



ATLIQ
Hardware

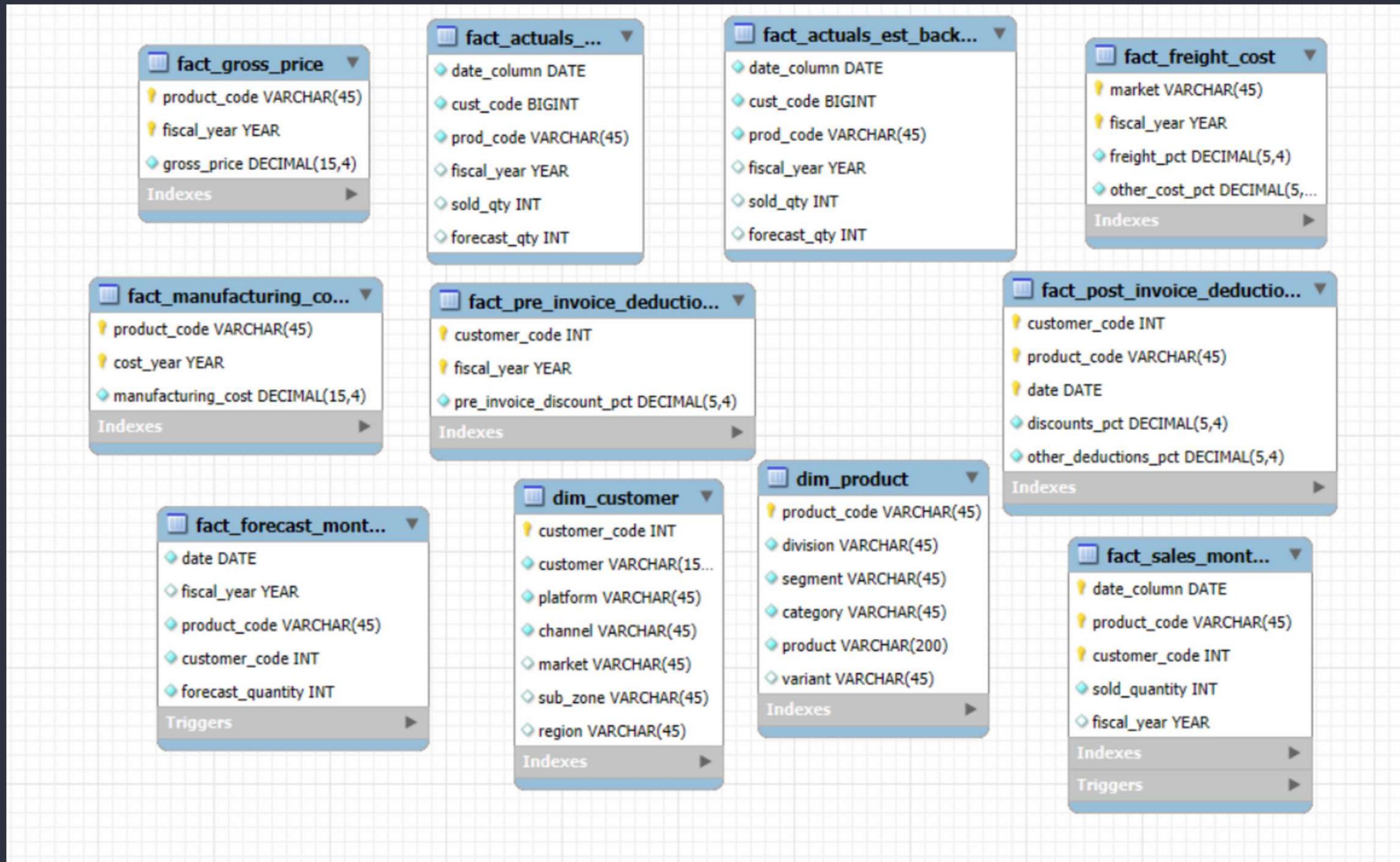
SQL-BASED ANALYTICS FOR FINANCE & SUPPLY CHAIN

Presented by: Arjit Dabral

INTRODUCTION

- This project focuses on analyzing sales and supply chain performance using SQL. The key areas are Finance Analytics and Supply Chain Analytics, including forecasting accuracy, customer sales, and top-performing products.
- Tech Stack:
- MySQL for database queries.
- SQL Procedures, Views, and Functions.

