

Checking Data

The screenshot shows the Azure Data Studio interface with a SQL query executed in the Bank Loan DB. The query includes a 'Checking data' comment, a 'Total loan applications' query, an 'MTD query (month-to-date)', and an 'MOM query'. The results pane shows a table with columns: id, address_state, application_type, emp_length, emp_title, grade, home_ownership, issue_date, and last_crr. The results pane also shows a table with columns: id, address_state, application_type, emp_length, emp_title, grade, home_ownership, issue_date, and last_crr.

id	address_state	application_type	emp_length	emp_title	grade	home_ownership	issue_date	last_crr
1	1077430	GA	INDIVIDUAL	< 1 year	Ryder	C	RENT	2021-02-11
2	1072053	CA	INDIVIDUAL	9 years	MKC Accounting	E	RENT	2021-01-01
3	1069243	CA	INDIVIDUAL	4 years	Chemat Technology Inc	C	RENT	2021-01-05
4	1041756	TX	INDIVIDUAL	< 1 year	barnes distribution	B	MORTGAGE	2021-02-25
5	1068350	IL	INDIVIDUAL	10+ years	J&J Steel Inc	A	MORTGAGE	2021-01-01
6	1062608	CA	INDIVIDUAL	3 years	Studio 94 Corp	C	RENT	2021-07-17
7	1067441	TX	INDIVIDUAL	10+ years	American Airlines	C	MORTGAGE	2021-11-19

Total_Loan_Applications

The screenshot shows the SQL query editor with the 'Total_Loan_Applications' query. The query includes a 'Checking data' comment, a 'Total loan applications' query, an 'MTD query (month-to-date)', and an 'MOM query'. The results pane shows a table with columns: id, address_state, application_type, emp_length, emp_title, grade, home_ownership, issue_date, and last_crr. The results pane also shows a table with columns: id, address_state, application_type, emp_length, emp_title, grade, home_ownership, issue_date, and last_crr.

Total_Loan_Applications
38576

-- MTD query (month-to-date)

```
SELECT count(id) as MTD_Total_Loan_Applications from bank_loan_data
Where year(issue_date)=(Select year(max(issue_date)) from bank_loan_data) and
      month(issue_date)=(Select month(max(issue_date)) from bank_loan_data)
```

Results		Messages
	MTD_Total_Loan_Applications	▼
1	4314	Results grid

-- MOM query using self join method

```

With monthly_data as (
SELECT
year(issue_date) as yr,
MONTH(issue_date) as mn,
count(id) as Total_Loan_Applications,
row_number() over(order by year(issue_date) desc, month(issue_date) desc) as rn
From
bank_loan_data
Group by year(issue_date),MONTH(issue_date)
)

SELECT
curr.yr as current_year,
curr.mn as current_month,
curr.Total_Loan_Applications as current_total,
prev.Total_Loan_Applications as previous_total,
ROUND(
  CAST((curr.Total_Loan_Applications - prev.Total_Loan_Applications) * 100.0
    / NULLIF(prev.Total_Loan_Applications, 0) AS DECIMAL(10,2))
, 2) AS MOM_Percentage

FROM
monthly_data curr left join monthly_data prev on curr.rn=prev.rn-1

```

Results		Messages			
	current_year	current_month	current_total	previous_total	MOM_Percentage
1	2021	12	4314	4035	6.91
2	2021	11	4035	3796	6.30
3	2021	10	3796	3536	7.35
4	2021	9	3536	3441	2.76
5	2021	8	3441	3366	2.23
6	2021	7	3366	3184	5.72
7	2021	6	3184	2911	9.38
8	2021	5	2911	2755	5.66
9	2021	4	2755	2627	4.87
10	2021	3	2627	2279	15.27
11	2021	2	2279	2332	-2.27
12	2021	1	2332	NULL	NULL

-- Total funded amount

```
SELECT sum(loan_amount) as Total_Funded_Amount
FROM bank_loan_data
```

Results		Messages			
	Total_Funded_Amount				
1	435757075				

-- MTD funded amount

```
SELECT
sum(loan_amount) as MTD_Total_Funded_Amount
FROM bank_loan_data
where year(issue_date)=(select year(max(issue_date)) from bank_loan_data) and month(issue_date)=(select
MONTH(max(issue_date)) from bank_loan_data)
```

