WELCOME

ad-hoc project SQL

By: Adarsh Ranjan

AD-HOC PROJECT

An ad-hoc project is initiated to address sudden business requirements, unexpected challenges, or time-sensitive opportunities. These projects are usually one-time in nature and may not strictly follow predefined project management frameworks.

BUSINESS OVERVIEW

Atliq Hardware

Brick & Mortar

croma in-store | online | mobile

Reliancedigital

Hardware

E-Commerce

Flipkart

amazon

Hardware

OBJECTIVES

- Quickly resolve unforeseen business questions
- Provide adaptable and flexible analytical solutions
- Focus on short-term, urgent decision-making needs
- Deliver customized insights for unique scenarios
- Maintain efficiency when standard workflows are insufficient

DATABASE OVERVIEW

Dimension Tables

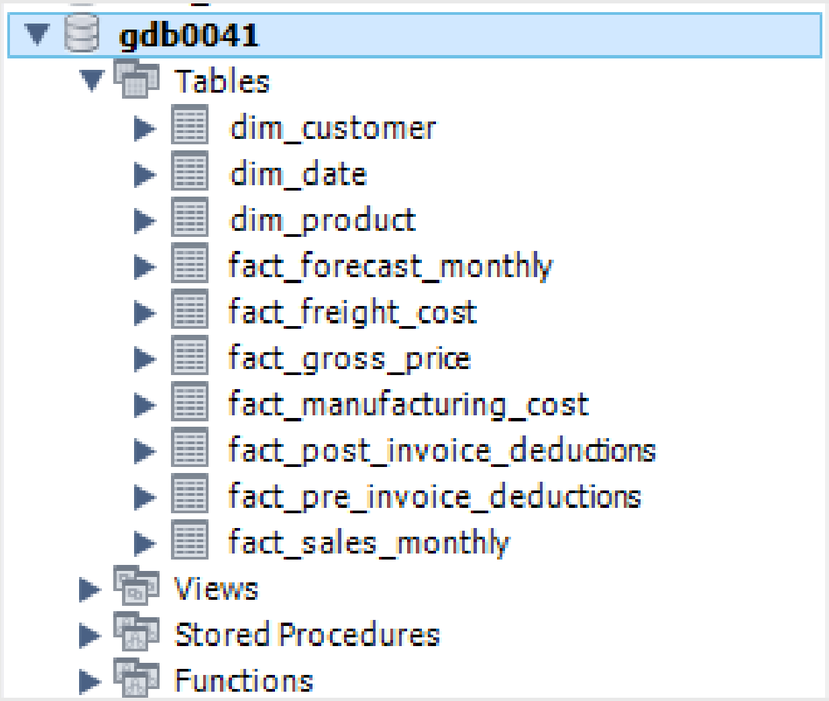
- dim_customers
- dim_products

Fact Tables

- fact_sales_monthly
- fact_forecast_monthly
- fact_freight_cost
- fact_manufacturing_cost
- fact_pre_invoice_deductions
- fact_post_invoice_deductions

DATABASE STRUCTURE

- This slide displays the database schema, highlighting tables, views, stored procedures, and functions used to support ad-hoc analysis and reporting.



AD-HOC REQUEST 1 (PRODUCT-LEVEL SALES ANALYSIS)

Business Requirement:

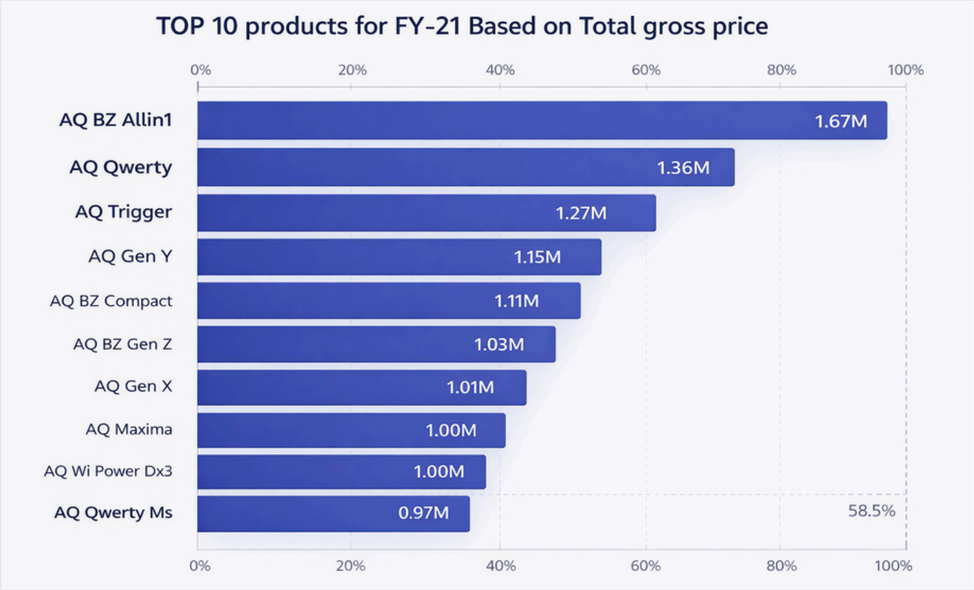
Generate a monthly aggregated sales report at the product level for **Amazon India** customers for **FY 2021** to track individual product performance.

