month_wise,
    AVG(`Close`) AS avg_close
FROM
    amzn
GROUP BY DATE_FORMAT(`Date`, '%Y-%m')
ORDER BY DATE_FORMAT(`Date`, '%Y-%m') DESC
LIMIT 10;
```

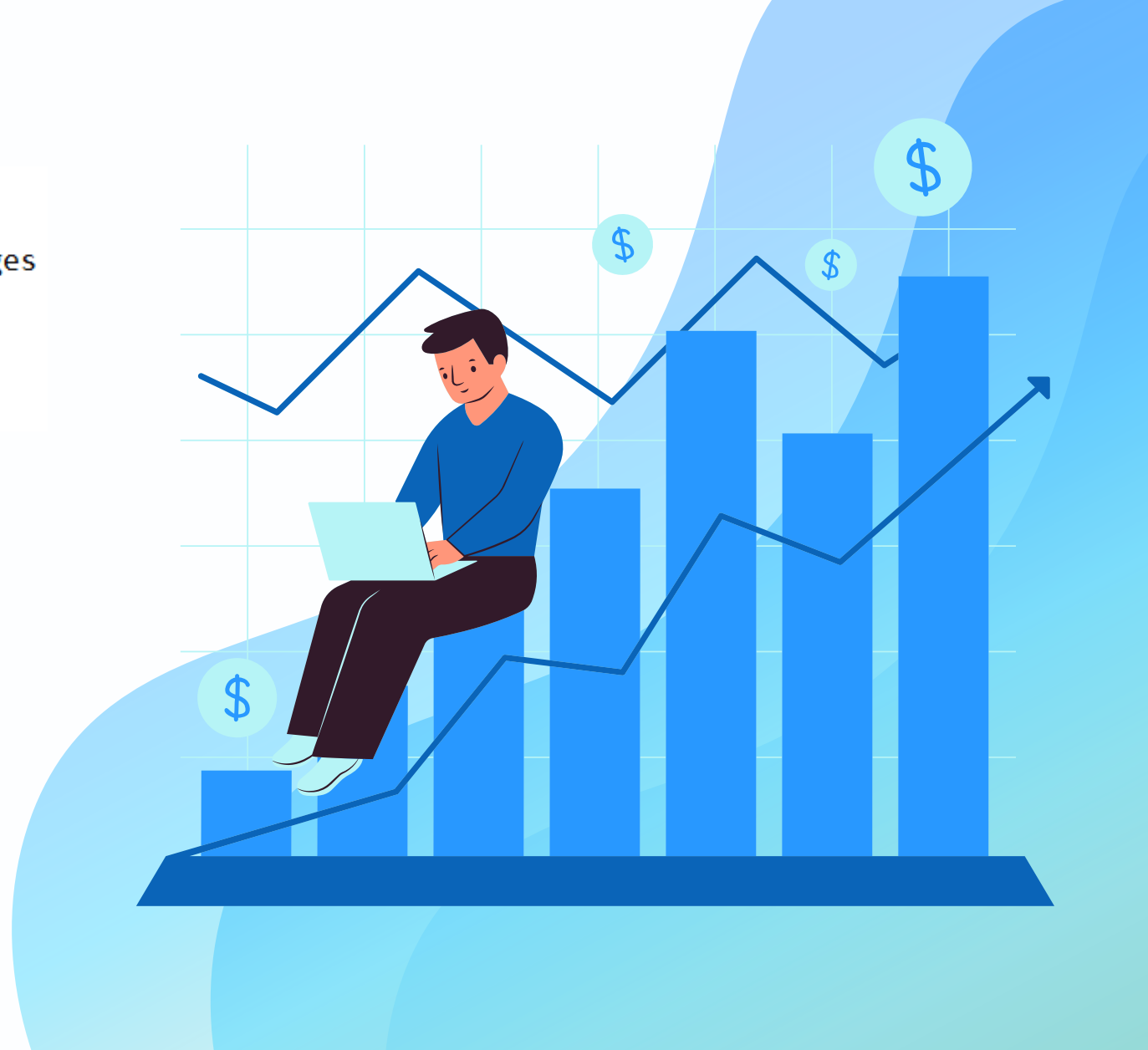
Month_wise	avg_close
2023-04	102.48666633333335
2023-03	96.54695591304346
2023-02	99.21421142105264
2023-01	94.2235001
2022-12	87.9371424761905
2022-11	93.94809533333333
2022-10	114.46047619047617
2022-09	123.2138106190476
2022-08	137.43956556521738
2022-07	117.03749965



4.Calculating the daily price change (Close - Open)

```sql
SELECT
    `Date`, (Close - Open) AS daily_price_changes
FROM
    amzn
ORDER BY `Date` DESC;
```

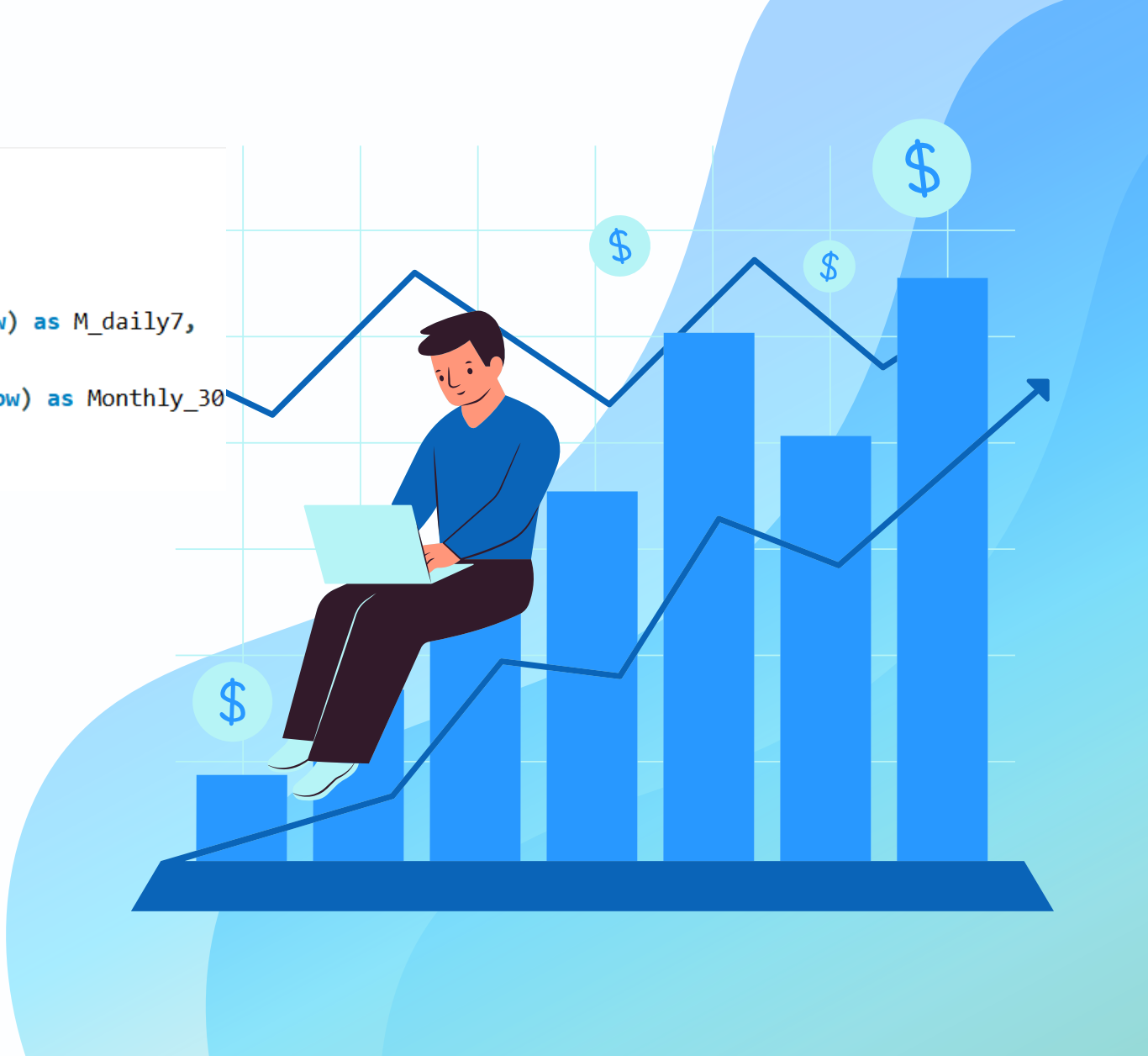
Date	daily_price_changes
2023-04-05	-2.8100060000000013
2023-04-04	1.1999969999999962
2023-04-03	0.1100009999999969
2023-03-31	1.129997000000003
2023-03-30	0.4499969999999962
2023-03-29	1.5599979999999931
2023-03-28	-0.870002999999997
2023-03-27	-1.0299989999999895
2023-03-24	0.05999700000000985
2023-03-23	-1.7200010000000105
2023-03-22	-1.75
