



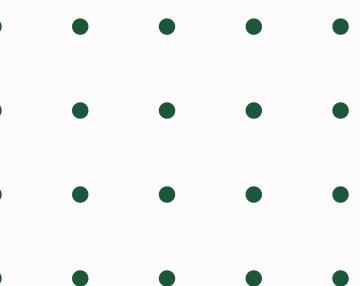
SQL : Structured Query Language

# FINANCIAL ANALYSIS

Under the guidance of Prof. Jaffrin

Besant Technologies

Submitted by Anjali Raj



# SQL

## What is SQL

SQL (Structured Query Language) is a programming language designed for managing and interacting with relational databases. It allows users to query, manipulate, and control data in a structured and standardized manner.

## Uses of SQL

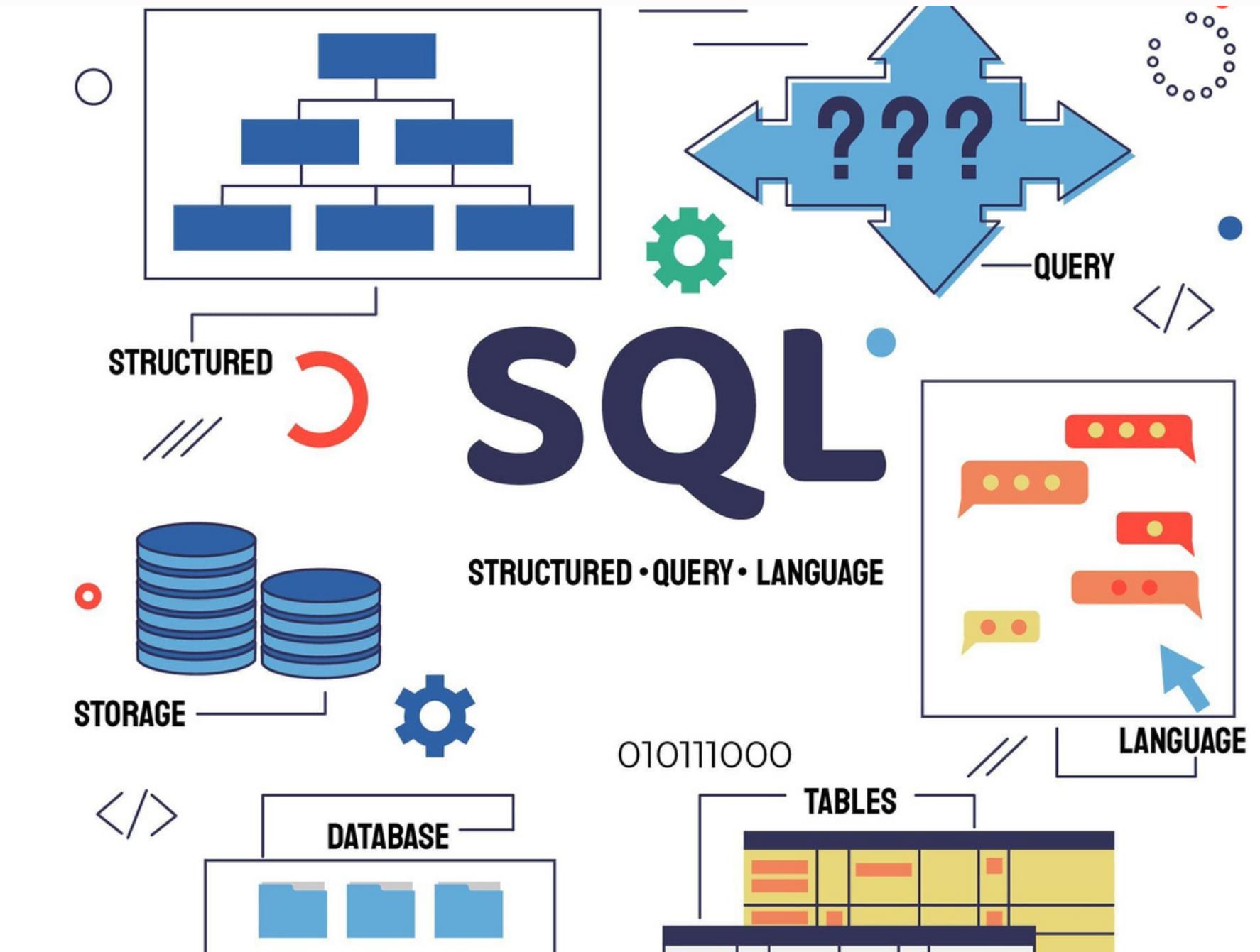
- Data Querying
- Data Modification
- Data Definition
- Data Control
- Data Integrity
- Data Transactions
- Data Aggregation and Reporting
- Stored Procedures and Functions

## Why SQL is used

SQL is used to manage and interact with relational databases, enabling tasks such as data retrieval, modification, definition, access control, data integrity maintenance, transaction management, and report generation. It is a fundamental language in database management and is widely used in software development and data analysis.