ENTITY RELATIONSHIP DIAGRAM



FINANCE ANALYTICS

PROBLEM STATEMENT

The goal of this section is to analyze sales data by calculating the gross price, applying pre-invoice and post-invoice discounts, and then deriving the net sales. The views `post_inv_discount` and `net_sales` are created to streamline the process of analyzing these data points and provide financial insights on sales performance.

FUNCTIONS AND PROCEDURES

Function: get_fiscal_year

This function calculates the fiscal year based on a calendar date.

The fiscal year is calculated by adding 4 months to the current date.

```
delimiter $$  
  
CREATE FUNCTION get_fiscal_year(calendar_date DATE) RETURNS INT  
DETERMINISTIC  
BEGIN  
    DECLARE fiscal_year INT;  
    SET fiscal_year = YEAR(DATE_ADD(calendar_date, INTERVAL 4 MONTH));  
    RETURN fiscal_year;  
END$$  
  
delimiter ;
```

FUNCTIONS AND PROCEDURES

Procedure: *get_badge*

This procedure calculates the total quantity sold for a specific market and fiscal year, assigning a badge based on the sales quantity.

```
delimiter $$  
CREATE PROCEDURE get_batch(market TEXT, fiscal_year INT)  
BEGIN  
    SELECT  
        SUM(s.sold_quantity) AS total_sold_quantity,  
        CASE  
            WHEN SUM(s.sold_quantity) > 500000 THEN 'GOLD BADGE'  
            ELSE 'SILVER BADGE'  
        END AS MARKET_BADGE  
    FROM fact_sales_monthly AS s  
    JOIN dim_customer AS c ON s.customer_code = c.customer_code  
    WHERE get_fiscal_year(s.date_column) = fiscal_year  
        AND c.market = market;  
END$$  
delimiter ;
```

```
call get_batch('India',2021);
```

Result Grid		
	total_sold_quantity	MARKET_BADGE
▶	13751429	GOLD BADGE

KEY QUERIES AND VIEWS

View: sales_pre_invoice_discount

This view calculates pre-invoice discounts based on sales data.

```
CREATE VIEW sales_pre_invoice_discount AS
SELECT
    s.date_column, s.fiscal_year, s.product_code, s.customer_code,
    p.product, p.variant, c.market, s.sold_quantity,
    g.gross_price AS gross_price_per_item,
    ROUND(g.gross_price * s.sold_quantity, 2) AS gross_price_total,
    pre.pre_invoice_discount_pct
FROM fact_sales_monthly AS s
JOIN dim_product AS p ON s.product_code = p.product_code
JOIN fact_gross_price AS g ON s.product_code = g.product_code
    AND g.fiscal_year = s.fiscal_year
JOIN fact_pre_invoice_deductions AS pre ON pre.customer_code = s.customer_code
    AND pre.fiscal_year = s.fiscal_year
JOIN dim_customer AS c ON c.customer_code = s.customer_code;
```

KEY QUERIES AND VIEWS

View: sales_pre_invoice_discount

This view calculates pre-invoice discounts based on sales data.

Result Grid											
	date_column	fiscal_year	product_code	customer_code	product	variant	market	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct
▶	2020-06-01	2020	A0118150102	70009134	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 34	19.8577	675.16	0.2029	
	2020-08-01	2020	A0118150102	70009134	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 2	19.8577	39.72	0.2029	
	2019-09-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 20	19.8577	397.15	0.1086	
	2019-10-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 11	19.8577	218.43	0.1086	
	2019-11-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 27	19.8577	536.16	0.1086	
	2020-01-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 20	19.8577	397.15	0.1086	
	2020-02-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 3	19.8577	59.57	0.1086	
	2020-03-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 1	19.8577	19.86	0.1086	
	2020-05-01	2020	A0118150102	70009133	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	Newzealar 2	19.8577	39.72	0.1086	