Results		Messages			
	MTD_Total_Funded_Amount				
1	53981425				

-- MOM Total funded amount (using LAG() method)

```
with monthly_data as (
```

SELECT

Year(issue_date) as yr,
MONTH(issue_date) as mn,
sum(loan_amount) as monthly_funded_amount

From

bank_loan_data

Group BY

Year(issue_date),
MONTH(issue_date)
)

SELECT

yr as Year,
mn as Month,
monthly_funded_amount as Current_Month_funded_amount,
Lag(monthly_funded_amount,1) over (order by yr, mn) as previous_month_funded_amount,
Round(CAST((monthly_funded_amount-(Lag(monthly_funded_amount,1) over (order by yr,
mn)))*100.0/(Lag(monthly_funded_amount,1) over (order by yr, mn)) AS DECIMAL(10,2)),2) as
MOM_percentage

from monthly_data

order by

yr desc,
mn desc

75

76

77

yr desc,

mn desc

Results

Messages

	Year	Month	Current_Month_funded_amount	previous_month_funded_amount	MOM_percentage
1	2021	12	53981425	47754825	13.04
2	2021	11	47754825	44893800	6.37
3	2021	10	44893800	40907725	9.74
4	2021	9	40907725	38149600	7.23
5	2021	8	38149600	35813900	6.52
6	2021	7	35813900	34161475	4.84
7	2021	6	34161475	31738350	7.63
8	2021	5	31738350	29800800	6.50
9	2021	4	29800800	28875700	3.20
10	2021	3	28875700	24647825	17.15
11	2021	2	24647825	25031650	-1.53
12	2021	1	25031650	NULL	NULL

PROBLEMS

OUTPUT

TERMINAL

TASKS

Flat File

```
-- Total_Amount_Received
```

```
SELECT sum(total_payment) as Total_Amount_Received FROM bank_loan_data;
```

Results	Messages
	Total_Amount_Received
1	473070933

```
-- MTD_Total_Amount_Received
```

```
SELECT sum(total_payment) as MTD_Total_Amount_Received FROM bank_loan_data
where year(issue_date)=(select year(max(issue_date)) from bank_loan_data) and MONTH(issue_date)= (select
month(max(issue_date)) from bank_loan_data);
```

Results	Messages
	MTD_Total_Amount_Received
1	58074380

```
-- MOM_Total_Amount_Received
```

```
with monthly_data as (
```

SELECT

```
year(issue_date) as yr,  
month(issue_date) as mn,  
sum(total_payment) as monthly_payment_received,  
row_number() over( partition by year(issue_date) order by year(issue_date),month(issue_date) ) as rn
```

FROM

bank_loan_data

group BY

```
year(issue_date),  
month(issue_date)  
)
```

SELECT

```
curr.yr as Year,  
curr.mn as Month,  
curr.monthly_payment_received as cuurent_month_payment_received,  
prev.monthly_payment_received as previous_month_payment_received,  
ROUND(CAST(((curr.monthly_payment_received -  
prev.monthly_payment_received)*100.0/prev.monthly_payment_received) as DECIMAL(10,2)),2) as  
MOM_percentage
```

FROM

```
monthly_data curr left join monthly_data prev on curr.rn = prev.rn-1
```

```
112 monthly_data curr left join monthly_data prev on curr.rn = prev.rn-1
```

Results		Messages			
	Year	Month	cuurent_month_payment_received	previous_month_payment_received	MOM_percentage
1	2021	1	27578836	27717745	-0.50
2	2021	2	27717745	32264400	-14.09
3	2021	3	32264400	32495533	-0.71
4	2021	4	32495533	33750523	-3.72
5	2021	5	33750523	36164533	-6.68
6	2021	6	36164533	38827220	-6.86
7	2021	7	38827220	42682218	-9.03
8	2021	8	42682218	43983948	-2.96
9	2021	9	43983948	49399567	-10.96
10	2021	10	49399567	50132030	-1.46
11	2021	11	50132030	58074380	-13.68
12	2021	12	58074380	NULL	NULL

-- AVG interest rate

```
SELECT AVG(int_rate) as Avg_Interest_Rate from bank_loan_data;
```

Results Messages	
	Avg_Interest_Rate ▾
1	12.05

```
-- MTD_Avg_Interest_Rate
```

```
SELECT round(AVG(int_rate)*100,2) as MTD_Avg_Interest_Rate FROM bank_loan_data
where year(issue_date)=(select year(max(issue_date)) from bank_loan_data) and MONTH(issue_date)= (select
month(max(issue_date)) from bank_loan_data);
```