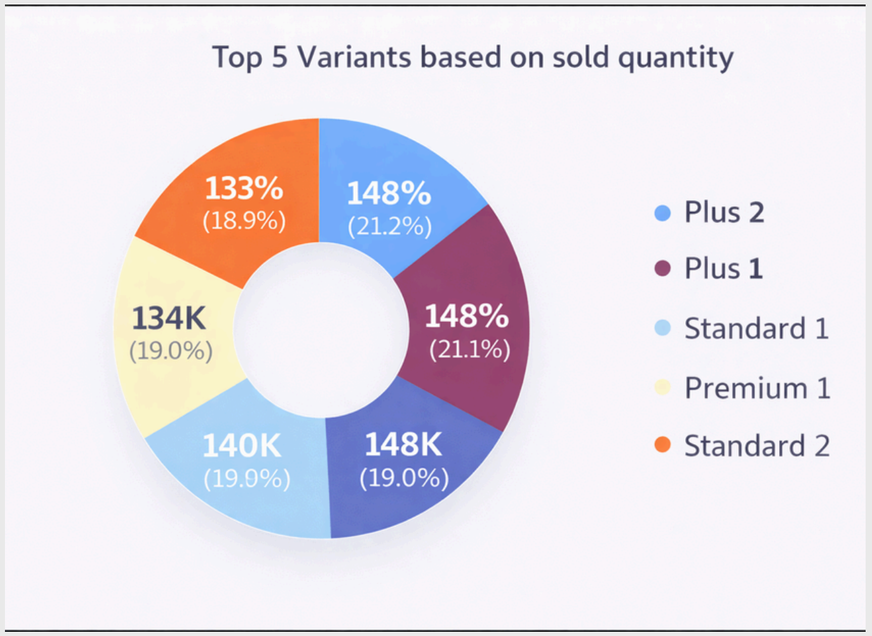
Required Fields:

- Month
- Product Name and Variant
- Quantity Sold
- Gross Price per Unit
- Total Gross Revenue

```sql
SELECT
            s.date,
            s.product_code,
            p.product,
            p.variant,
            s.sold_quantity,
            g.gross_price as gross_price_per_item,
            ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total,
            pre.pre_invoice_discount_pct
    FROM fact_sales_monthly s
    JOIN dim_product p
            ON s.product_code=p.product_code
    JOIN fact_gross_price g
            ON g.fiscal_year=get_fiscal_year(s.date)
            AND g.product_code=s.product_code
    JOIN fact_pre_invoice_deductions as pre
            ON pre.customer_code = s.customer_code AND
            pre.fiscal_year=get_fiscal_year(s.date)
    WHERE
        s.customer_code=90002008 or s.customer_code = 90002016 AND
            get_fiscal_year(s.date)=2021
    LIMIT 1000000;
```

INSIGHTS FROM AD-HOC REQUEST 1





- **Plus 2** emerged as the **top-selling** variant, contributing **21.16%** of total sales volume
- **Plus 1** followed closely with **21.08%** share
- **AQ BZ Allin1** was the **highest-selling** product with **1.67** million units
- **AQ Qwerty** ranked **second** with **1.36** million units sold

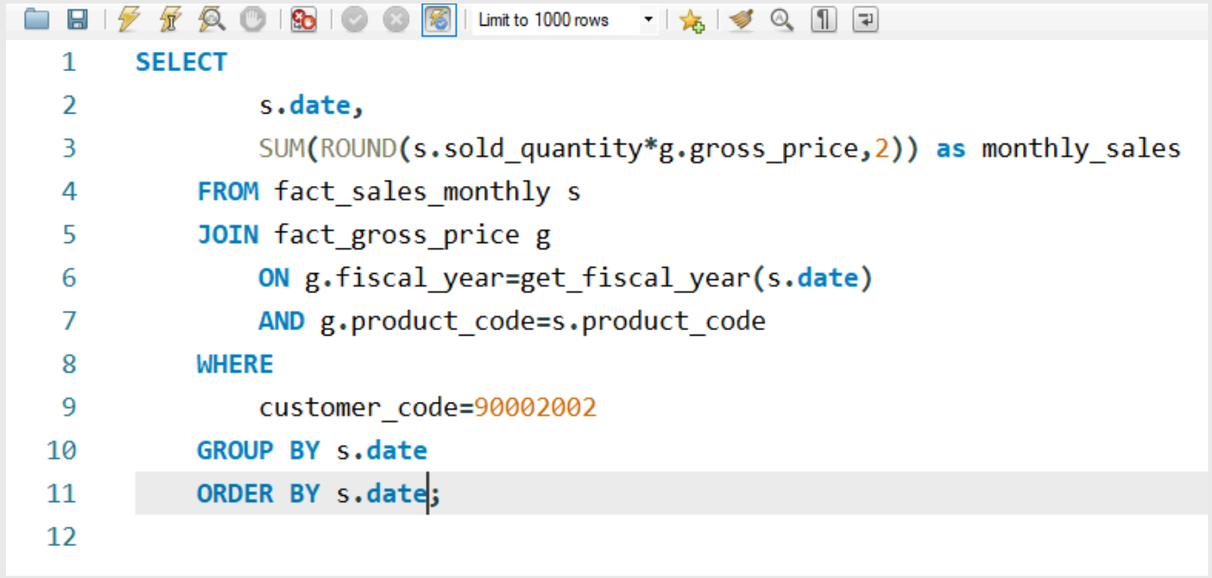
AD-HOC REQUEST 2 (CUSTOMER GROSS SALES REPORT)