2023-03-21	2.470001999999994
2023-03-20	-0.7000050000000044
2023-03-17	-0.8400040000000075
2023-03-16	4.290001000000004
2023-03-15	2.979996



5.Moving Averages Calculate 7-day and 30-day moving averages of closing prices

```sql
select
`Date`,
`Close`,
avg(`Close`)
over (order by `Date` rows between 6 preceding and current row) as M_daily7,
avg(`Close`)
over (order by `Date` rows between 29 preceding and current row) as Monthly_30
from amzn
order by `Date`;
```

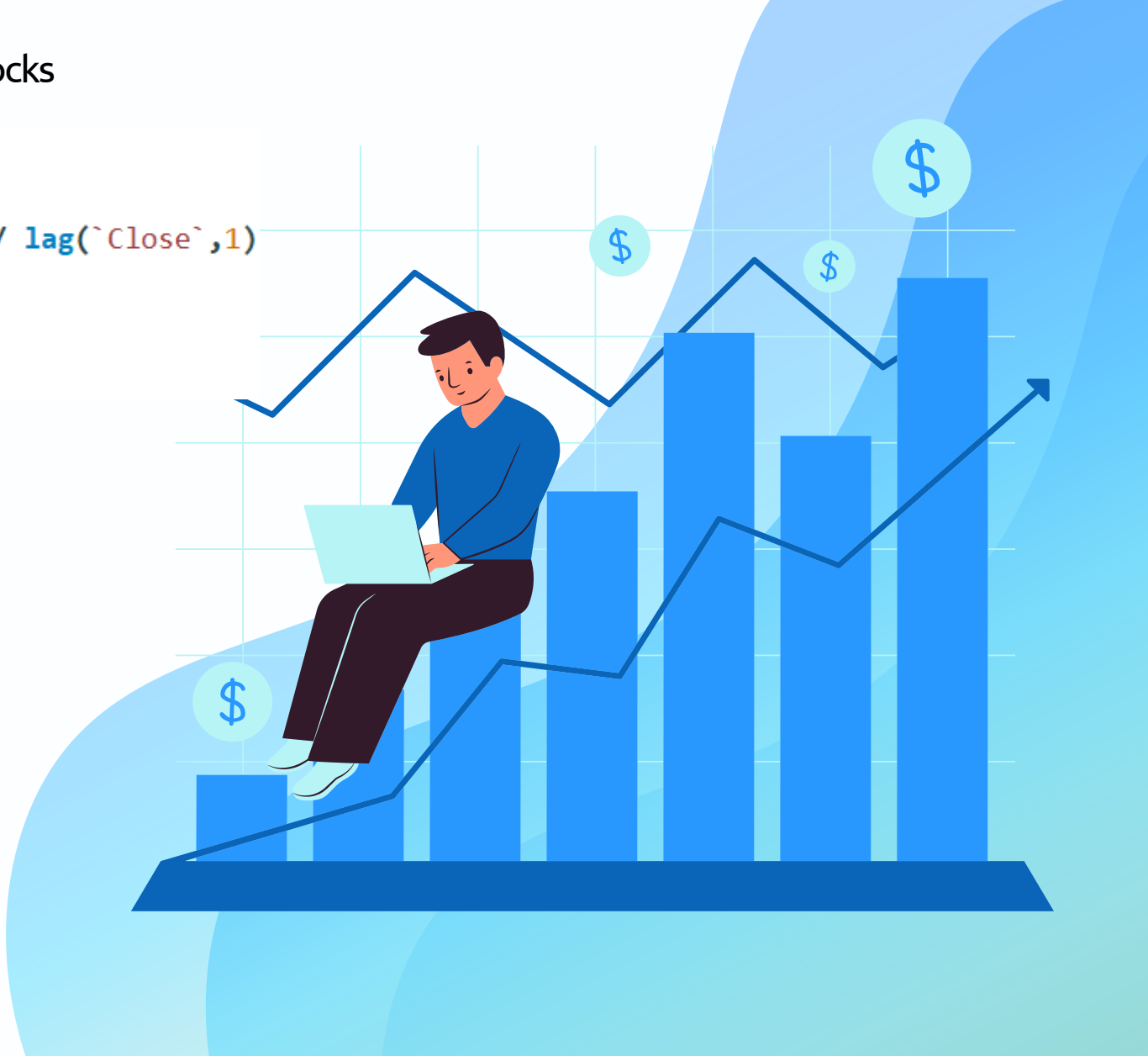
Date	Close	M_daily7	Monthly_30
1997-05-15	0.097917	0.097917	0.097917
1997-05-16	0.086458	0.0921875	0.0921875
1997-05-19	0.085417	0.08993066666666667	0.08993066666666667
1997-05-20	0.081771	0.08789075	0.08789075
1997-05-21	0.071354	0.0845834	0.0845834
1997-05-22	0.069792	0.08211816666666667	0.08211816666666667
1997-05-23	0.075	0.08110128571428572	0.08110128571428572
1997-05-27	0.079167	0.07842271428571428	0.0808595
1997-05-28	0.076563	0.07700914285714286	0.08038211111111111
1997-05-29	0.07526	0.07555814285714287	0.07986990000000001
