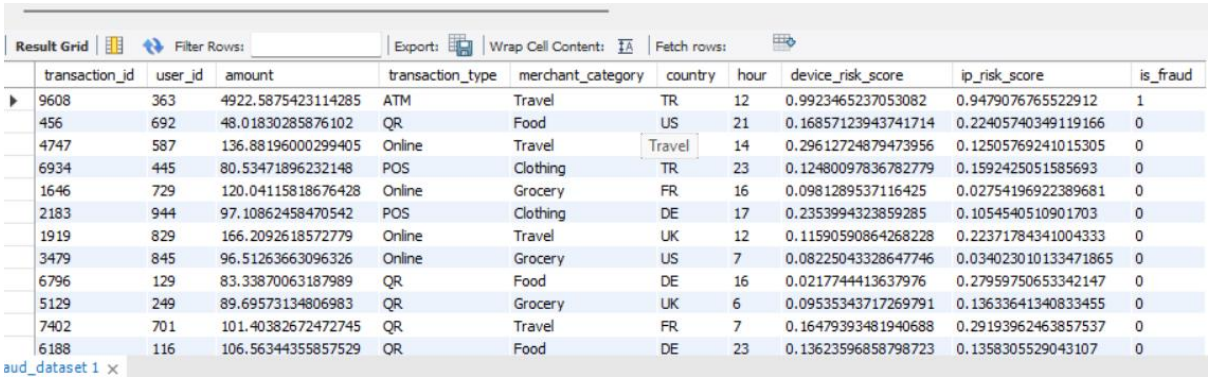
MySQL Queries

create database Financial_Fraud

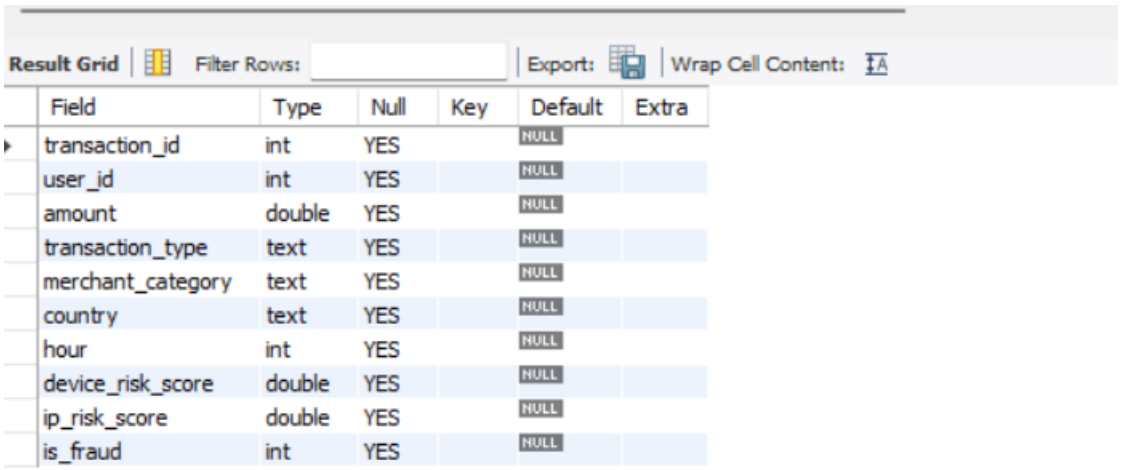
use Financial_Fraud

select * from Financial_Fraud_dataset



transaction_id	user_id	amount	transaction_type	merchant_category	country	hour	device_risk_score	ip_risk_score	is_fraud
9608	363	4922.5875423114285	ATM	Travel	TR	12	0.9923465237053082	0.9479076765522912	1
456	692	48.01830285876102	QR	Food	US	21	0.16857123943741714	0.22405740349119166	0
4747	587	136.88196000299405	Online	Travel	Travel	14	0.29612724879473956	0.12505769241015305	0
6934	445	80.53471896232148	POS	Clothing	TR	23	0.12480097836782779	0.1592425051585693	0
1646	729	120.04115818676428	Online	Grocery	FR	16	0.0981289537116425	0.02754196922389681	0
2183	944	97.10862458470542	POS	Clothing	DE	17	0.2353994323859285	0.1054540510901703	0
1919	829	166.2092618572779	Online	Travel	UK	12	0.11590590864268228	0.22371784341004333	0
3479	845	96.51263663096326	Online	Grocery	US	7	0.08225043328647746	0.034023010133471865	0
6796	129	83.33870063187989	QR	Food	DE	16	0.021774441363797 6	0.27959750653342147	0
5129	249	89.69573134806983	QR	Grocery	UK	6	0.09535343717269791	0.13633641340833455	0
7402	701	101.40382672472745	QR	Travel	FR	7	0.16479393481940688	0.29193962463857537	0
6188	116	106.56344355857529	QR	Food	DE	23	0.13623596858798723	0.1358305529043107	0