KEY QUERIES AND VIEWS

View: *post_inv_discount*

The post_inv_discount view calculates the net invoice sales by applying pre-invoice discounts and calculates the post-invoice discount percentage.

```
CREATE VIEW post_inv_discount AS
SELECT
    s.* , -- All columns from sales_pre_invoice_discount view
    s.gross_price_total - (s.pre_invoice_discount_pct * s.gross_price_total) AS net_invoice_sales, -- Apply pre-invoice discount
    (po.discounts_pct + po.other_deductions_pct) AS post_invoice_discount_pct -- Combine other deductions and discounts
FROM sales_pre_invoice_discount AS s
JOIN fact_post_invoice_deductions AS po
    ON po.date = s.date_column
    AND po.customer_code = s.customer_code
    AND po.product_code = s.product_code;
```

n	fiscal_year	product_code	customer_code	product	variant	market	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	953	25.9354	24716.44	0.0824	22679.805344	0.3262
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	557	25.9354	14446.02	0.0824	13255.667952	0.3013
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	569	25.9354	14757.24	0.0824	13541.243424	0.3349
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	360	25.9354	9336.74	0.0824	8567.392624	0.3621
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	805	25.9354	20878.00	0.0824	19157.652800	0.3455
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	782	25.9354	20281.48	0.0824	18610.286048	0.3360
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	406	25.9354	10529.77	0.0824	9662.116952	0.3370
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	940	25.9354	24379.28	0.0824	22370.427328	0.3301
	2018	A7118160101	70002017	AQ Wi Power Dx1	Standard	India	279	25.9354	7235.98	0.0824	6639.735248	0.3657
	2018	A6218160102	70002017	AQ Digit SSD	Plus	India	296	14.3429	4245.50	0.0824	3895.670800	0.3935
	2018	A6218160102	70002017	AQ Digit SSD	Plus	India	697	14.3429	9997.00	0.0824	9173.247200	0.3355
	2018	A6218160102	70002017	AQ Digit SSD	Plus	India	399	14.3429	5722.82	0.0824	5251.259632	0.3050
	2018	A6218160102	70002017	AO Dlait SSD	Plus	India	106	14.3429	1520.35	0.0824	1395.073160	0.3577

KEY QUERIES AND VIEWS

View: *net_sales*

The net_sales view builds upon the post_inv_discount view by applying the post-invoice discounts and deriving the net sales. This is the final value of sales after all discounts are considered.

```
CREATE VIEW net_sales AS
SELECT
  *,
  (1 - post_invoice_discount_pct) * net_invoice_sales AS netsales -- Final net sales calculation
FROM post_inv_discount;
```

Use Case:

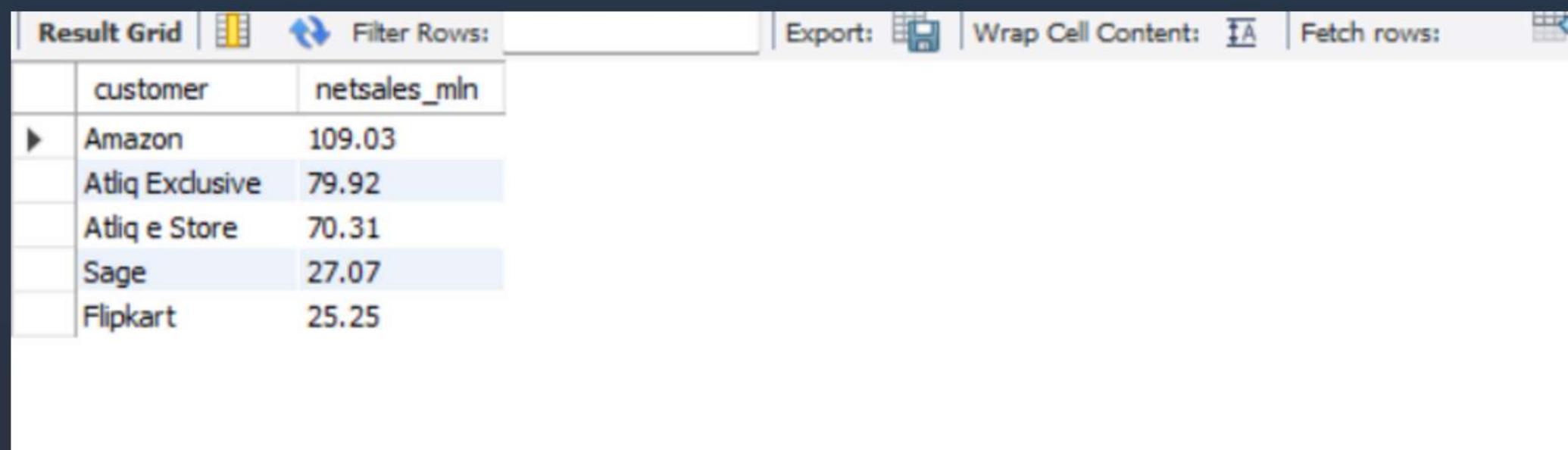
The net_sales view is essential for getting the final net sales numbers for each product, customer, and date after both pre- and post-invoice discounts. This view simplifies queries related to overall revenue analysis by providing a single field for net sales.

