



Finance Analytics

Gross Sales Report 1: Monthly Product Transactions for Croma Customer in FY 21

Query

```
30 • SELECT
31     s.date,
32     s.product_code,
33     p.product,
34     p.variant,
35     s.sold_quantity,
36     g.gross_price,
37     ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total
38 FROM fact_sales_monthly s
39 JOIN dim_product p
40     ON s.product_code=p.product_code
41 JOIN fact_gross_price g
42     ON g.fiscal_year=get_fiscal_year(s.date)
43     AND g.product_code=s.product_code
44 WHERE
45     customer_code=90002002 AND
46     get_fiscal_year(s.date)=2021
47 LIMIT 1000000;
```

Result

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.6987	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107	24.7312	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123	23.6154	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.7223	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.1027	6396.24
2020-09-01	A0321150303	AQ Zion Saga	Premium	137	28.0059	3836.81
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	23	19.5235	449.04
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	82	19.9239	1633.76
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	86	20.0766	1726.59
2020-09-01	A0418150106	AQ Mforce Gen X	Plus 3	48	19.9365	956.95
2020-09-01	A0519150201	AQ Mforce Gen Y	Standard 1	138	22.3984	3090.98

Gross Sales Report 2: Monthly Product Transactions for all FY

Query and Result

```
43  -- Generate monthly gross sales report for Croma India for all the years
44  •  SELECT
45      s.date,
46      SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
47  FROM fact_sales_monthly s
48  JOIN fact_gross_price g
49      ON g.fiscal_year=get_fy(s.date) AND g.product_code=s.product_code
50  WHERE
51      customer_code=90002002
52  GROUP BY date;
```

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

date	monthly_sales
2021-04-01	2253575.01
2021-05-01	2181587.87
2021-06-01	2288587.49
2021-08-01	2349478.81
2021-09-01	11192823.18
2021-10-01	13908229.35
2021-12-01	19537146.58

Generate monthly gross sales report for any customer using stored procedure

Query for Stored Procedure

```
60 DELIMITER $
61 • CREATE PROCEDURE `get_monthly_gross_sales_for_customer` (
62     in_customer_codes TEXT
63 )
64 BEGIN
65     SELECT
66         s.date,
67         SUM(ROUND(s.sold_quantity * g.gross_price, 2)) AS monthly_sales
68     FROM fact_sales_monthly s
69     JOIN fact_gross_price g
70         ON g.fiscal_year = get_fy(s.date)
71         AND g.product_code = s.product_code
72     WHERE FIND_IN_SET(s.customer_code, in_customer_codes) > 0
73     GROUP BY s.date
74     ORDER BY s.date DESC;
75 END$
76 DELIMITER ;
77 • CALL `get_monthly_gross_sales_for_customer` ("90002002");
```

Result

	date	gross_price_total
▶	2017-09-01	122407.5582
	2017-10-01	162687.5716
	2017-12-01	245673.8042
	2018-01-01	127574.7372
	2018-02-01	144799.5182
	2018-04-01	130643.8976
	2018-05-01	139165.0975
	2018-06-01	125735.3786
	2018-08-01	125409.8801
	2018-09-01	343337.1651

Stored Procedure for retrieving market badge

Market Badge logic : If **total sold quantity > 5 million** that market is considered "**Gold**" else "**Silver**"

Query and Result

```
63 DELIMITER $
64 CREATE PROCEDURE `get_market_badge` (
65     IN in_market VARCHAR(45),
66     IN in_fiscal_year YEAR,
67     OUT out_level VARCHAR(45)
68 )
69 BEGIN
70     DECLARE qty INT DEFAULT 0;
71     # Default market is India
72     IF in_market = "" THEN
73         SET in_market="India";
74     END IF;
75     # Retrieve total sold quantity for a given market in a given year
76     SELECT
77         SUM(s.sold_quantity) INTO qty
78     FROM fact_sales_monthly s
79     JOIN dim_customer c
80     ON s.customer code=c.customer code
```

```
81 WHERE
82     get_fiscal_year(s.date)=in_fiscal_year AND
83     c.market=in_market;
84 # Determine Gold vs Silver status
85 IF qty > 5000000 THEN
86     SET out_level = 'Gold';
87 ELSE
88     SET out_level = 'Silver';
89 END IF;
90 END$
91 DELIMITER ;
```

```
1 • set @out_badge = '0';
2 • call gdb0041.get_market_badge('India', 2021, @out_badge);
3 • select @out_badge;
```