Results Messages	
	MTD_Avg_Interest_Rate ▾
1	12.36

```
-- MOM_AVG interest rate
```

```
with monthly_data as (
```

```
SELECT
```

```
year(issue_date) as yr,
```

```
month(issue_date) as mn,
```

```
AVG(int_rate) as monthly_AVG_interest_rate,
```

```
row_number() over( partition by year(issue_date) order by year(issue_date) desc,month(issue_date) desc ) as
```

```
rn
```

```
FROM
```

```
bank_loan_data
```

```
group BY
```

```
year(issue_date),
```

```
month(issue_date)
```

```
)
```

```
SELECT
```

```
curr.yr as Year,
```

```
curr.mn as Month,
```

```
round(curr.monthly_AVG_interest_rate*100,2) as cuurent_month_AVG_interest_rate,
```

```
round(prev.monthly_AVG_interest_rate*100,2) as previous_AVG_interest_rate,
```

```
ROUND(CAST(((curr.monthly_AVG_interest_rate -
prev.monthly_AVG_interest_rate)*100.0/prev.monthly_AVG_interest_rate) as DECIMAL(10,2)),2) as
MOM_percentage
```

FROM

```
monthly_data curr left join monthly_data prev on curr.rn = prev.rn-1
```

Results		Messages			
	Year	Month	current_month_AVG_interest_rate	previous_AVG_interest_rate	MOM_percentage
1	2021	12	12.36	11.94	3.47
2	2021	11	11.94	12.02	-0.69
3	2021	10	12.02	12	0.17
4	2021	9	12	12.3	-2.41
5	2021	8	12.3	12.24	0.51
6	2021	7	12.24	12.27	-0.30
7	2021	6	12.27	12.26	0.13
8	2021	5	12.26	11.74	4.40
9	2021	4	11.74	11.86	-0.99
10	2021	3	11.86	11.72	1.17
11	2021	2	11.72	11.46	2.27
12	2021	1	11.46	NULL	NULL

-- AVG_DTI

```
SELECT round(AVG(dti)*100,2) as Avg_DTI from bank_loan_data;
```

Results		Messages	
	Avg_DTI		
1	13.33		

-- MTD_Avg_dti

```
SELECT ROUND(AVG(dti) * 100, 2) AS MTD_Avg_DTI
```

```
FROM bank_loan_data
```

```
WHERE YEAR(issue_date) = (SELECT YEAR(MAX(issue_date)) FROM bank_loan_data)
```

```
AND MONTH(issue_date) = (SELECT MONTH(MAX(issue_date)) FROM bank_loan_data);
```


Results Messages

	MTD_Avg_DTI	▼
1	13.67	

-- MOM_AVG_DTI

with monthly_data as (

SELECT

year(issue_date) as yr,

month(issue_date) as mn,

AVG(dti) as monthly_AVG_DTI,

row_number() over(partition by year(issue_date) order by year(issue_date) desc,month(issue_date) desc) as

rn

FROM

bank_loan_data

group BY

year(issue_date),

month(issue_date)

)

SELECT

curr.yr as Year,

curr.mn as Month,

round(curr.monthly_AVG_DTI*100,2) as cuurent_month_AVG_DTI,

round(prev.monthly_AVG_DTI*100,2) as previous_AVG_DTI,

ROUND(CAST(((curr.monthly_AVG_DTI - prev.monthly_AVG_DTI)*100.0/prev.monthly_AVG_DTI) as

DECIMAL(10,2)),2) as MOM_percentage

FROM

monthly_data curr left join monthly_data prev on curr.rn = prev.rn-1

Results Messages

	Year	Month	cuurent_month_AVG_DTI	previous_AVG_DTI	MOM_percentage
1	2021	12	13.67	13.3	2.73
2	2021	11	13.3	13.41	-0.83
3	2021	10	13.41	13.3	0.88
4	2021	9	13.3	13.35	-0.41
5	2021	8	13.35	13.29	0.44
6	2021	7	13.29	13.24	0.39
7	2021	6	13.24	13.33	-0.67
8	2021	5	13.33	13.22	0.87
9	2021	4	13.22	13.22	0.03
10	2021	3	13.22	13.41	-1.44
11	2021	2	13.41	12.94	3.65
12	2021	1	12.94	NULL	NULL

-- checking_loan_status

Select distinct loan_status from bank_loan_data;