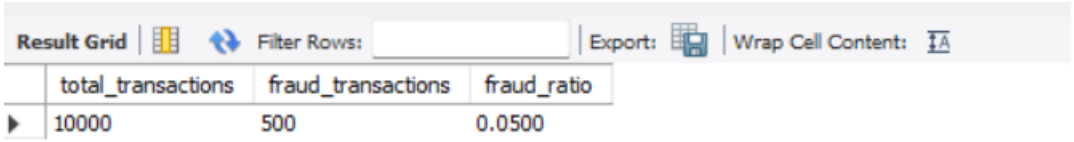
DESCRIBE Financial_Fraud_dataset;



Field	Type	Null	Key	Default	Extra
transaction_id	int	YES		NULL	
user_id	int	YES		NULL	
amount	double	YES		NULL	
transaction_type	text	YES		NULL	
merchant_category	text	YES		NULL	
country	text	YES		NULL	
hour	int	YES		NULL	
device_risk_score	double	YES		NULL	
ip_risk_score	double	YES		NULL	
is_fraud	int	YES		NULL	

Overall Fraud Ratio

```
SELECT
    COUNT(*) AS total_transactions,
    SUM(is_fraud) AS fraud_transactions,
    ROUND(SUM(is_fraud) / COUNT(*), 4) AS fraud_ratio
FROM financial_fraud_dataset;
```



total_transactions	fraud_transactions	fraud_ratio
10000	500	0.0500

Fraud vs Legit Comparison

```
SELECT
    is_fraud,
    COUNT(*) AS txn_count,
    ROUND(AVG(device_risk_score), 3) AS avg_device_risk,
    ROUND(AVG(ip_risk_score), 3) AS avg_ip_risk,
    ROUND(AVG(amount), 2) AS avg_amount
FROM financial_fraud_dataset
GROUP BY is_fraud;
```

is_fraud	txn_count	avg_device_risk	avg_ip_risk	avg_amount
1	500	0.858	0.852	1657.58
0	9500	0.148	0.15	100.28

Feature Influence

```
SELECT
    'device_risk_score' AS feature,
    ROUND(
```

```
        AVG(CASE WHEN is_fraud = 1 THEN device_risk_score END) -
        AVG(CASE WHEN is_fraud = 0 THEN device_risk_score END),
        3
    ) AS separation_score
FROM financial_fraud_dataset

UNION ALL

SELECT
    'ip_risk_score',
    ROUND(
        AVG(CASE WHEN is_fraud = 1 THEN ip_risk_score END) -
        AVG(CASE WHEN is_fraud = 0 THEN ip_risk_score END),
        3
    )
FROM financial_fraud_dataset

UNION ALL

SELECT
    'amount',
    ROUND(
        AVG(CASE WHEN is_fraud = 1 THEN amount END) -
        AVG(CASE WHEN is_fraud = 0 THEN amount END),
        2
    )
FROM financial_fraud_dataset
```

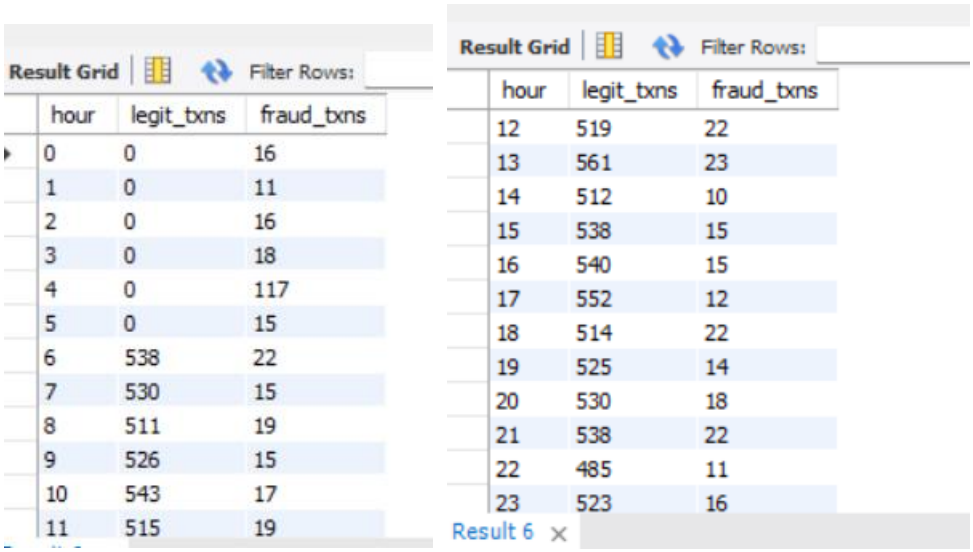
```
UNION ALL

SELECT
  'hour',
  ROUND(
    AVG(CASE WHEN is_fraud = 1 THEN hour END) -
    AVG(CASE WHEN is_fraud = 0 THEN hour END),
    2
  )
FROM financial_fraud_dataset;
```

