

- 1) Create a stored procedure for top 10 markets by net sales.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_markets_by_net_sales`(  
    IN fiscal_year INT,  
    IN top_n INT  
)  
BEGIN  
    SELECT  
        market,  
        ROUND(sum(net_sales)/1000000,2) as net_sales_mln  
    FROM net_sales  
    WHERE fiscal_year = fiscal_year  
    GROUP BY market  
    ORDER BY net_sales_mln DESC  
    LIMIT top_n;  
END
```

- 2) Create a stored procedure for top 10 products by net sales.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_products_by_net-sales`(  
    IN fiscal_year INT,  
    IN market VARCHAR(45),  
    IN top_n INT  
)  
BEGIN  
    SELECT  
        product,  
        ROUND(SUM(net_sales),2) as net_sales_mln  
    FROM net_sales  
    WHERE fiscal_year = fiscal_year AND market = market  
    GROUP BY product  
    LIMIT top_n;  
END
```

3) Create a stored procedure for top 10 customer by net sales.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_customers_by_net_sales`(  
    IN fiscal_year INT,  
    IN top_n INT,  
    IN market VARCHAR(40)  
)  
BEGIN  
    SELECT  
        c.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 fiscal_year = fiscal_year AND s.market = market  
    GROUP BY customer  
    ORDER BY net_sales_mln DESC  
    LIMIT top_n;  
END
```

4) Create a stored procedure for market share by net sales.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `market_share_by_net_sales`(  
    IN fiscal_year INT,  
    IN top_n INT  
)  
BEGIN  
    WITH abc as (  
        SELECT
```

```

        c.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 fiscal_year = fiscal_year
GROUP BY customer
)
SELECT
    customer,
    net_sales_mln*100/sum(net_sales_mln) over() as pct
FROM abc
ORDER BY pct DESC
LIMIT top_n;
END

```

5) Create a stored procedure for market badge.

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
    IN market VARCHAR(45),
    IN fiscal_year YEAR,
    OUT badge VARCHAR(10)
)
BEGIN
    DECLARE qty INT DEFAULT 0;

    # set default market to be india

```

```
IF market = "" THEN  
SET market = "india";  
END IF;
```

retriving total quantity for market+fiscal 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) = fiscal_year AND c.market = market  
GROUP BY c.market;
```

Determine market badge

```
IF qty > 5000000 then  
SET badge = "Gold";  
ELSE  
SET badge = "Silver";  
END IF;  
END
```

6) Create a stored procedure for forecast accuracy.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `forecast_accuracy`(  
    IN fiscal_year INT  
)  
BEGIN  
    WITH forecast_acc_table as (  
        SELECT  
            customer_code,  
            SUM(sold_quantity) as sold_quantity,  
            SUM(forecast_quantity) as forecast_quantity,  
            SUM(forecast_quantity - sold_quantity) as net_error,  
            SUM(forecast_quantity - sold_quantity)*100/SUM(forecast_quantity) as net_error_pct,  
            ABS(SUM(forecast_quantity - sold_quantity)) as abs_error,  
            ABS(SUM(forecast_quantity -  
sold_quantity))*100/SUM(forecast_quantity) as abs_error_pct  
        FROM gdb0041.fact_act_est  
        WHERE fiscal_year = fiscal_year  
        GROUP BY customer_code  
    )  
    SELECT  
        a.customer_code, c.customer,  
        c.market,  
        a.sold_quantity, a.forecast_quantity,  
        a.net_error, a.net_error_pct,  
        a.abs_error, a.abs_error_pct,  
        IF(abs_error_pct > 100, 0, (100 - abs_error_pct)) as forecast_accuracy  
    FROM forecast_acc_table a  
    JOIN dim_customer c  
    USING (customer_code)
```

ORDER BY forecast_accuracy **DESC**;

END

7) Create a stored procedure for monthly gross sales for customers.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`get_monthly_gross_sales_for_customer`(
in_customer_code TEXT
)
BEGIN
SELECT
        s.date,
        ROUND(SUM(g.gross_price*s.sold_quantity),2) as gross_price_total
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) >0
GROUP BY s.date;
END
```