1997-05-30	0.075	0.07459085714285714	0.07942718181818181
1997-06-02	0.075521	0.07518614285714285	0.07910166666666667
1997-06-03	0.073958	0.0757812857142857	0.078706
1997-06-04	0.070833	0.075186	0.07814364285714284
1997-06-05	0.077083	0.07488828571428571	0.07807293333333333
1997-06-06	0.082813	0.07578114285714285	0.07836918749999999
1997-06-09	0.084375	0.07708328571428572	0.0787224705882353



6.Calculating daily return percentage of the amazon stocks

```sql
select
`Date`,
(( `Close`- lag(`Close`-1) over(order by `Date`)) / lag(`Close`,1)
over (order by `Date`)) * 100 as Daily_per_Return
from amzn
order by `Date`;
```

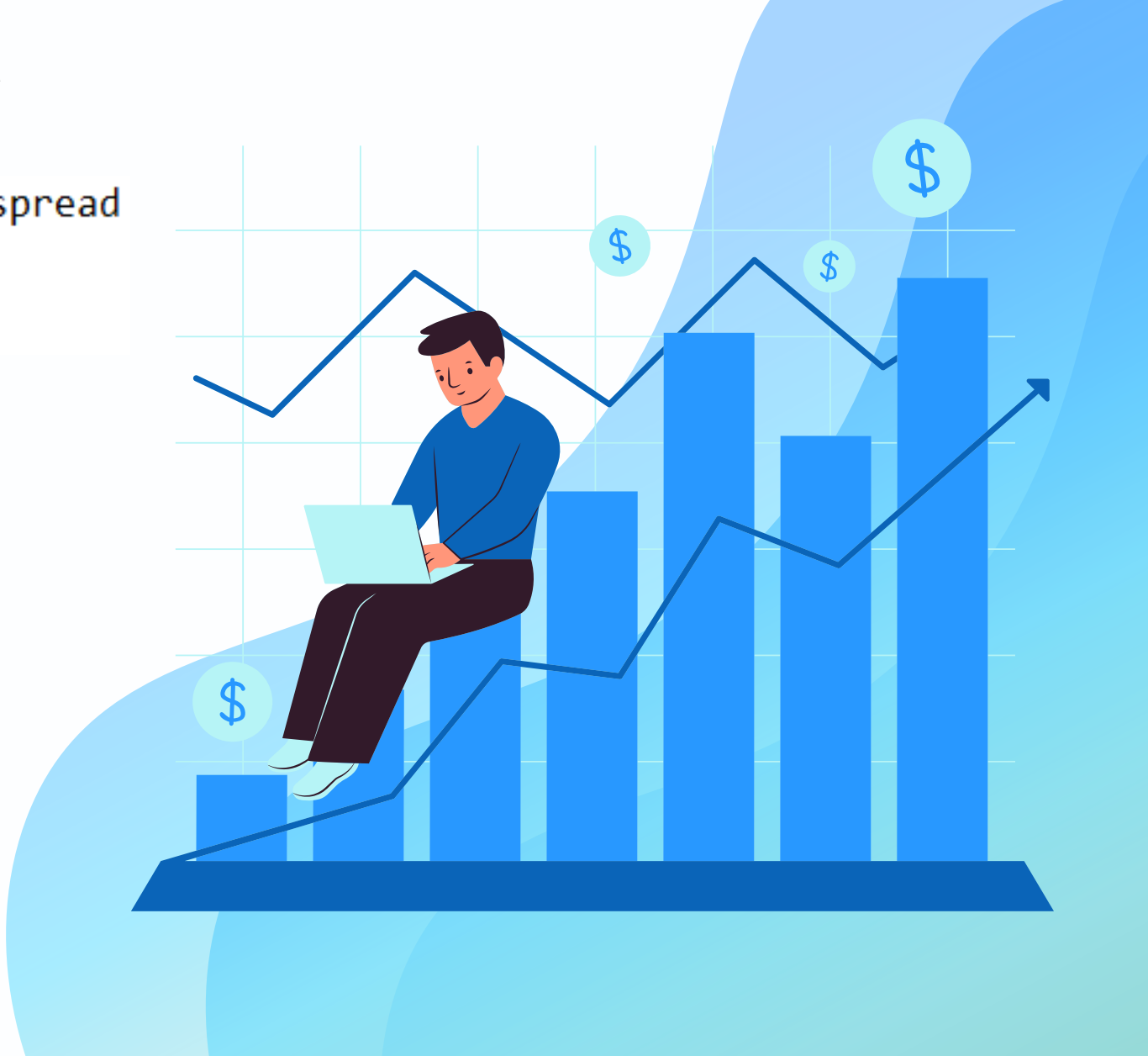
Date	Daily_per_Return
1997-05-15	NULL
1997-05-16	1009.5703503988071
1997-05-19	1155.4269124893012
1997-05-20	1166.458667478371
1997-05-21	1210.1882085335876
1997-05-22	1399.2740420999523
1997-05-23	1440.2911508482348
1997-05-27	1338.8893333333333
1997-05-28	1259.8633268912552
1997-05-29	1304.4120528192468
1997-05-30	1328.3816104172204
1997-06-02	1334.028