# What is Query:



A query in databases, particularly in SQL, is a concise command for interacting with data. It's a specific request to retrieve, modify, or manage information stored in a database. Queries are crucial for tasks like data analysis, reporting, and maintaining database integrity.

# Why is SQL crucial for websites?

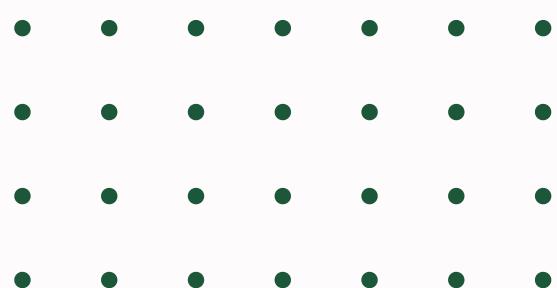
*SQL is crucial for websites because it enables efficient data management. SQL databases, like MySQL or PostgreSQL, store and retrieve data, allowing websites to dynamically generate content, manage user authentication, process e-commerce transactions, and ensure data integrity. SQL supports scalable solutions, essential for growing websites, and facilitates powerful search functionality, analytics, and reporting. It serves as a robust foundation for various web applications, contributing to their functionality, security, and effective handling of user interactions and data.*

# *Financial Analysis*

# Introduction to Financial Analysis

*Financial analysis is the process of evaluating the financial health and performance of a business or investment. It involves assessing various financial statements, ratios, and metrics to gain insights into the company's profitability, liquidity, solvency, and overall financial stability. Financial analysis is crucial for making informed business decisions, attracting investors, and understanding the economic viability of an enterprise.*

*Financial analysis in SQL enables organizations to harness the capabilities of a relational database management system to gain valuable insights into their financial performance. It is a systematic approach to leveraging SQL queries for informed decision-making, risk management, and strategic planning within the financial domain.*



# Entity Relationship Diagram

## COMPANY

- *company\_id INT PRIMARY KEY*
- *company\_name VARCHAR(100)*
- *industry\_sector VARCHAR(100)*

## INCOME TABLE

- *statement\_id INT PRIMARY KEY*
- *company\_id INT*
- *revenue DECIMAL(15, 2)*
- *cost\_of\_goods\_sold DECIMAL(15, 2)*
- *net\_income DECIMAL(15, 2)*
- *preferred\_dividends DECIMAL(15, 2)*
- *common\_shares\_outstanding INT, i\_foreignkey FOREIGN KEY (company\_id) REFERENCES company(company\_id)*

## BALANCE SHEET

- *balance\_sheet\_id INT PRIMARY KEY*
- *company\_id INT*
- *total\_assets DECIMAL(15,2)*
- *total\_liability DECIMAL(15,2)*
- *total\_equity INT,*  
*f\_foreignkey FOREIGN KEY (company\_id) REFERENCES company(company\_id)*

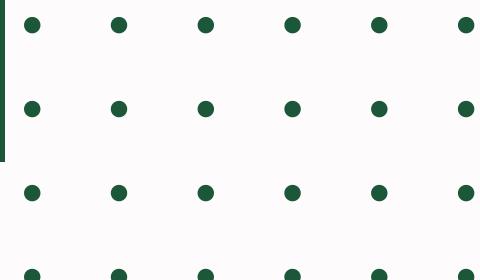


# CASH\_FLOW

- *cash\_flow\_id* INT PRIMARY KEY
- *company\_id* INT
- *net\_cash\_flow* DECIMAL(15,2)
- *expenditure* INT,  
*l\_foreignkey* FOREIGN  
KEY(*company\_id*)  
REFERENCES  
*company*(*company\_id*)