Business Requirement:

Create a monthly gross sales summary for Croma India to monitor customer contribution and support relationship management.

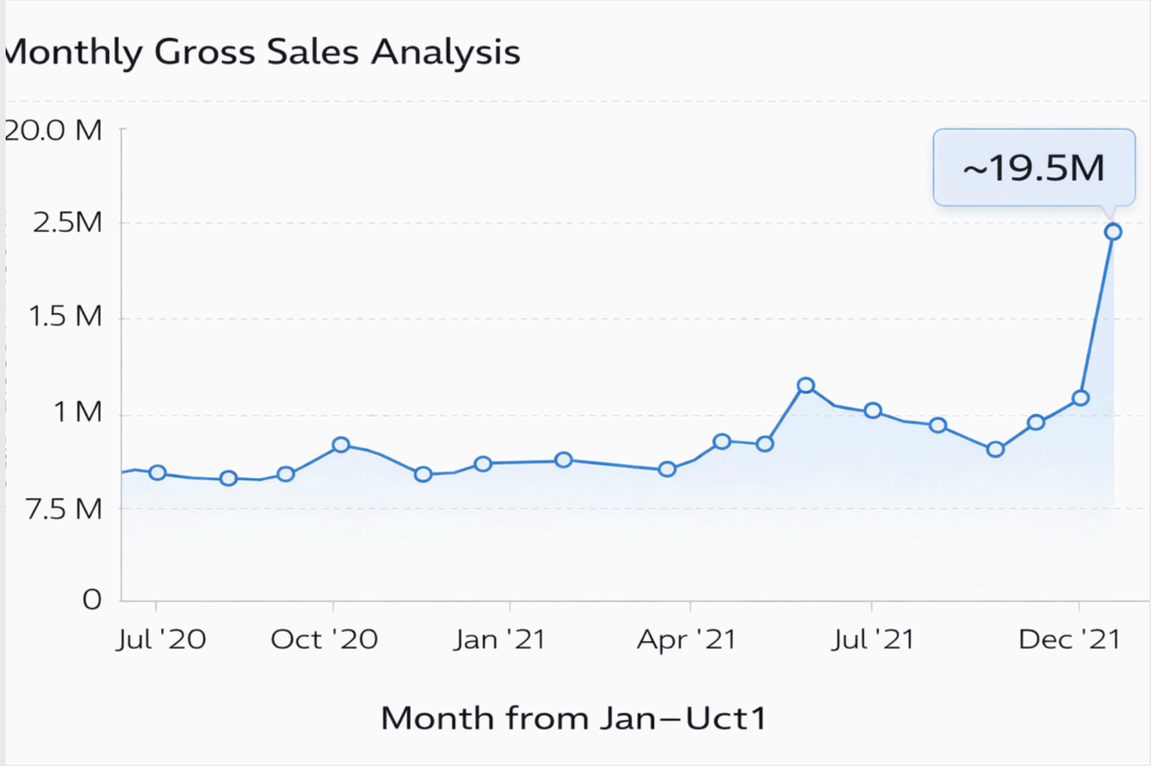
Required Fields:

- Month
- Total Gross Sales Amount



INSIGHTS FROM AD-HOC REQUEST 2

- The highest gross sales for Croma were recorded in **December 2021**, totaling approximately **19.5 million**
- This peak significantly exceeded sales from other months and years



AD-HOC REQUEST 3 (STORED PROCEDURE FOR GROSS SALES)

Objective:

Develop a reusable stored procedure to generate monthly gross sales reports, eliminating the need for manual query updates and reducing dependency on the analytics team.