1997-06-03	1322.0653857867346
1997-06-04	1347.8933989561644
1997-06-05	1420.5949204466845
1997-06-06	1304.736452914391
1997-06-09	1209.4260562955092



7.Calculate the spread between the highest and lowest prices each day

```sql
select `Date`, (High - Low) as Daily_spread
from amzn
order by `Date`;
```

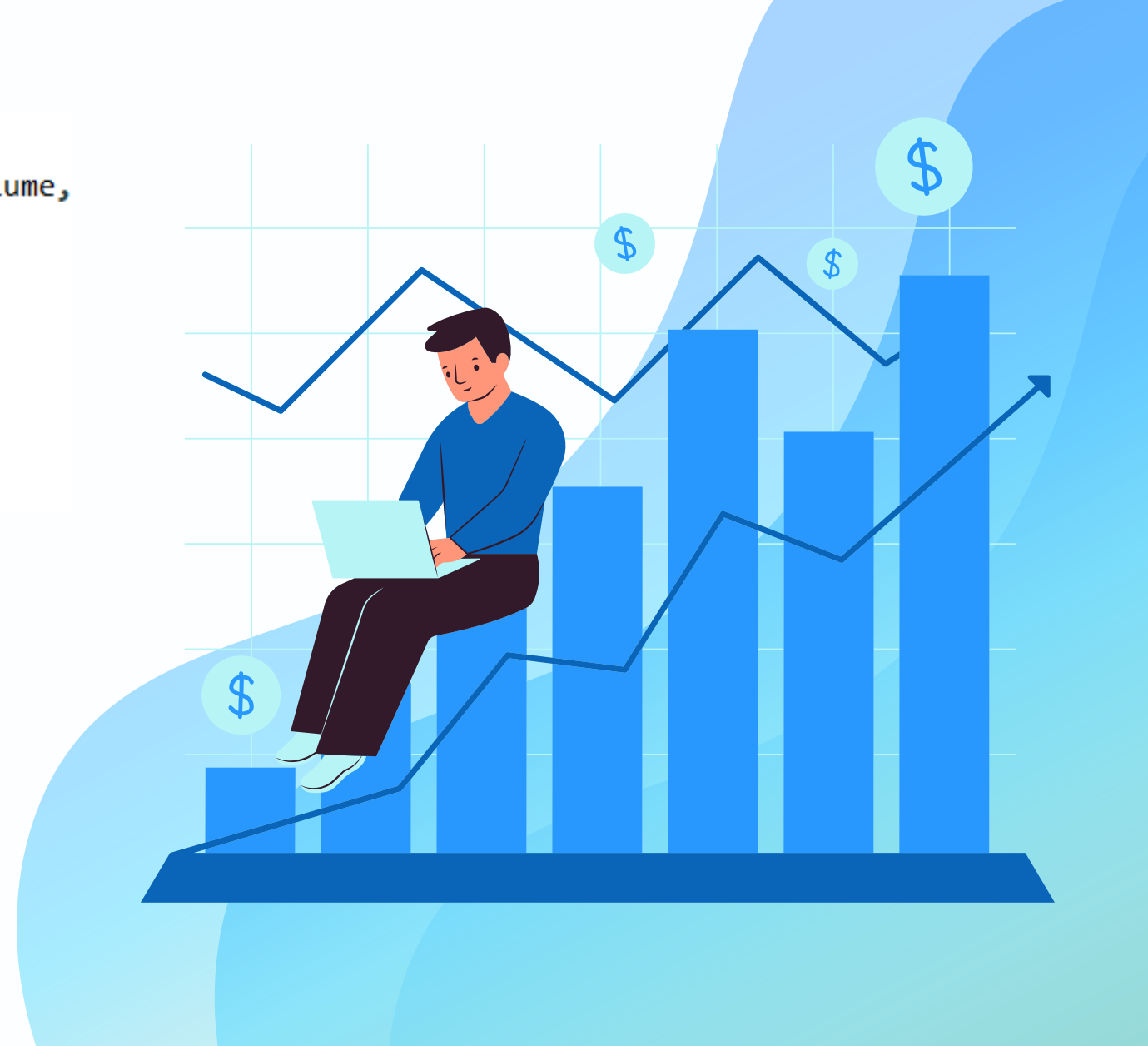
Date	Daily_spread
1997-05-15	0.028646000000000005
1997-05-16	0.013540999999999997
1997-05-19	0.007291999999999993
1997-05-20	0.005728999999999998
1997-05-21	0.013541999999999998
1997-05-22	0.006770999999999999
1997-05-23	0.009374999999999994
1997-05-27	0.009375000000000008
1997-05-28	0.00520799999999999
1997-05-29	0.0031250000000000028
1997-05-30	0.0015630000000000088
1997-06-02	0.0015630000000000088
1997-06-03	0.00260500000000001
1997-06-04	0.004686999999999997
1997-06-05	0.008332999999999993
1997-06-06	0.009896000000000002
1997-06-09	0.002604000000000009



8.Monthly total and average volume of the stocks

```sql
SELECT
    DATE_FORMAT(`Date`, '%Y-%m') AS Monthly_volume,