4

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

@out_badge

Gold

A large blue circle containing the text "Supply Chain Analytics". To the left of the circle, there is a series of five teal-colored line segments of varying lengths, arranged in a curved, dashed pattern. At the bottom right of the blue circle, there is a small purple circle.

Supply Chain Analytics

Creating table fact_actuals_est which contains sold_qty and forecast qty

```
2 • CREATE TABLE fact_actuals_est AS
3 (
4     select
5         s.date as date,
6         s.fiscal_year as fiscal_year,
7         s.product_code as product_code,
8         s.customer_code as customer_code,
9         s.sold_quantity as sold_quantity,
10        f.forecast_quantity as forecast_quantity
11    from
12        fact_sales_monthly s
13    left join fact_forecast_monthly f
14    using (date, customer_code, product_code)
15    UNION
16    select
17        f.date as date,
18        f.fiscal_year as fiscal_year,
19        f.product_code as product_code,
20        f.customer_code as customer_code,
21        s.sold_quantity as sold_quantity,
22        f.forecast_quantity as forecast_quantity
23    from
24        fact_forecast_monthly f
25    left join fact_sales_monthly s
26    using (date, customer_code, product_code)
27 );
28 • SELECT * from fact_actuals_est;
```

Creating table fact_actuals_est which contains sold_qty and forecast qty

Result

date	fiscal_year	product_code	customer_code	sold_quantity	forecast_quantity
2017-09-01	2018	A0118150101	70002017	51	18
2017-09-01	2018	A0118150101	70002018	77	11
2017-09-01	2018	A0118150101	70003181	17	9
2017-09-01	2018	A0118150101	70003182	6	6
2017-09-01	2018	A0118150101	70006157	5	5
2017-09-01	2018	A0118150101	70006158	7	6
2017-09-01	2018	A0118150101	70007198	29	4
2017-09-01	2018	A0118150101	70007199	34	7
2017-09-01	2018	A0118150101	70008169	22	7
2017-09-01	2018	A0118150101	70008170	5	8
2017-09-01	2018	A0118150101	70011193	10	5
2017-09-01	2018	A0118150101	70011194	4	7
2017-09-01	2018	A0118150101	70012042	0	0
2017-09-01	2018	A0118150101	70012043	0	0
2017-09-01	2018	A0118150101	70013125	1	2

Forecast Accuracy Report

```
29  #Forecast accuracy report
30 •  SET SQL_MODE="";
31 •  WITH forecast_err_table AS
32  (
33      SELECT
34          s.customer_code,
35          SUM(sold_quantity) AS total_sold_quantity,
36          SUM(forecast_quantity) AS total_forecast_quantity,
37          SUM((forecast_quantity - sold_quantity)) AS net_err,
38          SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity) AS net_err_pct,
39          SUM(ABS(forecast_quantity - sold_quantity)) AS abs_err,
40          SUM(ABS(forecast_quantity - sold_quantity))*100/SUM(forecast_quantity) AS abs_err_pct
41      FROM
42          fact_actuals_est s
43      WHERE s.fiscal_year = 2021
44      GROUP BY customer_code
```

Forecast Accuracy Report

```
45  )  
46  
47  SELECT  
48      e.*,  
49      c.customer,  
50      c.market,  
51      IF(abs_err_pct > 100, 0, 100-abs_err_pct) AS forecast_accuracy  
52  FROM forecast_err_table e  
53  JOIN dim_customer c  
54  ON e.customer_code = c.customer_code  
55  ORDER BY forecast_accuracy DESC  
56  ;
```