Result Grid | Filter Rows:

feature	separation_score
device_risk_score	0.71
ip_risk_score	0.702
amount	1557.3
hour	-4.45

Hourly Fraud Pattern

```
SELECT
  hour,
  SUM(is_fraud = 0) AS legit_txns,
  SUM(is_fraud = 1) AS fraud_txns
FROM financial_fraud_dataset
GROUP BY hour
ORDER BY hour;
```



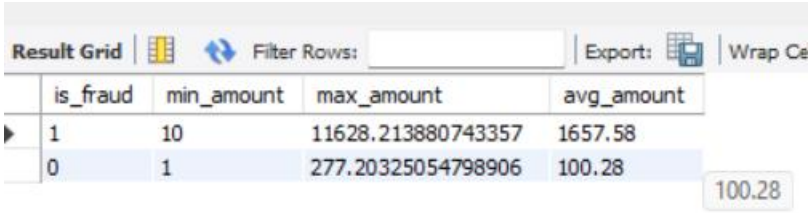
Result Grid | Filter Rows:

hour	legit_txns	fraud_txns
0	0	16
1	0	11
2	0	16
3	0	18
4	0	117
5	0	15
6	538	22
7	530	15
8	511	19
9	526	15
10	543	17
11	515	19
12	519	22
13	561	23
14	512	10
15	538	15
16	540	15
17	552	12
18	514	22
19	525	14
20	530	18
21	538	22
22	485	11
23	523	16

Result 6

Transaction Amount Distribution

```
SELECT
    is_fraud,
    MIN(amount) AS min_amount,
    MAX(amount) AS max_amount,
    ROUND(AVG(amount), 2) AS avg_amount
FROM financial_fraud_dataset
GROUP BY is_fraud;
```



Result Grid | Filter Rows: | Export: | Wrap Ce

is_fraud	min_amount	max_amount	avg_amount
1	10	11628.213880743357	1657.58
0	1	277.20325054798906	100.28

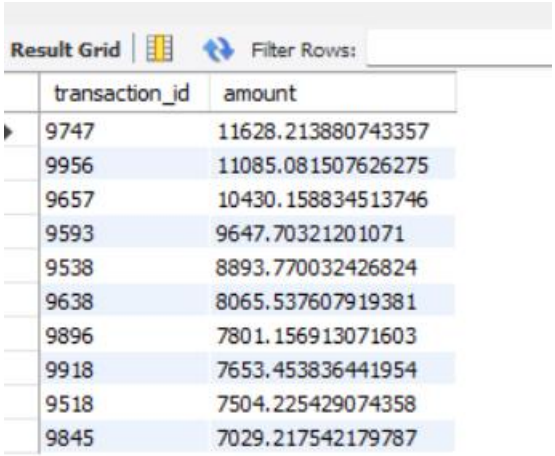
100.28

High-Value Fraud Outliers

```
SELECT
    transaction_id,
    amount
FROM financial_fraud_dataset
WHERE is_fraud = 1
```

ORDER BY amount DESC

LIMIT 10;



Result Grid | Filter Rows:

transaction_id	amount
9747	11628.213880743357
9956	11085.081507626275
9657	10430.158834513746
9593	9647.70321201071
9538	8893.770032426824
9638	8065.537607919381
9896	7801.156913071603
9918	7653.453836441954
9518	7504.225429074358
9845	7029.217542179787