    SUM(Volume) AS Total_Volume,
    AVG(Volume) AS Avg_volume
FROM
    amzn
GROUP BY DATE_FORMAT(`Date`, '%Y-%m')
ORDER BY DATE_FORMAT(`Date`, '%Y-%m') DESC;
```

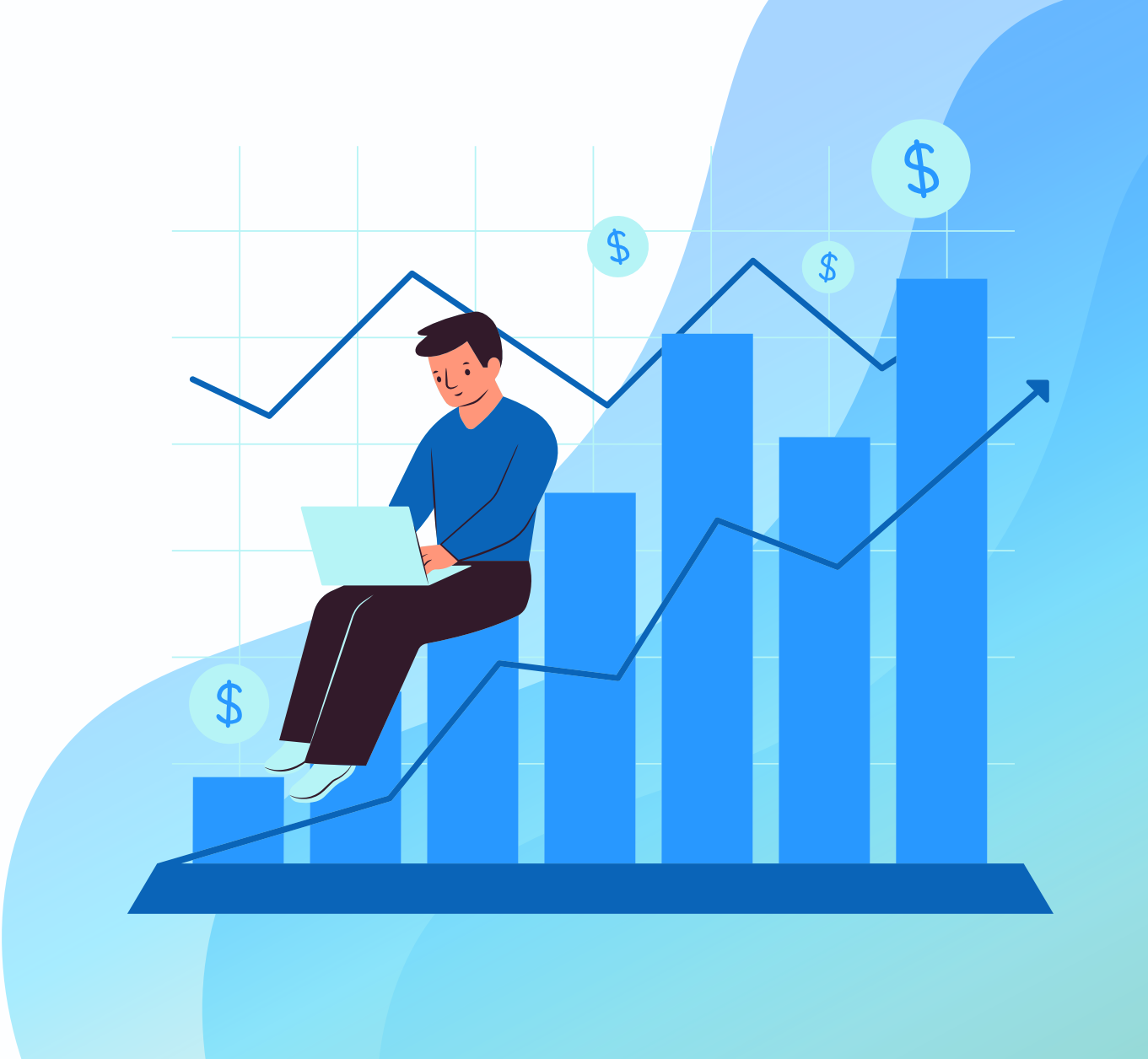
Monthly_volume	Total_Volume	Avg_volume
2023-04	134901200	44967066.6667
2023-03	1348923000	58648826.0870
2023-02	1364102000	71794842.1053
2023-01	1523798600	76189930.0000
2022-12	1549193300	73771109.5238
2022-11	2035133200	96911104.7619
2022-10	1459311500	69491023.8095
2022-09	1210487600	57642266.6667
2022-08	1170449000	50889086.9565
2022-07	1337852600	66892630.0000
2022-06	1767601100	84171480.9524
2022-05	2258476000	107546476.1905
2022-04	1465008000	73250400.0000
2022-03	1628486000	70803739.1304
2022-02	1689604000	88926526.3158
2022-01	1524654000	76232700.0000
2021-12	1287634000	58528818.1818



9.Yearly total and average volume

```sql
SELECT
    YEAR(`Date`) AS Yearly_volume,
    SUM(Volume) AS Total_Volume,
    AVG(Volume) AS Avg_volume
FROM
    amzn
GROUP BY YEAR(`Date`)
ORDER BY year(`Date`) DESC;