KEY QUERIES AND VIEWS

Top 5 Customers by Net Sales

This query retrieves the top 5 customers by net sales for the fiscal year 2021.

```
SELECT
    c.customer,
    ROUND(SUM(netsales) / 1000000, 2) AS netsales_mln
FROM net_sales n
JOIN dim_customer c ON n.customer_code = c.customer_code
WHERE fiscal_year = 2021
GROUP BY c.customer
ORDER BY netsales_mln DESC
LIMIT 5;
```



The screenshot shows a database query results grid with the following data:

	customer	netsales_mln
▶	Amazon	109.03
	Atiq Exclusive	79.92
	Atiq e Store	70.31
	Sage	27.07
	Flipkart	25.25

KEY QUERIES AND VIEWS

Top 5 Markets by Net Sales

This query retrieves the top 5 markets by net sales for the fiscal year 2021.

```
SELECT
    market,
    SUM(netsales) AS total_net_sales
FROM net_sales
WHERE fiscal_year = 2021
GROUP BY market
ORDER BY total_net_sales DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	market	total_net_sales
▶	India	210671647.6315448533
	USA	132049294.8651417843
	South Korea	64010020.2278520734
	Canada	45890890.2142843892
	United Kingdom	44730673.4950820142

KEY QUERIES AND VIEWS

Write a query to show a report displaying:

1. Total net sales per customer in millions for the year 2021.
2. Percentage contribution of each customer to the total net sales for that year.

```
WITH cte1 AS (
    SELECT
        c.customer,
        ROUND(SUM(netsales)/1000000, 2) AS netsales_mln
    FROM net_sales n
    JOIN dim_customer c ON n.customer_code = c.customer_code
    WHERE fiscal_year = 2021
    GROUP BY c.customer
)SELECT
    *,
    netsales_mln * 100 / SUM(netsales_mln) OVER () AS netsales_pct
FROM cte1
ORDER BY netsales_mln DESC;
```

	customer	netsales_mln	netsales_pct
▶	Amazon	109.03	13.233402
	Atliq Exclusive	79.92	9.700206
	Atliq e Store	70.31	8.533803
	Sage	27.07	3.285593
	Flipkart	25.25	3.064692
	Leader	24.52	2.976089
	Neptune	21.01	2.550067
	Ebay	19.88	2.412914
	Electricalsociety	16.25	1.972327
	Synthetic	16.10	1.954121
	Electricalslytical	15.64	1.898289
	Accclaimed Sto...	14.32	1.738075
	Propel	14.14	1.716228
	Novus	12.91	1.566938
	Expression	12.90	1.565724
	Reliance Digital	12.75	1.547518
	walmart	12.63	1.532953

KEY QUERIES AND VIEWS

Write a stored procedure for getting top n products in each division by their quantity sold in each division

Input Parameter:

The procedure takes an input parameter n, which specifies the number of top products you want to retrieve for each division.

Common Table Expression (CTE) 1: cte2:

Purpose: This CTE calculates the total quantity sold for each product in each division by summing the sold_quantity from the fact_sales_monthly table.

Join: It joins the fact_sales_monthly table (which contains sales data) with the dim_product table (which contains product and division details).

Grouping: The data is grouped by product and division so that we can calculate the total sales for each product in each division.

Common Table Expression (CTE) 2: cte3:

Purpose: This CTE adds a row number (ROW_NUMBER()) to each product in a division, ordered by the total quantity sold in descending order.

Window Function: The ROW_NUMBER() function is partitioned by division and ordered by the total quantity sold (total_qty DESC). This ensures that each product is ranked within its division based on sales.

Main Query:

The main query selects products where the row number (rn) is less than or equal to n, retrieving the top n products by sales for each division.

For example, if n = 5, the query will return the top 5 products in terms of sales for each division.