Results Messages

	loan_status
1	Fully Paid
2	Charged Off
3	Current

-- Good loan %

SELECT

(COUNT(CASE

WHEN loan_status IN ('Fully Paid', 'Current') THEN id

END) * 100.0) / COUNT(id) AS Good_loan_percentage

FROM bank_loan_data;

Results Messages

	Good_loan_percentage
1	86.175342181667

-- Good loan applications

SELECT

(COUNT(CASE

WHEN loan_status IN ('Fully Paid', 'Current') THEN id

```
END) ) AS Good_loan_applications
FROM bank_loan_data;
```

Results Messages

	Good_loan_applications
1	33243

-- Good loan funded amount

```
Select sum(loan_amount) as Good_Loan_Funded_Amount
FROM bank_loan_data
WHERE loan_status in ('Fully Paid', 'Current');
```

Results Messages

	Good_Loan_Funded_Amount
1	370224850

-- Good Loan Total Received Amount

```
Select sum(total_payment) as Good_Loan_Total_Received_Amount
FROM bank_loan_data
WHERE loan_status in ('Fully Paid', 'Current');
```

Results Messages

	Good_Loan_Total_Received_Amount
1	435786170

-- Bad loan %

```
SELECT
(COUNT(CASE
  WHEN loan_status IN ('Charged off') THEN id
END) * 100.0) / COUNT(id) AS Bad_loan_percentage
FROM bank_loan_data;
```

Results Messages	
	Bad_loan_percentage ▾
1	13.824657818332

-- Bad loan applications

```
SELECT
(COUNT(CASE
  WHEN loan_status IN ('Charged off') THEN id
END) ) AS Bad_loan_applications
FROM bank_loan_data;
```

Results Messages	
	Bad_loan_applications ▾
1	5333

-- Bad loan funded amount

```
Select sum(loan_amount) as Bad_Loan_Funded_Amount
FROM bank_loan_data
WHERE loan_status in ('Charged off');
```

Results Messages	
	Bad_Loan_Funded_Amount ▾
1	65532225

-- Bad Loan Total Received Amount

```
Select sum(total_payment) as Bad_Loan_Total_Received_Amount
FROM bank_loan_data
WHERE loan_status in ('Charged off');
```

Results Messages

	Bad_Loan_Total_Received_Amount
1	37284763

-- Loan status Grid view

```
SELECT
    loan_status,
    COUNT(id) AS Total_Loan_Applcations,
    SUM(total_payment) AS Total_Amount_Received,
    SUM(loan_amount) AS Total_Funded_Amount,
    AVG(int_rate * 100) AS Interest_Rate,
    AVG(dti * 100) AS DTI
FROM
    bank_loan_data
GROUP BY
    loan_status;
```

	loan_status	Total_Loan_Applcations	Total_Amount_Received	Total_Funded_Amount	Interest_Rate	DTI
1	Fully Paid	32145	411586256	351358350	11.641070773058658	13.167350754394164
2	Charged Off	5333	37284763	65532225	13.878574910931917	14.004732795799695
3	Current	1098	24199914	18866500	15.0993260473588	14.724344262295068

--MTD Loan status Grid view

```
SELECT
    loan_status,
    SUM(total_payment) AS MTD_Total_Amount_Received,
    SUM(loan_amount) AS MTD_Total_Funded_Amount
FROM bank_loan_data
WHERE MONTH(issue_date) = 12 and YEAR(issue_date)=2021
GROUP BY loan_status;
```

Results Messages

	loan_status	MTD_Total_Amount_Received	MTD_Total_Funded_Amount
1	Fully Paid	47815851	41302025
2	Charged Off	5324211	8732775
3	Current	4934318	3946625

-- Loan report Overview

SELECT

```
MONTH(issue_date) AS Month_Munber,  
DATENAME(MONTH, issue_date) AS Month_name,  
COUNT(id) AS Total_Loan_Applications,  
SUM(loan_amount) AS Total_Funded_Amount,  
SUM(total_payment) AS Total_Amount_Received
```

FROM bank_loan_data

GROUP BY MONTH(issue_date), DATENAME(MONTH, issue_date)

ORDER BY MONTH(issue_date);