```

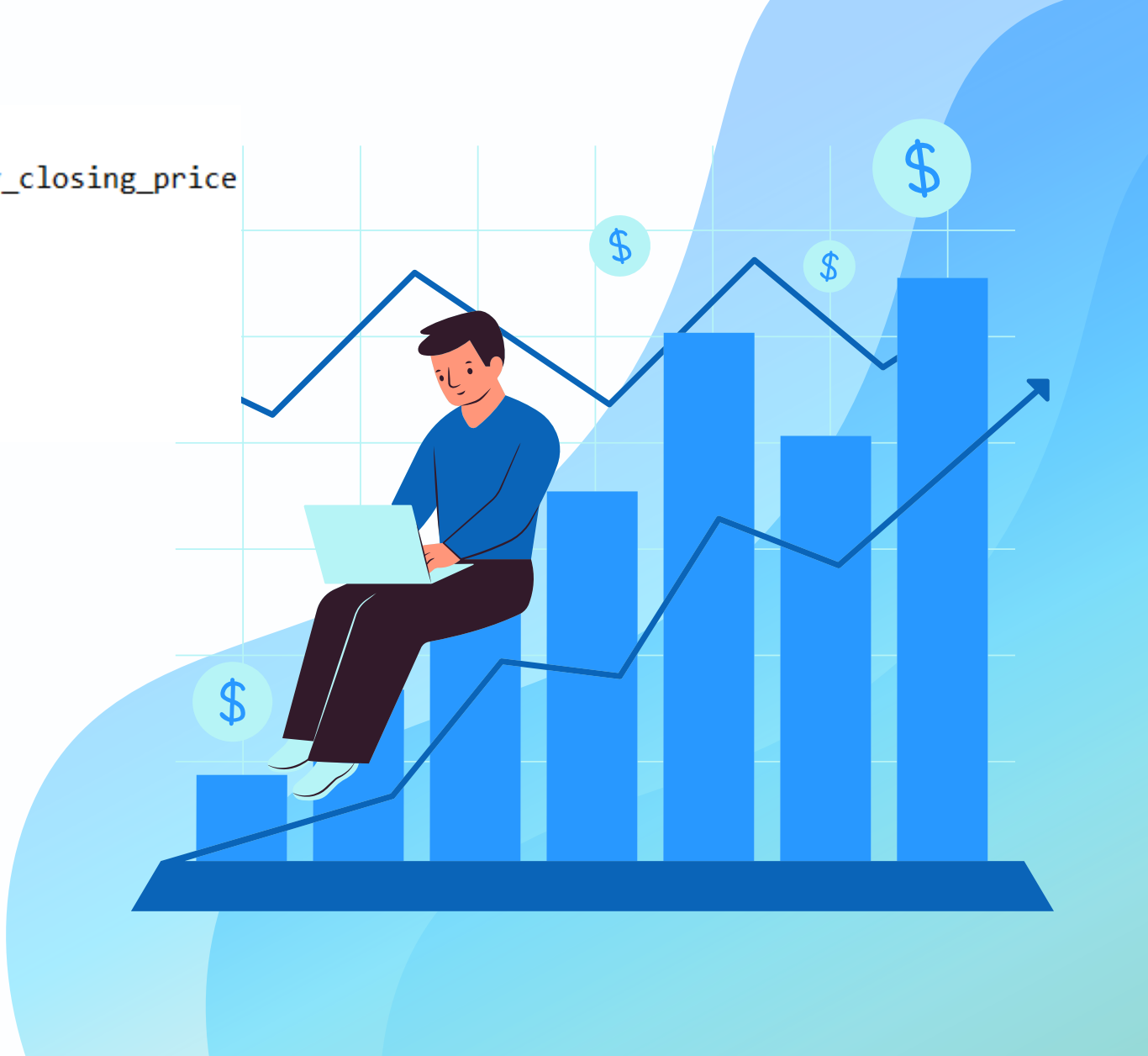
Yearly_volume	Total_Volume	Avg_volume
2023	4371724800	67257304.6154
2022	19096256300	76080702.3904
2021	17076362000	67763341.2698
2020	24950814000	98619818.1818
2019	19493002000	77353182.5397
2018	28357952000	112979888.4462
2017	17654108000	70335091.6335
2016	20775126000	82440976.1905
2015	19142040000	75960476.1905
2014	20581334000	81671960.3175
2013	14958114000	59357595.2381
2012	20998082000	83992328.0000
2011	29199572000	115871317.4603



10.Identifying the highest top 5 closing price by year

```sql
SELECT
    YEAR(`Date`) AS `year`, MAX(`Close`) AS Yearly_closing_price
FROM
    amzn
GROUP BY YEAR(`Date`)
ORDER BY YEAR(`Date`) DESC
LIMIT 5;
```

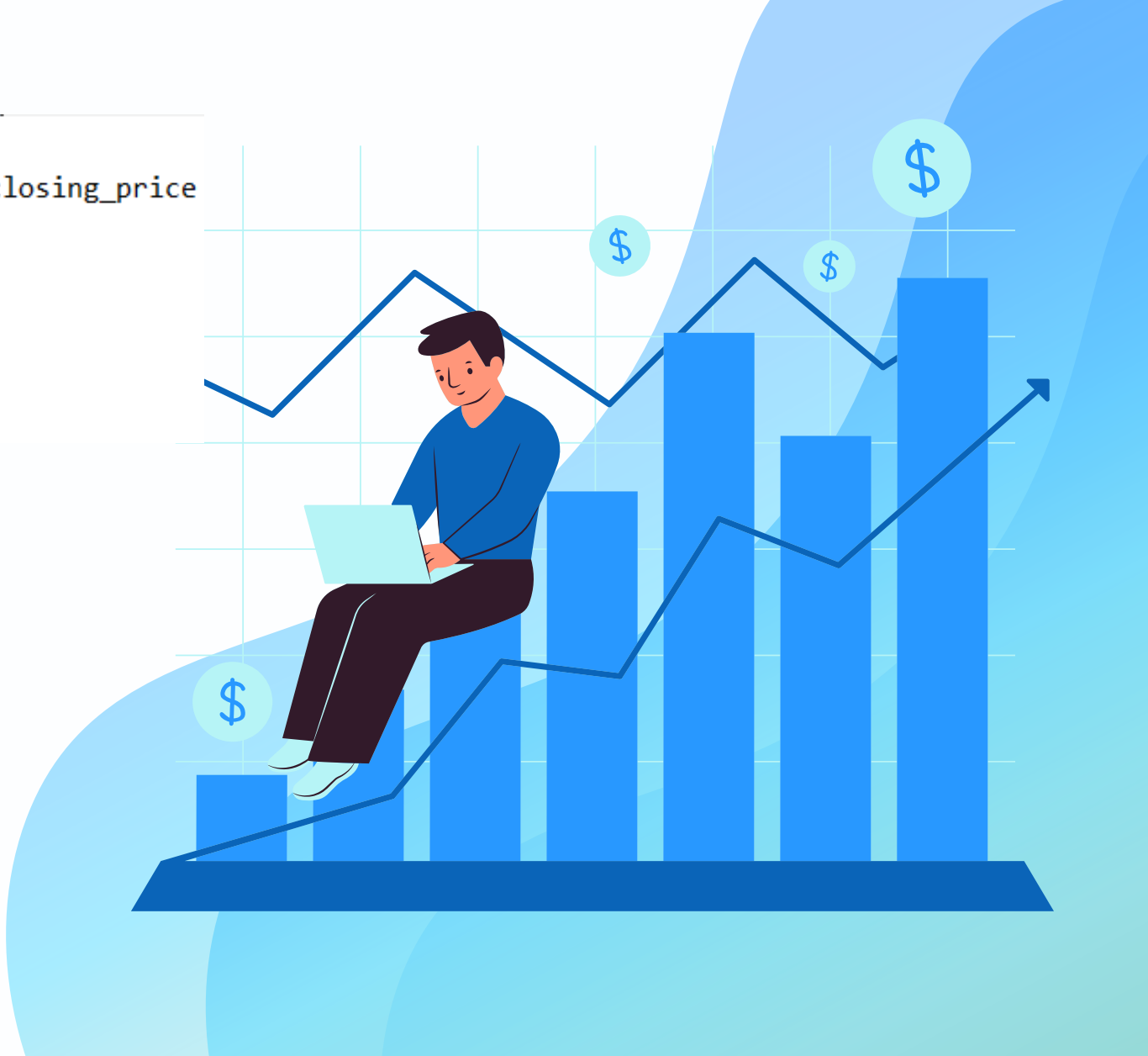
year	Yearly_closing_price
2023	112.910004
2022	170.404495
2021	186.570496
2020	176.572495
2019	101.0495



11.Identifying the lowest top 5 closing price each year

```sql
SELECT
    YEAR(`Date`) AS `year`, Min(`Close`) AS Yearly_closing_price
FROM
    amzn
GROUP BY YEAR(`Date`)