Output Columns:

- Month
- Total Gross Sales for a Selected Customer

Stored Procedure

Stored procedures allow:

- Automation of recurring reports
- Improved security by restricting direct query access
- Consistent and reusable logic for business reporting

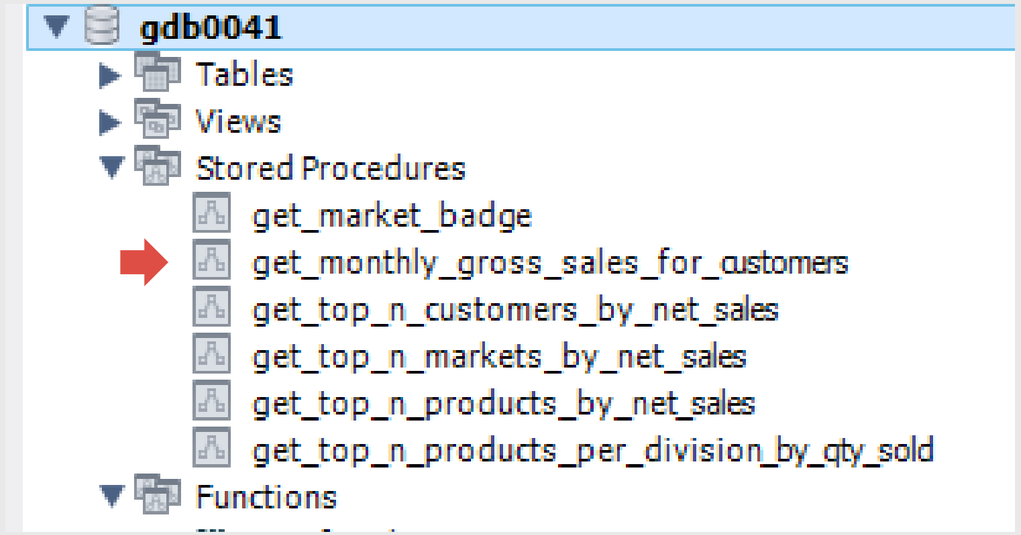
AD-HOC REQUEST 3 (STORED PROCEDURE FOR GROSS SALES)

Objective:

Develop a reusable stored procedure to generate monthly gross sales reports, eliminating the need for manual query updates and reducing dependency on the analytics team.

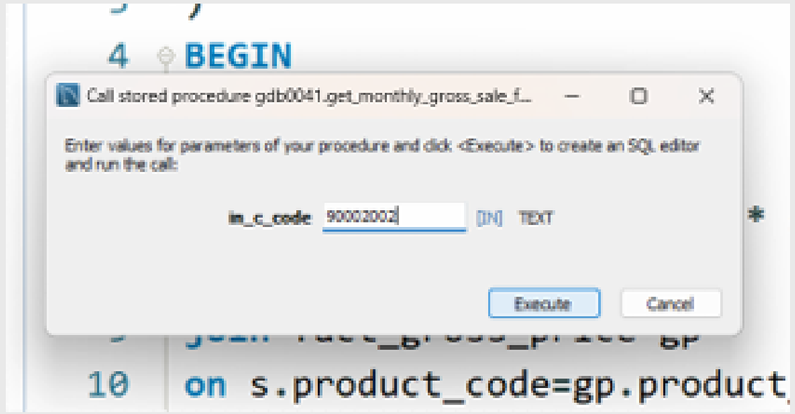
Output Columns:

- Month
- Total Gross Sales for a Selected Customer



QUERY EXECUTION & OUTPUT

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customers`(
    in_customer_code TEXT
)
BEGIN
SELECT
s.date,
Round(SUM(g.gross_price*s.sold_quantity),2) as monthly_sales
FROM fact_sales_monthly s
join fact_gross_price g
on
g.product_code=s.product_code and
g.fiscal_year= get_fiscal_year(s.date)
where
find_in_set(s.customer_code, in_customer_code)
group by s.date;
END
```



Result Grid | Filter Rows: | Export:

date	gross_total_price
2017-09-01	122407.56
2017-10-01	162687.57
2017-12-01	245673.80
2018-01-01	127574.74
2018-02-01	144799.52
2018-04-01	130643.90

AD-HOC REQUEST 4 (MARKET BADGE CLASSIFICATION)

Logic Definition:

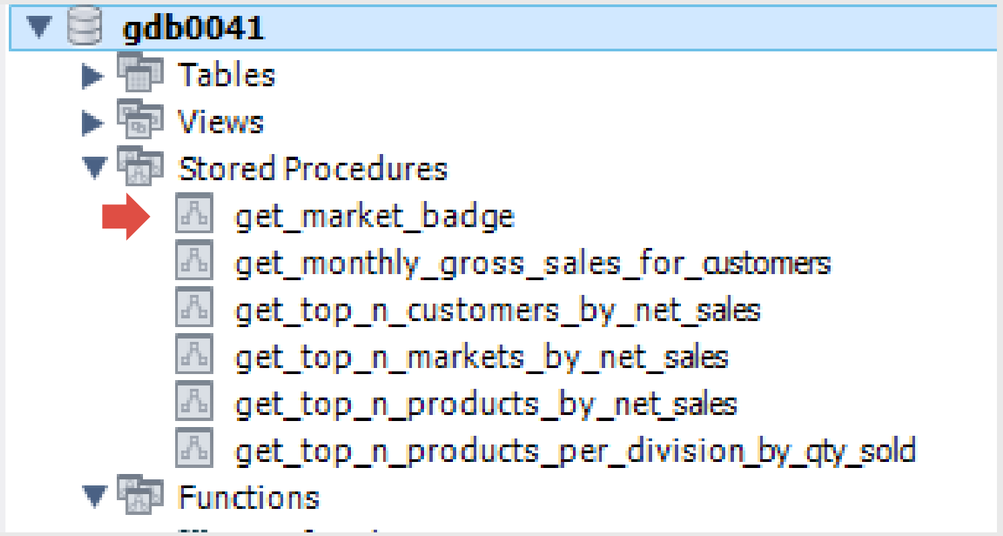
- If total sold quantity > 5 million → Gold Market
- Otherwise → Silver Market

Inputs:

- Market
- Fiscal Year

Output:

- Market Badge



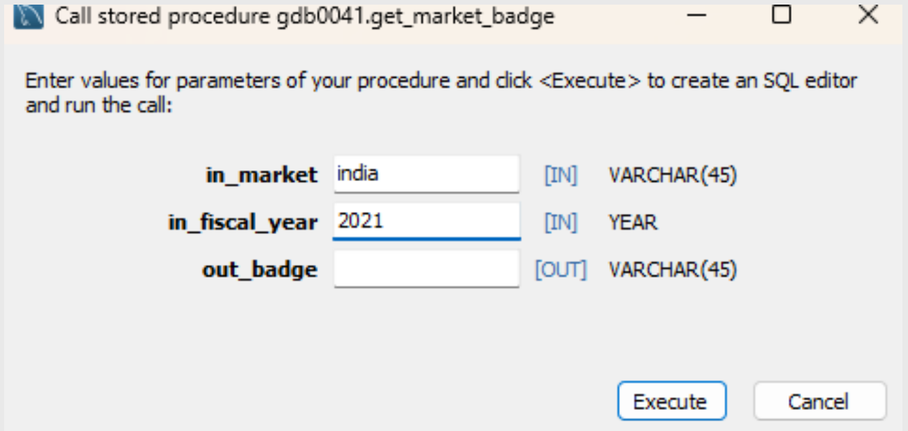
QUERY EXECUTION & OUTPUT

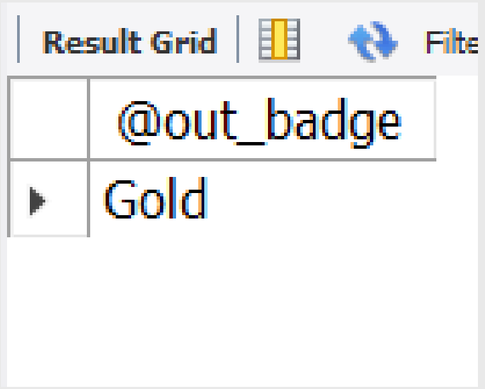
```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
            IN in_market VARCHAR(45),
            IN in_fiscal_year YEAR,
            OUT out_level VARCHAR(45)
    )
BEGIN
             DECLARE qty INT DEFAULT 0;

             # Default market is India
             IF in_market = "" THEN
                  SET in_market="India";
             END IF;

             # Retrieve total sold quantity for a given market in a given year
             SELECT
                  SUM(s.sold_quantity) INTO qty
             FROM fact_sales_monthly s
             JOIN dim_customer c
             ON s.customer_code=c.customer_code
             WHERE
                  get_fiscal_year(s.date)=in_fiscal_year AND
                  c.market=in_market;