Results		Messages				
	Month_Munber	Month_name	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received	
1	1	January	2332	25031650	27578836	
2	2	February	2279	24647825	27717745	
3	3	March	2627	28875700	32264400	
4	4	April	2755	29800800	32495533	
5	5	May	2911	31738350	33750523	
6	6	June	3184	34161475	36164533	
7	7	July	3366	35813900	38827220	
8	8	August	3441	38149600	42682218	
9	9	September	3536	40907725	43983948	
10	10	October	3796	44893800	49399567	
11	11	November	4035	47754825	50132030	
12	12	December	4314	53981425	58074380	

-- State Overview

SELECT

```
address_state AS State,  
COUNT(id) AS Total_Loan_Applications,  
SUM(loan_amount) AS Total_Funded_Amount,  
SUM(total_payment) AS Total_Amount_Received
```

FROM bank_loan_data

GROUP BY address_state

ORDER BY address_state;

results Messages

	State ▾	Total_Loan_Applications ▾	Total_Funded_Amount ▾	Total_Amount_Received ▾
1	AK	78	1031800	1108570
2	AL	432	4949225	5492272
3	AR	236	2529700	2777875
4	AZ	833	9206000	10041986
5	CA	6894	78484125	83901234
6	CO	770	8976000	9845810
7	CT	730	8435575	9357612
8	DC	214	2652350	2921854
9	DE	110	1138100	1269136
10	FL	2773	30046125	31601905
11	GA	1355	15480325	16728040
12	HI	170	1850525	2080184

-- Term Overview

SELECT

```
term AS Term,  
COUNT(id) AS Total_Loan_Applications,  
SUM(loan_amount) AS Total_Funded_Amount,  
SUM(total_payment) AS Total_Amount_Received
```

FROM bank_loan_data

GROUP BY term

ORDER BY term;

	Term ▾	Total_Loan_Applications ▾	Total_Funded_Amount ▾	Total_Amount_Received ▾
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

-- Employee Length Overview

SELECT

```
emp_length AS Employee_Length,  
COUNT(id) AS Total_Loan_Applications,  
SUM(loan_amount) AS Total_Funded_Amount,  
SUM(total_payment) AS Total_Amount_Received  
FROM bank_loan_data  
GROUP BY emp_length  
ORDER BY emp_length
```

	Employee_Length ▾	Total_Loan_Applications ▾	Total_Funded_Amount ▾	Total_Amount_Received ▾
1	< 1 year	4575	44210625	47545011
2	1 year	3229	32883125	35498348
3	10+ years	8870	116115950	125871616
4	2 years	4382	44967975	49206961
5	3 years	4088	43937850	47551832
6	4 years	3428	37600375	40964850
7	5 years	3273	36973625	40397571
8	6 years	2228	25612650	27908658
9	7 years	1772	20811725	22584136
10	8 years	1476	17558950	19025777
11	9 years	1255	15084225	16516173

-- Purpose Overview

SELECT

purpose AS PURPOSE,
COUNT(id) AS Total_Loan_Applications,
SUM(loan_amount) AS Total_Funded_Amount,
SUM(total_payment) AS Total_Amount_Received

FROM bank_loan_data

GROUP BY purpose

ORDER BY purpose

home improvement ▾

	PURPOSE ▾	Total_Loan_Applications ▾	Total_Funded_Amount ▾	Total_Amount_Received ▾
1	car	1497	10223575	11324914
2	credit card	4998	58885175	65214084
3	Debt consolidation	18214	232459675	253801871
4	educational	315	2161650	2248380
5	home improvement	2876	33350775	36380930
6	house	366	4824925	5185538
7	major purchase	2110	17251600	18676927
8	medical	667	5533225	5851372
9	moving	559	3748125	3999899
10	other	3824	31155750	33289676
11	renewable_energy	94	845750	898931
12	small business	1776	24123100	23814817

PROBLEMS OUTPUT TERMINAL TASKS

-- Home Ownership Overview

SELECT

home_ownership AS Home_Ownership,
COUNT(id) AS Total_Loan_Applications,
SUM(loan_amount) AS Total_Funded_Amount,
SUM(total_payment) AS Total_Amount_Received

FROM bank_loan_data
GROUP BY home_ownership
ORDER BY home_ownership;

Results Messages

	Home_Ownership ▾	Total_Loan_Applications ▾	Total_Funded_Amount ▾	Total_Amount_Received ▾
1	MORTGAGE	17198	219329150	238474438
2	NONE	3	16800	19053
3	OTHER	98	1044975	1025257
4	OWN	2838	29597675	31729129
5	RENT	18439	185768475	201823056