```
DROP PROCEDURE IF EXISTS get_top_prod;
DELIMITER $$

CREATE PROCEDURE get_top_prod(n INT)
BEGIN
    WITH cte2 AS (
        SELECT
            p.division,
            p.product,
            SUM(fs.sold_quantity) AS total_qty
        FROM dim_product AS p
        JOIN fact_sales_monthly AS fs ON p.product_code = fs.product_code
        WHERE fs.fiscal_year = 2021
        GROUP BY p.product, p.division
    ),
    cte3 AS (
        SELECT
            *,
            ROW_NUMBER() OVER (PARTITION BY division ORDER BY total_qty DESC) AS rn
        FROM cte2
    )
    SELECT *
    FROM cte3
    WHERE rn <= n;
END $$

DELIMITER ;
```

KEY QUERIES AND VIEWS

Write a stored procedure for getting top n products in each division by their quantity sold in each division

OUTPUT

call get_top_prod(5)

	division	product	total_qty	rn
N & S	AQ Pen Drive DRC	2034569	1	
	AQ Digit SSD	1240149	2	
	AQ Clx1	1238683	3	
	AQ Neuer SSD	1225985	4	
	AQ Clx2	1201025	5	
P & A	AQ Gamers Ms	2477098	1	
	AQ Maxima Ms	2461991	2	
	AQ Master wireless x1 Ms	2448784	3	
	AQ Master wired x1 Ms	2447468	4	
	AQ Lite Ms	2443425	5	
PC	AQ Digit	135092	1	
	AQ Gen Y	135031	2	
	AQ Elite	134431	3	
	AQ Gen X	134264	4	
	AQ Velocity	101757	5	

CONCLUSION

The finance analytics section provides deep insights into the effects of pre- and post-invoice discounts on sales. The use of views like `post_inv_discount` and `net_sales` helps to simplify the analysis by abstracting complex discount calculations into reusable components. By leveraging these views, you can quickly query financial performance metrics and derive meaningful insights for decision-making.

SUPPLY CHAIN ANALYTICS

PROBLEM STATEMENT

The supply chain analytics section focuses on analyzing ***actual sales*** vs. ***forecast data*** and determining ***forecast accuracy***. This analysis helps in identifying areas where forecasting needs improvement.

DATA INTEGRATION AND TABLE CREATION

Table : *fact_actuals_est*

This table integrates actual sales data from *fact_sales_monthly* and forecast data from *fact_forecast_monthly*.

Updating null values

```
UPDATE fact_actuals_est  
SET sold_qty = 0  
WHERE sold_qty IS NULL;  
  
UPDATE fact_actuals_est  
SET forecast_qty = 0  
WHERE forecast_qty IS NULL;
```

```
CREATE TABLE fact_actuals_est AS  
(SELECT  
    fs.date_column AS date_column,  
    fs.customer_code AS cust_code,  
    fs.product_code AS prod_code,  
    fs.fiscal_year AS fiscal_year,  
    fs.sold_quantity AS sold_qty,  
    ff.forecast_quantity AS forecast_qty  
  FROM fact_sales_monthly AS fs  
  LEFT JOIN fact_forecast_monthly AS ff  
    ON fs.customer_code = ff.customer_code  
    AND fs.product_code = ff.product_code  
    AND fs.date_column = ff.date  
UNION  
SELECT  
    ff.date AS date_column,  
    ff.customer_code AS cust_code,  
    ff.product_code AS prod_code,  
    ff.fiscal_year AS fiscal_year,  
    fs.sold_quantity AS sold_qty,  
    ff.forecast_quantity AS forecast_qty  
  FROM fact_forecast_monthly AS ff  
  LEFT JOIN fact_sales_monthly AS fs  
    ON fs.customer_code = ff.customer_code  
    AND fs.product_code = ff.product_code  
    AND fs.date_column = ff.date);
```

DATA INTEGRATION AND TABLE CREATION

Table : *fact_actuals_est*

This table integrates actual sales data from *fact_sales_monthly* and forecast data from *fact_forecast_monthly*.

	date_column	cust_code	prod_code	fiscal_year	sold_qty	forecast_qty
▶	2017-09-01	70002017	A0118150101	2018	51	18
	2017-09-01	70002017	A0118150102	2018	77	62
	2017-09-01	70002017	A0118150103	2018	54	11
	2017-09-01	70002017	A0118150104	2018	42	25
	2017-09-01	70002017	A0418150101	2018	67	117
	2017-09-01	70002017	A0418150102	2018	17	102
	2017-09-01	70002017	A0418150103	2018	25	82
	2017-09-01	70002017	A0418150104	2018	17	96
	2017-09-01	70002017	A0418150105	2018	51	80
	2017-09-01	70002017	A1018150101	2018	54	25
	2017-09-01	70002017	A1018150102	2018	26	49
	2017-09-01	70002017	A1018150103	2018	15	47
	2017-09-01	70002017	A1118150201	2018	47	44
	2017-09-01	70002017	A1118150202	2018	8	19
	2017-09-01	70002017	A1119150203	2018	37	72
	2017-09-01	70002017	A1219150301	2018	56	60
	2017-09-01	70002017	A1618150101	2018	3	3
	2017-09-01	70002017	A1618150102	2018	2	1
	2017-09-01	70002017	A1618150103	2018	1	7
	2017-09-01	70002017	A1618150104	2018	1	4
	2017-09-01	70002017	A1718150201	2018	5	8
	2017-09-01	70002017	A2118150101	2018	281	161
	2017-09-01	70002017	A2118150102	2018	431	471
	2017-09-01	70002017	A2118150103	2018	197	758

KEY QUERIES AND FORECAST ANALYSIS

Forecast Error Table for 2021:

This table calculates forecast errors for the year 2021.