ORDER BY YEAR(`Date`) Asc
LIMIT 5;
```

year	Yearly_closing_price
1997	0.069792
1998	0.213021
1999	2.1375
2000	0.759375
2001	0.2985



- Identify dates with the highest single-day returns

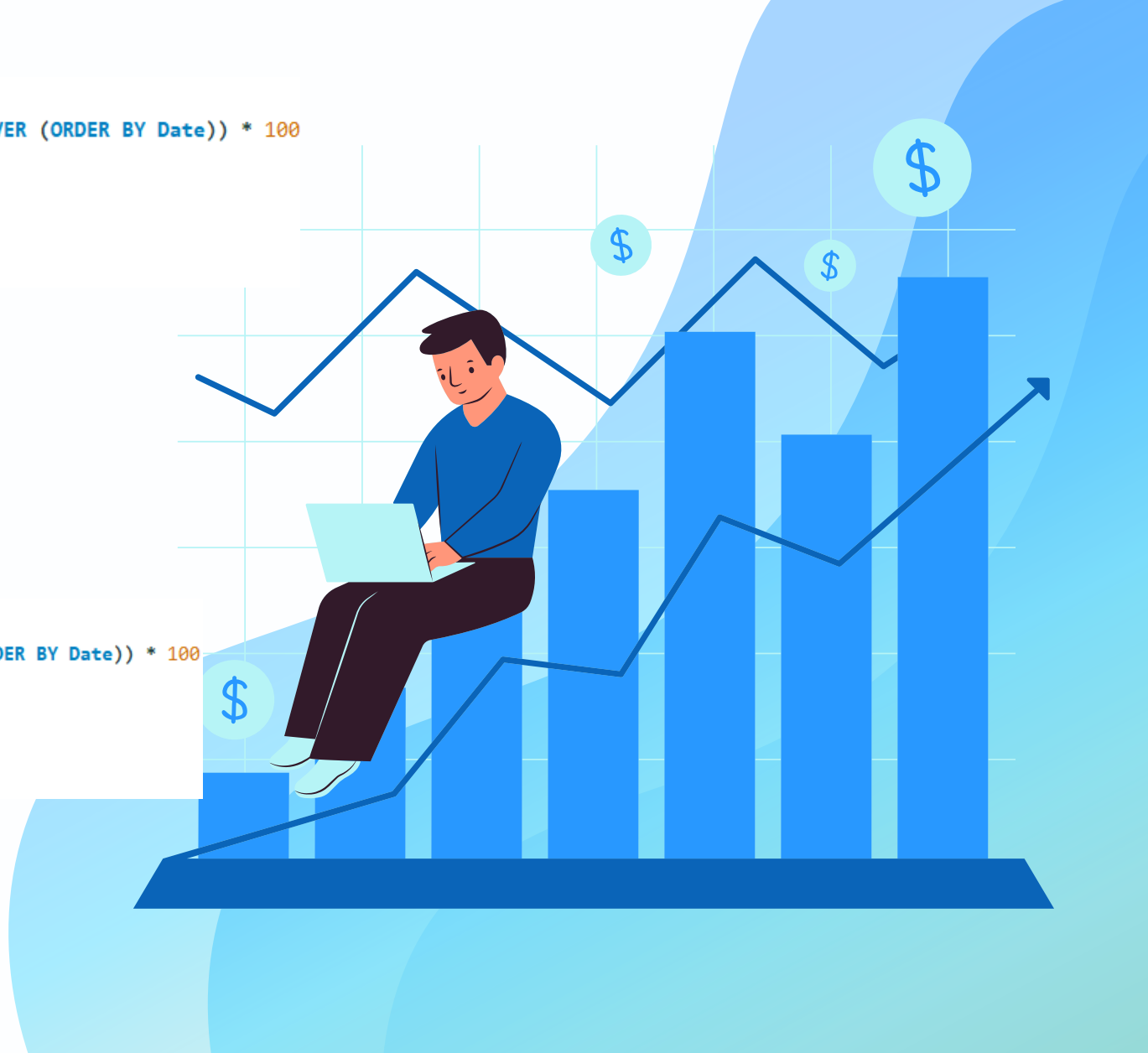
```sql
SELECT Date,
       ((Close - LAG(Close, 1) OVER (ORDER BY Date)) / LAG(Close, 1) OVER (ORDER BY Date)) * 100
       AS Daily_percentage_Return
FROM amzn
ORDER BY Daily_percentage_Return DESC
LIMIT 1; -- For highest return
```

Date	Daily_percentage_Return
2001-11-26	34.47136563876653

- Identify dates with the lowest single-day returns

```sql
SELECT Date,
       ((Close - LAG(Close, 1) OVER (ORDER BY Date)) / LAG(Close, 1) OVER (ORDER BY Date)) * 100
       AS Daily_percentage_Return
FROM amzn
ORDER BY Daily_percentage_Return ASC
LIMIT 2 ; -- For lowest return
```

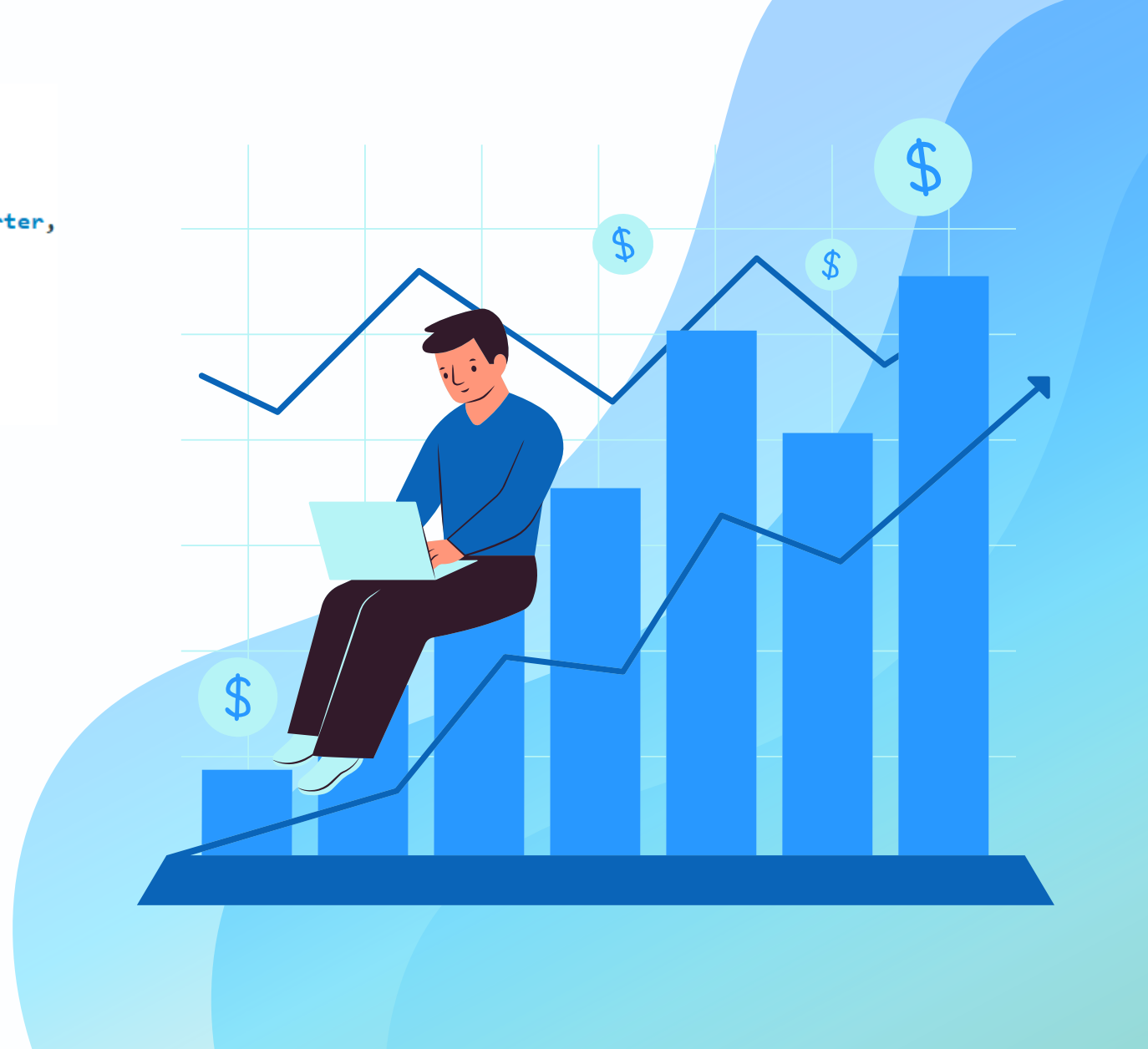
Date	Daily_percentage_Return
1997-05-15	NULL
2001-07-24	-24.766063630692454



12.Quarterly average closing price of the stocks

```sql
SELECT Quarter,