             # Determine Gold vs Silver status
             IF qty > 5000000 THEN
                  SET out_level = 'Gold';
             ELSE
                  SET out_level = 'Silver';
             END IF;
```





AD-HOC REQUEST 5 (TOP ENTITIES ANALYSIS)

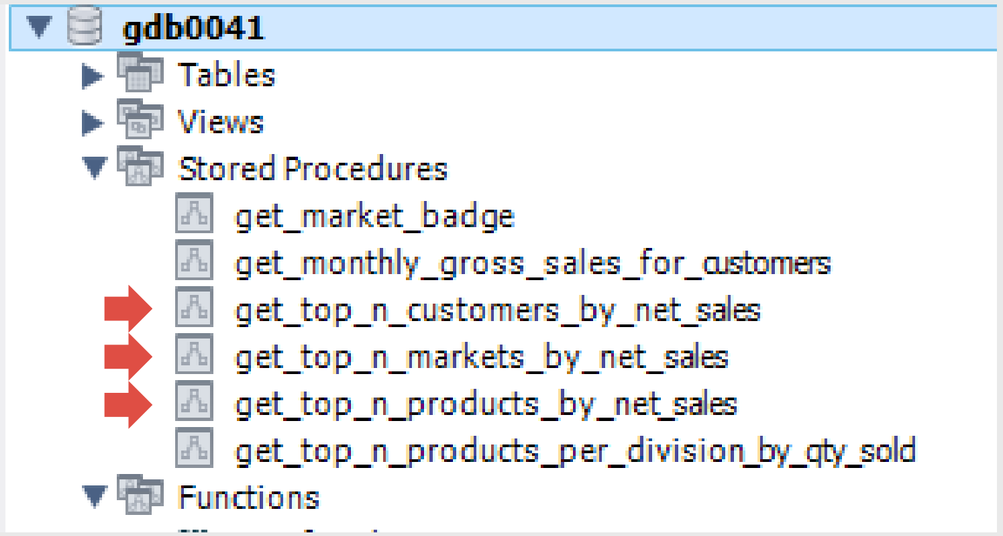
Objective:

Create stored procedures to identify:

- Top Customers
- Top Products
- Top Markets

Output Fields:

- Entity Name (Customer / Product / Market)
- Net Sales (in Millions)

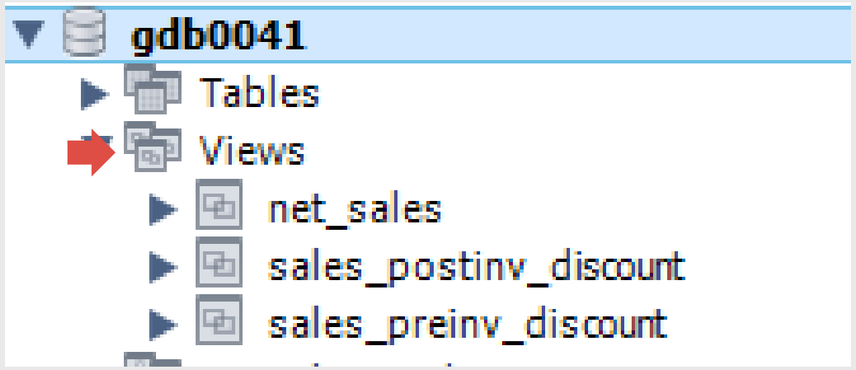


AD-HOC REQUEST 5 (TOP ENTITIES ANALYSIS)

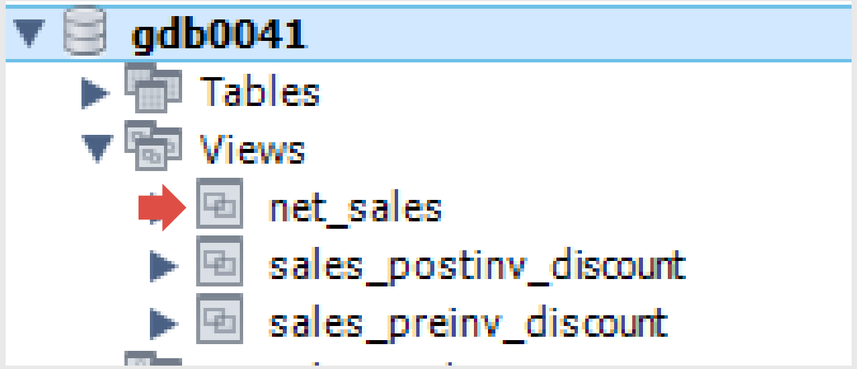
Views Concept

A view is a virtual table created from a SQL query. It:

- Does not store data physically
- Simplifies complex queries
- Enhances readability and reusability



AD-HOC REQUEST 5 (TOP ENTITIES ANALYSIS)

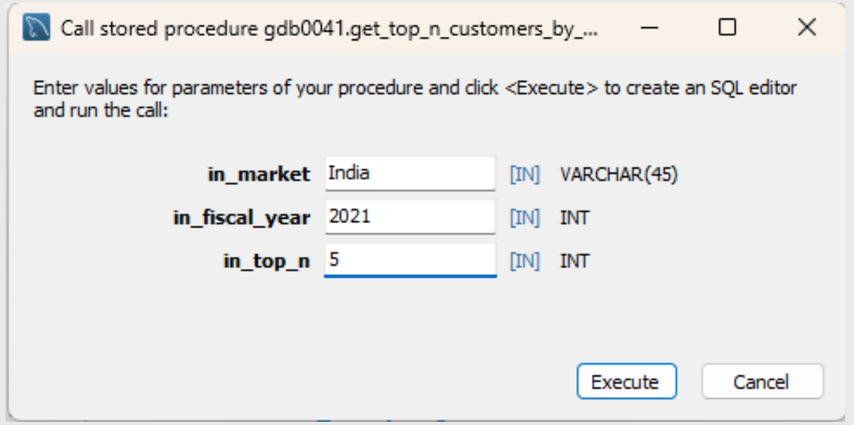


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct	net_sales
Standard	344	5295.95	0.2231	4114.423555	0.4628	2210.2683337460
Standard	274	4218.28	0.2990	2957.014280	0.3430	1942.7583819600
Standard	192	2955.88	0.2231	2296.423172	0.4423	1280.7152030244
Standard	149	2293.88	0.2231	1782.115372	0.3591	1142.1577419148
Standard	169	2601.79	0.2990	1823.854790	0.3881	1116.0167460010
Standard	136	2093.75	0.2156	1642.337500	0.3539	1061.1142587500
Standard	158	2281.39	0.2483	1714.920863	0.4016	1026.2086444192
Standard	140	2155.33	0.2231	1674.475877	0.3895	1022.2675229085
Standard	138	2124.54	0.2231	1650.555126	0.3983	993.1390193142
Standard	122	1878.21	0.2385	1430.256915	0.3294	959.1302871990
Standard	131	2016.77	0.2005	1612.407615	0.4128	946.8057515280
Standard	122	1878.21	0.2385	1430.256915	0.3637	910.0724750145
Standard	120	1847.42	0.2103	1458.907574	0.3818	901.8966622468
Standard	116	1785.84	0.2860	1275.089760	0.3132	875.7316471680

QUERY EXECUTION & OUTPUT

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_customers_by_net_sales`(
            in_market VARCHAR(45),
            in_fiscal_year INT,
            in_top_n INT
    )
BEGIN
            select
                    customer,
                    round(sum(net_sales)/1000000,2) as net_sales_mln
            from net_sales s
            join dim_customer c
                on s.customer_code=c.customer_code
            where
            s.fiscal_year=in_fiscal_year
            and s.market=in_market
            group by customer
            order by net_sales_mln desc
            limit in_top_n;
    END
```