```
CREATE TEMPORARY TABLE forecast_err_table AS
SELECT
    cust_code AS customer_code,
    SUM(sold_qty) AS total_sold_qty,
    SUM(forecast_qty) AS total_forecast_qty,
    SUM(forecast_qty - sold_qty) AS net_err,
    SUM(forecast_qty - sold_qty) * 100 / SUM(forecast_qty) AS net_err_pct,
    SUM(ABS(forecast_qty - sold_qty)) AS abs_err,
    SUM(ABS(forecast_qty - sold_qty)) * 100 / SUM(forecast_qty) AS abs_err_pct
FROM fact_actuals_est
WHERE fiscal_year = 2021
GROUP BY customer_code;
```

KEY QUERIES AND FORECAST ANALYSIS

Forecast Error Table for 2021:

It calculates various metrics such as net error, absolute error, and error percentages based on actual sales and forecasted sales.

```
CREATE TEMPORARY TABLE forecast_err_table AS
SELECT
    cust_code AS customer_code,
    SUM(sold_qty) AS total_sold_qty,
    SUM(forecast_qty) AS total_forecast_qty,
    SUM(forecast_qty - sold_qty) AS net_err,
    SUM(forecast_qty - sold_qty) * 100 / SUM(forecast_qty) AS net_err_pct,
    SUM(ABS(forecast_qty - sold_qty)) AS abs_err,
    SUM(ABS(forecast_qty - sold_qty)) * 100 / SUM(forecast_qty) AS abs_err_pct
FROM fact_actuals_est
WHERE fiscal_year = 2021
GROUP BY customer_code;
```

KEY QUERIES AND FORECAST ANALYSIS

Forecast Error Table for 2021:

	customer_code	total_sold_qty	total_forecast_qty	net_err	net_err_pct	abs_err	abs_err_pct
▶	70002017	685218	728761	43543	5.9749	417293	57.2606
	70002018	701283	742759	41476	5.5840	420936	56.6719
	70003181	341482	306647	-34835	-11.3600	200709	65.4528
	70003182	350840	310630	-40210	-12.9447	194688	62.6752
	70004069	48332	38362	-9970	-25.9893	26050	67.9057
	70004070	50355	38714	-11641	-30.0692	27483	70.9898
	70005163	91411	81908	-9503	-11.6020	50279	61.3847
	70006157	193424	141114	-52310	-37.0693	105552	74.7991
	70006158	202406	142913	-59493	-41.6288	107913	75.5096
	70007198	345667	228104	-117563	-51.5392	188585	82.6750
	70007199	358064	236637	-121427	-51.3136	194953	82.3848
	70008169	244355	222226	-22129	-9.9579	137573	61.9068
	70008170	245921	223965	-21956	-9.8033	137200	61.2596
	70009133	100607	109662	9055	8.2572	62523	57.0143
	70009134	103747	110791	7044	6.3579	60726	54.8113
	70010047	116158	134945	18787	13.9220	75769	56.1481
	70011193	202295	202380	85	0.0420	117075	57.8491
	70011194	202537	200844	-1693	-0.8429	117685	58.5952
	70012042	109259	77193	-32066	-41.5400	59532	77.1210
	70012043	114116	80288	-33828	-42.1333	64402	80.2137
	70013125	101658	123428	21770	17.6378	67546	54.7250
	70013126	103808	125312	21504	17.1604	70542	56.2931
	70014142	81324	44740	-36584	-81.7702	47940	107.1524
	70014143	81942	45345	-36597	-80.7079	48575	107.1232

KEY QUERIES AND FORECAST ANALYSIS

Forecast Accuracy Analysis by Customer