       AVG(`Close`) AS Avg_Close
FROM (
    SELECT CONCAT(YEAR(`Date`), '-Q', QUARTER(`Date`)) AS Quarter,
           `Close`
    FROM amzn
) AS sub
GROUP BY Quarter
ORDER BY Quarter;
```

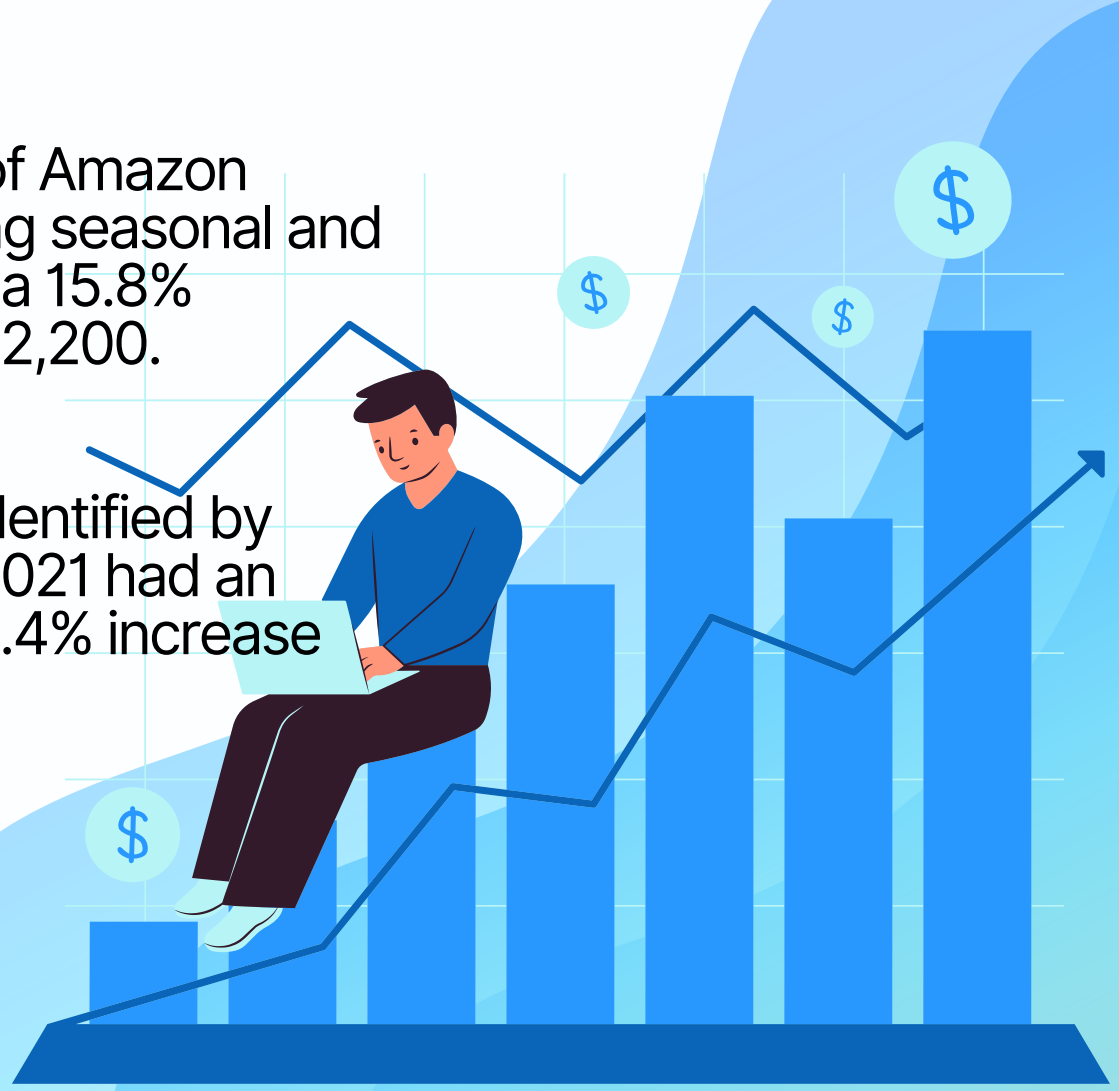
Quarter	Avg_Close
1997-Q2	0.07775881250000002
1997-Q3	0.13181559375000004
1997-Q4	0.22010503124999997
1998-Q1	0.2826813114754098
1998-Q2	0.4474641587301587
1998-Q3	0.9137574374999998
1998-Q4	1.459621171875
1999-Q1	3.1637680819672123
1999-Q2	3.552232349206349
1999-Q3	2.962256
1999-Q4	4.122192390625
2000-Q1	3.4478670634920636
2000-Q2	2.5815104126984125



Insights and Conclusion

Quarterly Trends: The average closing price of Amazon stock varies across different quarters, reflecting seasonal and market influences. For example, Q2 2020 saw a 15.8% increase from Q1 2020, rising from $1,900 to $2,200.

Highs and Lows: Significant quarters can be identified by their average closing prices. For instance, Q1 2021 had an average closing price of $3,000, showing a 36.4% increase from Q2 2020.



Thank you