Device Risk Score

SELECT

FLOOR(device_risk_score * 10) / 10 AS risk_bucket,

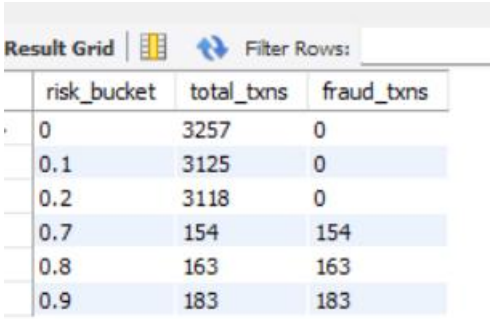
COUNT(*) AS total_txns,

SUM(is_fraud) AS fraud_txns

FROM financial_fraud_dataset

GROUP BY risk_bucket

ORDER BY risk_bucket;



Result Grid | Filter Rows:

risk_bucket	total_txns	fraud_txns
0	3257	0
0.1	3125	0
0.2	3118	0
0.7	154	154
0.8	163	163
0.9	183	183

IP Risk Score

SELECT

FLOOR(ip_risk_score * 10) / 10 AS risk_bucket,

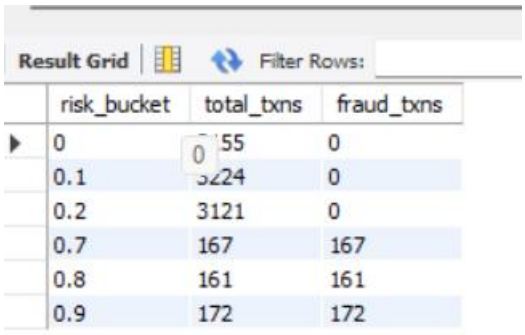
COUNT(*) AS total_txns,

SUM(is_fraud) AS fraud_txns

FROM financial_fraud_dataset

GROUP BY risk_bucket

ORDER BY risk_bucket;



Result Grid | Filter Rows:

risk_bucket	total_txns	fraud_txns
0	55	0
0.1	5224	0
0.2	3121	0
0.7	167	167
0.8	161	161
0.9	172	172

Geographic Fraud Distribution

SELECT

country,

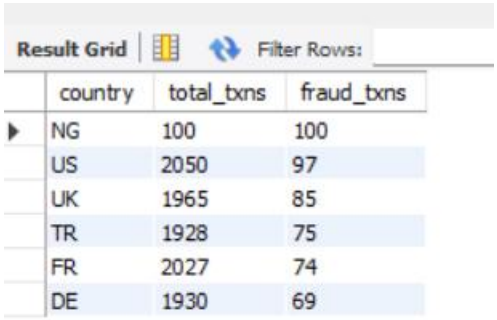
COUNT(*) AS total_txns,

SUM(is_fraud) AS fraud_txns

FROM financial_fraud_dataset

GROUP BY country

ORDER BY fraud_txns DESC;



Result Grid | Filter Rows:

country	total_txns	fraud_txns
NG	100	100
US	2050	97
UK	1965	85
TR	1928	75
FR	2027	74
DE	1930	69

Time Range

ALTER TABLE financial_fraud_dataset

ADD COLUMN time_range VARCHAR(15);

```
UPDATE financial_fraud_dataset
SET time_range =
   CASE
      WHEN hour BETWEEN 5 AND 11 THEN 'Morning'
      WHEN hour BETWEEN 12 AND 17 THEN 'Afternoon'
      WHEN hour BETWEEN 18 AND 22 THEN 'Evening'
      ELSE 'Night'
   END;
```

Rule-Based Fraud Flag

```
SELECT
   transaction_id,
   amount,
   device_risk_score,
   ip_risk_score,
   CASE
      WHEN device_risk_score > 0.8
       AND ip_risk_score > 0.8
       AND amount > 1000
      THEN 'High Fraud Risk'
      ELSE 'Low Risk'
   END AS risk_flag
FROM financial_fraud_dataset
LIMIT 6;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

transaction_id	amount	device_risk_score	ip_risk_score	risk_flag
9608	4922.5875423114285	0.9923465237053082	0.9479076765522912	High Fraud Risk
456	48.01830285876102	0.16857123943741714	0.22405740349119166	Low Risk
4747	136.88196000299405	0.296127248794739 56	0.12505769241015305	Low Risk
6934	80.53471896232148	0.12480097836782779	0.1592425051585693	Low Risk
1646	120.04115818676428	0.0981289537116425	0.02754196922389681	Low Risk
2183	97.10862458470542	0.2353994323859285	0.1054540510901703	Low Risk