# MARKET\_DATA

- *market\_id* INT PRIMARY KEY
- *company\_id* INT
- *stock\_price* DECIMAL(15, 2)
- *dividends\_per\_share*  
INT,m\_foreignkey FOREIGN  
KEY (*company\_id*)  
REFERENCES  
*company*(*company\_id*));



# COMPANY TABLE

company_id	company_name	industry_sector
1	Global_Finance_Group	Finance
2	Tech_Innovator_Ldt	Technology
3	Fashion_Hub	Retail
4	Health_Solutions	Health
5	Green_Energy	Energy
6	Food_Trends	Food_and_Beverages
7	Transport_Ltd	Transportation
8	Logistics_pvt	Logistics
9	Media_dynamics	Entertainment
10	Gourmet_Delights_Co	Food_and_Beverage
11	Luxury_Auto_Makers	Automotive
12	Education_Innovators	Education
13	Home_Decor_Emporium	Retail
14	Advanced_Robotics_...	Technology

# INCOME TABLE

# BALANCE\_SHEET TABLE

	balance_sheet_id	company_id	total_assets	total_liability	total_equity
▶	29	1	10000000.00	6000000.00	4000000
	30	2	15000000.00	9000000.00	6000000
	31	3	7000000.00	4000000.00	3000000
	32	4	12000000.00	8000000.00	4000000
	33	5	18000000.00	12000000.00	6000000
	34	6	9000000.00	6000000.00	3000000
	35	7	16000000.00	10000000.00	6000000
	36	8	11000000.00	7000000.00	4000000
	37	9	6000000.00	4000000.00	2000000
	38	10	20000000.00	12000000.00	8000000
	39	11	14000000.00	9000000.00	5000000
	40	12	9000000.00	6000000.00	3000000
	41	13	22000000.00	16000000.00	6000000
	42	14	12000000.00	8000000.00	4000000
	NULL	NULL	NULL	NULL	NULL

# CASH FLOW TABLE

	cash_flow_id	company_id	net_cash_flow	expenditure
▶	43	1	2000000.00	500000
	44	2	3500000.00	800000
	45	3	1500000.00	300000
	46	4	2800000.00	600000
	47	5	4000000.00	900000
	48	6	1800000.00	400000
	49	7	3000000.00	700000
	50	8	2200000.00	500000
	51	9	1200000.00	200000
	52	10	3600000.00	1000000
	53	11	2800000.00	600000
	54	12	1700000.00	400000
	55	13	4500000.00	1200000
	56	14	2400000.00	500000

# MARKET\_DATA TABLE

	market_id	company_id	stock_price	dividends_per_share
▶	60	1	50.25	2
	61	2	75.60	1
	62	3	30.80	2
	63	4	110.45	4
	64	5	45.75	2
	65	6	92.30	3
	66	7	65.20	5
	67	8	80.15	1
	68	9	55.50	7
	69	10	120.75	2
	71	11	40.60	1
	72	12	95.90	3
⋮	73	13	70.25	2
⋮	74	14	85.40	4
⋮	⋮	⋮	⋮	⋮

# Query to Update value in the table

- *UPDATE company SET industry\_sector = 'Healthcare' WHERE company\_id = 4;*
- *UPDATE company SET industry\_sector = 'Food\_and\_Beverage' WHERE company\_id = 6;*

company_id	company_name	industry_sector
1	Global_Finance_Group	Finance
2	Tech_Innovator_Ldt	Technology
3	Fashion_Hub	Retail
4	Health_Solutions	Healthcare
5	Green_Energy	Energy
6	Food_Trends	Food_and_Beverage
7	Transport_Ltd	Transportation
8	Logistics_pvt	Logistics
9	Media_dynamics	Entertainment
10	Gourmet_Delights_Co	Food_and_Beverage
11	Luxury_Auto_Makers	Automotive
12	Education_Innovators	Education
13	Home_Decor_Emporium	Retail
14	Advanced_Robotics_...	Technology

# Calculate gross profit margin

```
SELECT(revenue-cost_of_goods_sold)  
/revenue as gross_profit_margin FROM  
income;
```

gross_profit_margin
0.400000
0.437500
0.428571
0.384615
0.388889
0.404762
0.360000
0.396552
0.437500
0.409091
0.416667
0.440000
0.387755
0.387097