Result Grid | Filter Rows:

customer	net_sales_mln
Amazon	30.00
Atliq Exclusive	23.98
Flipkart	12.96
Electricalsocity	12.31
Propel	11.86

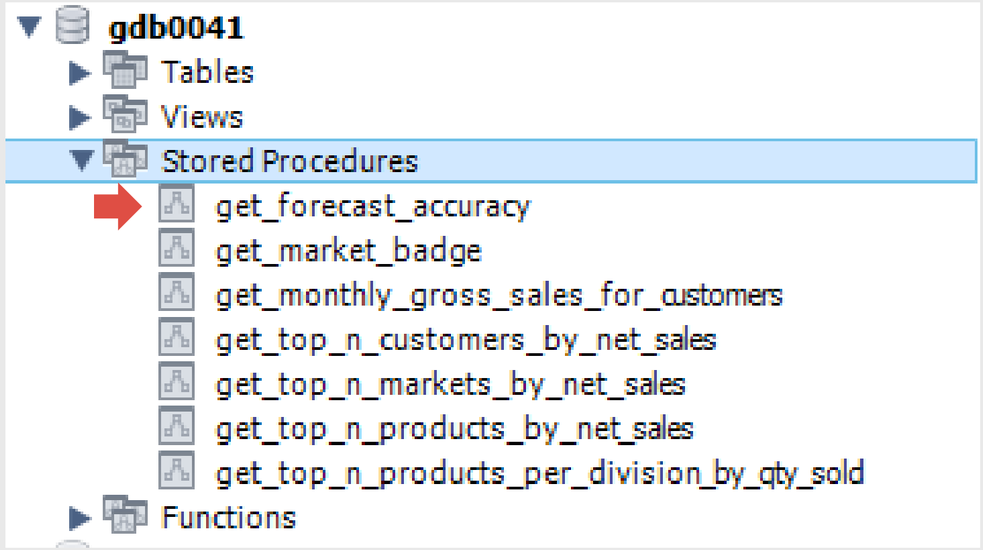
AD-HOC REQUEST 6 (FORECAST ACCURACY ANALYSIS)

Business Requirement:

- Generate an aggregate forecast accuracy report for all customers for a selected fiscal year.

Required Fields:

- Customer Code, Name, Market
- Total Sold Quantity
- Total Forecast Quantity
- Net Error
- Absolute Error
- Forecast Accuracy Percentage