```
SELECT
    e.*,
    c.customer,
    c.market,
    IF(abs_err_pct > 100, 0, 100 - abs_err_pct) AS forecast_accuracy
FROM forecast_err_table e
JOIN dim_customer c USING (customer_code)
ORDER BY forecast_accuracy DESC;
```

Forecast Accuracy: A metric that measures how closely the forecast matches the actual sales, calculated as $100 - \text{abs_err_pct}$ where `abs_err_pct` is the absolute percentage error between forecasted and actual sales.

KEY QUERIES AND FORECAST ANALYSIS

Forecast Accuracy Analysis by Customer

	customer_code	total_sold_qty	total_forecast_qty	net_err	net_err_pct	abs_err	abs_err_pct	customer	market	forecast_accuracy
▶	90013120	109547	133532	23985	17.9620	70467	52.7716	Coolblue	Italy	47.2284
	70010048	119439	142010	22571	15.8940	75711	53.3139	Atliq e Store	Bangladesh	46.6861
	90023027	236189	279962	43773	15.6353	149303	53.3297	Costco	Canada	46.6703
	90023026	228988	273492	44504	16.2725	146948	53.7303	Relief	Canada	46.2697
	90017051	86823	118067	31244	26.4629	63568	53.8406	Forward Stores	Portugal	46.1594
	90017058	86860	110195	23335	21.1761	59473	53.9707	Mbit	Portugal	46.0293
	90023028	239081	283323	44242	15.6154	153058	54.0224	walmart	Canada	45.9776
	90023024	246397	287233	40836	14.2170	155610	54.1755	Sage	Canada	45.8245
	90013124	110898	136116	25218	18.5268	73826	54.2376	Amazon	Italy	45.7624
	90015146	147152	210507	63355	30.0964	114189	54.2448	Mbit	Norway	45.7552
	90017054	84371	114698	30327	26.4407	62483	54.4761	Flawless Stores	Portugal	45.5239
	70027208	33713	47321	13608	28.7568	25784	54.4874	Atliq e Store	Brazil	45.5126
	90015147	154897	223867	68970	30.8085	122100	54.5413	Chiptec	Norway	45.4587
	80001019	1113979	1275248	161269	12.6461	695779	54.5603	Neptune	China	45.4397
	90015144	160074	225637	65563	29.0568	123257	54.6262	Sound	Norway	45.3738
	90009130	103290	110175	6885	6.2491	60225	54.6630	Logic Stores	Newzealand	45.3370
	90015149	142086	212500	70414	33.1360	116172	54.6692	UniEuro	Norway	45.3308
	90021088	224350	323689	99339	30.6896	176975	54.6744	Electricalslytical	United Kin...	45.3256
	90017050	85272	114688	29416	25.6487	62760	54.7224	Electricalsara ...	Portugal	45.2776
	70013125	101658	123428	21770	17.6378	67546	54.7250	Atliq Exclusive	Italy	45.2750
	90021094	208512	301367	92855	30.8113	165043	54.7648	Coolblue	United Kin...	45.2352
	70009134	103747	110791	7044	6.3579	60726	54.8113	Atliq e Store	Newzealand	45.1887
	90013118	101847	126289	24442	19.3540	69242	54.8282	Fnac-Darty	Italy	45.1718
	70017060	89925	120744	30819	25.5242	66285	54.8971	Atliq e Store	Portugal	45.1029

KEY QUERIES AND FORECAST ANALYSIS

The supply chain business manager wants to see which customers' forecast accuracy has dropped from 2020 to 2021. Provide a complete report with these columns: customer_code, customer_name, market, forecast_accuracy_2020, forecast_accuracy_2021

Step 1: Creating the 2020 Forecast Error Table

This query creates a temporary table to calculate forecast errors for the year 2020.

Purpose: This step calculates various forecast error metrics for each customer based on their 2020 data.