# Debt to equity ratio

```
SELECT  
company_id, total_liability  
/total_equity AS  
debt_to_equity_ratio  
FROM balance_sheet;
```

	company_id	debt_to_equity_ratio
▶	1	1.500000
	2	1.500000
	3	1.333333
	4	2.000000
	5	2.000000
	6	2.000000
	7	1.666667
	8	1.750000
	9	2.000000
	10	1.500000
	11	1.800000
	12	2.000000
	13	2.666667
	14	2.000000

# Earnings per share

```
SELECT company_id,  
       (net_income-  
        preferred_dividends)/common_  
        shares_outstanding AS  
        earnings_per_share FROM  
        income;
```

	company_id	earnings_per_share
▶	1	14.800000
	2	21.133333
	3	14.812500
	4	18.125000
	5	22.027778
	6	16.466667
	7	17.312500
	8	17.981818
	9	21.466667
	10	17.800000
	11	17.300000
	12	18.688889
	13	17.068182
	14	16.942857

# Positive free cash flow

```
SELECT  
c.company_name,cf.net_cash_flow,  
cf.expenditure FROM company C  
JOIN cash_flow CF ON  
C.company_id = CF.company_id  
WHERE expenditure < net_cash_flow  
order by net_cash_flow desc limit 4;
```

company_name	net_cash_flow	expenditure
Home_Decor_Emporium	4500000.00	1200000
Green_Energy	4000000.00	900000
Gourmet_Delights_Co	3600000.00	1000000
Tech_Innovator_Ldt	3500000.00	800000

# Calculate returns on assets

```
SELECT  
income.company_id,  
      (net_income/  
total_assets) AS roa  
FROM income JOIN  
balance_sheet ON  
income.company_id  
      =  
balance_sheet.comp  
any_id;
```

	company_id	roa
▶	1	0.150000
	2	0.213333
	3	0.171429
	4	0.183333
	5	0.222222
	6	0.166667
	7	0.175000
	8	0.181818
	9	0.216667
	10	0.180000
	11	0.200000
	12	0.188889
	13	0.172727
	14	0.200000

# Free cash flow

```
SELECT  
company_id,net_cash_  
flow - expenditure AS  
free_cash_flow FROM  
cash_flow;
```

	company_id	free_cash_flow
▶	1	1500000.00
	2	2700000.00
	3	1200000.00
	4	2200000.00
	5	3100000.00
	6	1400000.00
	7	2300000.00
	8	1700000.00
	9	1000000.00
	10	2600000.00
	11	2200000.00
	12	1300000.00
	13	3300000.00
	14	1900000.00

# Calculate dividend yield

```
SELECT  
company.company_name,  
(dividends_per_share /  
stock_price) * 100 AS  
dividend_yield FROM  
company inner JOIN  
market_data ON  
company.company_id=market  
_data.company_id;
```

	company_name	dividend_yield
▶	Global_Finance_Group	3.9801
	Tech_Innovator_Ldt	1.3228
	Fashion_Hub	6.4935
	Health_Solutions	3.6215
	Green_Energy	4.3716
	Food_Trends	3.2503
	Transport_Ltd	7.6687
	Logistics_pvt	1.2477
	Media_dynamics	12.6126
	Gourmet_Delights_Co	1.6563
	Luxury_Auto_Makers	2.4631
	Education_Innovators	3.1283
	Home_Decor_Emporium	2.8470
	Advanced_Robotics_...	4.6838

# Average income of each company

```
SELECT company_id,  
AVG(net_income) as  
avg_net_income  
FROM income  
GROUP BY  
company_id;
```

	company_id	avg_net_income
▶	1	1500000.000000
	2	3200000.000000
	3	1200000.000000
	4	2200000.000000
	5	4000000.000000
	6	1500000.000000
	7	2800000.000000
	8	2000000.000000
	9	1300000.000000
	10	3600000.000000
	11	2800000.000000
	12	1700000.000000
	13	3800000.000000
	14	2400000.000000

# Average stock price and dividends per share for each industry sector

```
SELECT c.industry_sector,  
       AVG(m.stock_price) as  
             avg_stock_price,  
       AVG(m.dividends_per_share)  
             as avg_dividends_per_share  
FROM market_data m JOIN  
      company c ON  
      m.company_id =  
c.company_id GROUP BY  
      c.industry_sector;
```

	industry_sector	avg_stock_price	avg_dividends_per_share
▶	Finance	50.250000	2.0000
	Technology	80.500000	2.5000
	Retail	50.525000	2.0000
	Healthcare	110.450000	4.0000
	Energy	45.750000	2.0000
	Food_and_Beverage	106.525000	2.5000
	Transportation	65.200000	5.0000
	Logistics	80.150000	1.0000
	Entertainment	55.500000	7.0000
	Automotive	40.600000	1.0000
	Education	95.900000	3.0000