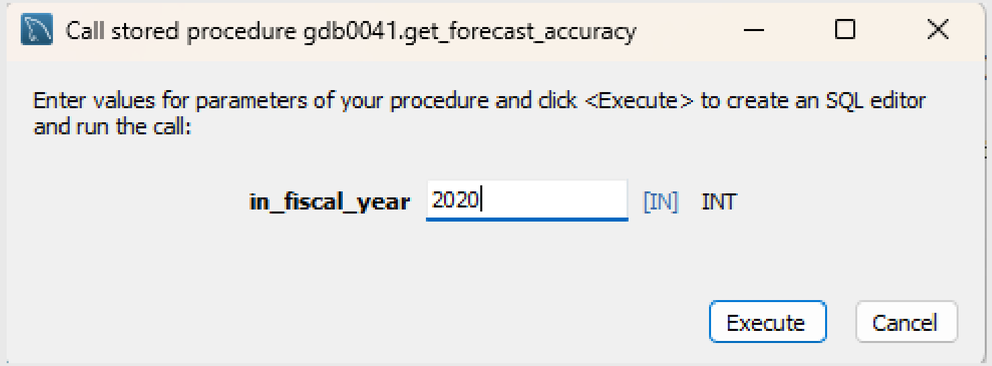
QUERY EXECUTION

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_forecast_accuracy`(
            in_fiscal_year INT
    )
BEGIN
    with forecast_err_table as (
        select
          s.customer_code as customer_code,
          c.customer as customer_name,
          c.market as market,
          sum(s.sold_quantity) as total_sold_qty,
          sum(s.forecast_quantity) as total_forecast_qty,
          sum(s.forecast_quantity-s.sold_quantity) as net_error,
          round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
          sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
          round(sum(abs(s.forecast_quantity-sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
        from fact_act_est s
        join dim_customer c
        on s.customer_code = c.customer_code
        where s.fiscal_year=in_fiscal_year
        group by customer_code
        )
        select
            *,
        if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
        from forecast_err_table
        order by forecast_accuracy desc;
    END
```

QUERY OUTPUT

Forecast Accuracy Results:

The results enable evaluation of forecasting performance and identification of gaps between predicted and actual sales.



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

customer_code	customer_name	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_accuracy
70006158	Atliq e Store	Philiphines	136991	155044	18053	11.6	88917	57.35	42.65
90010046	Amazon	Bangladesh	55532	58644	3112	5.3	33716	57.49	42.51
90023030	Amazon	Canada	127854	140248	12394	8.8	82588	58.89	41.11
70008170	Atliq e Store	Australia	178182	192586	14404	7.5	113708	59.04	40.96
90023026	Relief	Canada	85944	143041	57097	39.9	85555	59.81	40.19
90005161	Zone	Pakistan	68017	94196	26179	27.8	56441	59.92	40.08
70011194	Atliq e Store	France	106020	115390	9370	8.1	69596	60.31	39.69
90023024	Sage	Canada	83292	145672	62380	42.8	89462	61.41	38.59
90014140	Radio Popular	Netherlands	36344	62794	26450	42.1	38602	61.47	38.53
90008166	Sound	Australia	115804	200228	84424	42.2	123118	61.49	38.51
90010044	Surface Stores	Bangladesh	36764	63634	26870	42.2	39240	61.67	38.33
70014143	Atliq e Store	Netherlands	53780	60667	6887	11.4	37421	61.68	38.32
90004062	Flawless Stores	Japan	16715	22166	5451	24.6	13695	61.78	38.22
90021086	Electricalsquip...	United Kin...	72030	91645	19615	21.4	56717	61.89	38.11
90021089	Atlas Stores	United Kin...	67559	91158	23599	25.9	56423	61.90	38.10
90021094	Coolblue	United Kin...	73185	90956	17771	19.5	56449	62.06	37.94
90014137	Media Markt	Netherlands	32289	58838	26549	45.1	36565	62.15	37.85
90014138	Mbit	Netherlands	34832	59438	24606	41.4	36954	62.17	37.83
70004069	Atliq Exclusive	Japan	15449	21225	5776	27.2	13240	62.38	37.62
90014136	Reliance Digital	Netherlands	35477	62399	26922	43.1	38946	62.41	37.59

RESULTS (PROJECT OUTCOMES)

- Successfully delivered multiple business-critical ad-hoc reports across sales, customers, markets, and forecasting.
- Built reusable stored procedures that automated monthly reporting and reduced manual query dependency (as shown in stored procedure implementation on pages 10–12).
- Identified top-performing products and variants, enabling product-level performance tracking (see insights on page 7).
- Discovered peak revenue trends, such as Croma India's highest monthly gross sales in Dec 2021 (~19.5M), supporting customer performance evaluation (page 9 insights).
- Implemented market classification logic (Gold/Silver) based on sales thresholds to support strategic market segmentation (page 13).
- Generated Top Customers, Products, and Markets analysis, helping identify major revenue contributors (pages 15–18).
- Developed a forecast accuracy reporting system to measure prediction vs. actual sales performance and highlight gaps (pages 19–21).

CONCLUSION

This project demonstrates how SQL-based ad-hoc analysis can directly support business decision-making in a fast-paced environment.

Through structured use of:

- Complex joins & aggregations
- Stored procedures
- Views
- Business logic implementation

the solution transformed raw transactional data into actionable business insights.

BUSINESS VALUE DELIVERED

- Reduced reporting effort through automation
- Improved data accessibility and security
- Enabled faster response to urgent business questions
- Supported product, customer, and market performance tracking
- Strengthened forecasting evaluation and accuracy monitoring

FINAL TAKEAWAY

- This project highlights the importance of combining technical SQL expertise with business understanding to solve real-world problems efficiently.

THANK YOU