```
CREATE TEMPORARY TABLE forecast_err_table_2020 AS
SELECT
    cust_code AS customer_code,
    SUM(sold_qty) AS total_sold_qty,
    SUM(forecast_qty) AS total_forecast_qty,
    SUM(forecast_qty - sold_qty) AS net_err,
    SUM(forecast_qty - sold_qty) * 100 / SUM(forecast_qty) AS net_err_pct,
    SUM(ABS(forecast_qty - sold_qty)) AS abs_err,
    SUM(ABS(forecast_qty - sold_qty)) * 100 / SUM(forecast_qty) AS abs_err_pct
FROM fact_actuals_est
WHERE fiscal_year = 2020
GROUP BY customer_code;
```

KEY QUERIES AND FORECAST ANALYSIS

- **Step 2: Compare 2020 and 2021 Forecast Accuracy**
- In this query, a CTE (cte_2021) is used to calculate forecast accuracy for 2021, and the results are joined with the 2020 forecast accuracy from the previously created table (forecast_err_table_2020).

- **CTE for 2021 (cte_2021):**
 - This part of the query calculates the forecast accuracy for 2021. It retrieves the customer and market data from the dim_customer table, along with forecast accuracy (forecast_accuracy_2021), calculated as $100 - \text{abs_err_pct}$ (capped at 0 if error exceeds 100%).
- **Comparison with 2020:**
 - The main query compares the forecast accuracy between 2020 and 2021 for each customer.
 - The inner query retrieves forecast accuracy for 2020 from the forecast_err_table_2020 and joins it with the 2021 data using the customer_code.
 - The WHERE clause filters results to show customers where forecast accuracy in 2021 is worse than in 2020 ($\text{forecast_accuracy_2020} > \text{forecast_accuracy_2021}$).
- **Metrics:**
 - $\text{forecast_accuracy_2020}$: Forecast accuracy in 2020, calculated as $100 - \text{abs_err_pct}$.
 - $\text{forecast_accuracy_2021}$: Forecast accuracy in 2021, calculated similarly.
 - Customers where forecast accuracy dropped in 2021 are highlighted.

```
WITH cte_2021 AS (
    SELECT
        e.*,
        c.customer,
        c.market,
        IF(abs_err_pct > 100, 0, 100 - abs_err_pct) AS forecast_accuracy_2021
    FROM forecast_err_table e
    JOIN dim_customer c USING (customer_code)
)

SELECT
    t.customer_code,
    t.customer,
    t.market,
    c.forecast_accuracy_2020,
    t.forecast_accuracy_2021
FROM cte_2021 AS t
JOIN (
    SELECT
        e.*,
        c.customer,
        c.market,
        IF(abs_err_pct > 100, 0, 100 - abs_err_pct) AS forecast_accuracy_2020
    FROM forecast_err_table_2020 e
    JOIN dim_customer c USING (customer_code)
) AS c USING (customer_code)
WHERE c.forecast_accuracy_2020 > t.forecast_accuracy_2021;
```

KEY QUERIES AND FORECAST ANALYSIS

The comparison highlights customers whose forecast accuracy worsened in 2021 compared to 2020.

	customer_code	customer	market	forecast_accuracy_2020	forecast_accuracy_2021
▶	70004069	Atliq Exclusive	Japan	37.6207	32.0943
	70004070	Atliq e Store	Japan	31.2752	29.0102
	70006157	Atliq Exclusive	Philippines	35.8366	25.2009
	70006158	Atliq e Store	Philippines	42.6505	24.4904
	70007198	Atliq Exclusive	South Korea	33.2437	17.3250
	70008170	Atliq e Store	Australia	40.9573	38.7404
	70012042	Atliq Exclusive	Germany	24.2759	22.8790
	70014142	Atliq Exclusive	Netherlands	37.4290	0
	70014143	Atliq e Store	Netherlands	38.3174	0
	70016177	Atliq Exclusive	Poland	36.2176	34.8715
	80006154	Synthetic	Philippines	37.4905	24.6261
	80006155	Novus	Philippines	36.5892	25.2821
	80007195	Sage	South Korea	30.0932	19.4357
	90004061	Info Stores	Japan	34.9935	30.8528
	90004062	Flawless Stores	Japan	38.2162	32.5617
	90004063	Electricalsbea...	Japan	33.9538	30.0547
	90004065	Neptune	Japan	35.6665	30.2674
	90004066	Surface Stores	Japan	36.0465	33.5083
	90004067	Amazon	Japan	31.1697	30.5913
	90005161	Zone	Pakistan	40.0813	37.0962
	90005162	Tech	Philippines	37.4442	36.9949

CONCLUSION

In the Supply Chain Analytics section, we focused on assessing and improving the company's ability to forecast demand accurately across various customers and markets. The analysis provided valuable insights into the effectiveness of the forecasting models and highlighted areas for improvement.

Thank's For Watching