# Net Income to Total Assets Ratio

```
SELECT income.company_id,
       AVG(net_income /
            total_assets) AS
    net_income_to_assets_ratio
   FROM income JOIN
        balance_sheet ON
    income.company_id =
balance_sheet.company_id
 GROUP BY company_id;
```

	company_id	net_income_to_assets_ratio
▶	1	0.1500000000
	2	0.2133330000
	3	0.1714290000
	4	0.1833330000
	5	0.2222220000
	6	0.1666670000
	7	0.1750000000
	8	0.1818180000
	9	0.2166670000
	10	0.1800000000
	11	0.2000000000
	12	0.1888890000
	13	0.1727270000
	14	0.2000000000

# Number of companies in each industry sector

```
SELECT  
industry_sector,  
COUNT(DISTINCT  
company_id) AS  
num_companies  
FROM company  
GROUP BY  
industry_sector;
```

industry_sector	num_companies
Automotive	1
Education	1
Energy	1
Entertainment	1
Finance	1
Food_and_Beverage	2
Healthcare	1
Logistics	1
Retail	2
Technology	2
Transportation	1

# Company with highest revenue

```
SELECT company_id,  
       SUM(revenue) AS  
total_revenue FROM  
income GROUP BY  
company_id ORDER BY  
total_revenue DESC  
LIMIT 1;
```

	company_id	total_revenue
▶	13	9800000.00

# Revenue rank

```
SELECT  
company_id,net_income,  
RANK() OVER (ORDER BY  
    net_income DESC) AS  
net_income_rank FROM  
income;
```

	company_id	net_income	net_income_rank
▶	5	4000000.00	1
	13	3800000.00	2
	10	3600000.00	3
	2	3200000.00	4
	7	2800000.00	5
	11	2800000.00	5
	14	2400000.00	7
	4	2200000.00	8
	8	2000000.00	9
	12	1700000.00	10
	1	1500000.00	11
	6	1500000.00	11
	9	1300000.00	13
	3	1200000.00	14

# Query To View The String Values In Upper And Lower Case

```
SELECT  
ucase(company_name)  
AS Upper_case FROM  
company;
```

```
SELECT  
lcase(company_name)  
AS Lower_case from  
company;
```

Upper_case	Lower_case
GLOBAL_FINANCE_GROUP	global_finance_group
TECH_INNOVATOR_LDT	tech_innovator_ldt
FASHION_HUB	fashion_hub
HEALTH_SOLUTIONS	health_solutions
GREEN_ENERGY	green_energy
FOOD_TRENDS	food_trends
TRANSPORT_LTD	transport_ltd
LOGISTICS_PVT	logistics_pvt
MEDIA_DYNAMICS	media_dynamics
GOURMET_DELIGHTS_CO	gourmet_delights_co
LUXURY_AUTO_MAKERS	luxury_auto_makers
EDUCATION_INNOVATORS	education_innovators
HOME_DECOR_EMPORIUM	home_decor_emporium
ADVANCED_ROBOTICS_S...	advanced_robotics_...

# CONCLUSION

**In concluding the project's financial analysis using SQL, we have successfully harnessed the power of structured query language to extract, manipulate, and interpret crucial financial data. Through the execution of SQL queries, we've gained a comprehensive understanding of the organization's financial landscape, uncovering key insights that are instrumental in strategic decision-making.**

**The SQL analysis allowed us to efficiently aggregate and summarize financial information, providing a clear overview of revenue streams, cost structures, and overall profitability. By leveraging SQL's capabilities for data transformation and aggregation, we've been able to streamline complex financial datasets into actionable information.**

**One of the notable achievements of this analysis is the ability to identify trends and anomalies within the financial data. SQL queries facilitated the detection of patterns over time, enabling us to recognize areas of growth and potential risks. This insight empowers stakeholders to proactively address challenges and capitalize on opportunities in a dynamic business environment.**

**In conclusion, the project's financial analysis in SQL has proven to be a valuable asset in transforming raw financial data into actionable intelligence. The methodologies employed and insights gained lay the groundwork for informed decision-making, facilitating the organization's ability to navigate financial challenges and pursue strategic initiatives with confidence.**



# THANK YOU