Forecast Accuracy Report

Result

customer_code	total_sold_quantity	total_forecast_quantity	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

Forecast Accuracy 2020 vs 2021

```
60 • SET SQL_MODE="";
61 • DROP TABLE IF EXISTS forecast_accuracy_2021;
62 • CREATE TEMPORARY TABLE forecast_accuracy_2021
63   WITH forecast_err_table AS
64   (
65     SELECT
66       s.customer_code,
67       c.customer,
68       c.market,
69       SUM(sold_quantity) AS total_sold_quantity,
70       SUM(forecast_quantity) AS total_forecast_quantity,
71       SUM((forecast_quantity - sold_quantity)) AS net_err,
72       ROUND(SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS net_err_pct,
73       SUM(ABS(forecast_quantity - sold_quantity)) AS abs_err,
74       ROUND(SUM(ABS(forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS abs_err_pct
75   FROM
76   fact_actuals_est s
```

Forecast Accuracy 2020 vs 2021

```
60 • SET SQL_MODE="";
61 • DROP TABLE IF EXISTS forecast_accuracy_2021;
62 • CREATE TEMPORARY TABLE forecast_accuracy_2021
63   WITH forecast_err_table AS
64   (
65     SELECT
66       s.customer_code,
67       c.customer,
68       c.market,
69       SUM(sold_quantity) AS total_sold_quantity,
70       SUM(forecast_quantity) AS total_forecast_quantity,
71       SUM((forecast_quantity - sold_quantity)) AS net_err,
72       ROUND(SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS net_err_pct,
73       SUM(ABS(forecast_quantity - sold_quantity)) AS abs_err,
74       ROUND(SUM(ABS(forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS abs_err_pct
75   FROM
76   fact_actuals_est s
```

Forecast Accuracy 2020 vs 2021

```
77     JOIN dim_customer c
78     ON s.customer_code = c.customer_code
79     WHERE s.fiscal_year = 2021
80     GROUP BY customer_code
81 )
82 SELECT *,
83     IF(abs_err_pct > 100, 0, 100-abs_err_pct) AS forecast_accuracy
84 FROM forecast_err_table
85 ORDER BY forecast_accuracy DESC;
86 • DROP TABLE IF EXISTS forecast_accuracy_2020;
87 • CREATE TEMPORARY TABLE forecast_accuracy_2020
88 WITH forecast_err_table AS
89 (
90     SELECT
91         s.customer_code,
92         c.customer,
93         c.market,
```


Forecast Accuracy 2020 vs 2021

```
94      SUM(sold_quantity) AS total_sold_quantity,  
95      SUM(forecast_quantity) AS total_forecast_quantity,  
96      SUM((forecast_quantity - sold_quantity)) AS net_err,  
97      ROUND(SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS net_err_pct,  
98      SUM(ABS(forecast_quantity - sold_quantity)) AS abs_err,  
99      ROUND(SUM(ABS(forecast_quantity - sold_quantity))*100/SUM(forecast_quantity),2) AS abs_err_pct  
100  FROM  
101  fact_actuals_est s  
102  JOIN dim_customer c  
103  ON s.customer_code = c.customer_code  
104  WHERE s.fiscal_year = 2020  
105  GROUP BY customer_code  
106  )  
107  SELECT *,  
108      IF(abs_err_pct > 100, 0,100-abs_err_pct) AS forecast_accuracy  
109  FROM forecast_err_table  
110  ORDER BY forecast_accuracy DESC;
```

Forecast Accuracy 2020 vs 2021

```
111
112 • SELECT
113     f_2020.customer_code,
114     f_2020.customer,
115     f_2020.market,
116     f_2020.forecast_accuracy AS forecast_accuracy_2020,
117     f_2021.forecast_accuracy AS forecast_accuracy_2021
118 FROM forecast_accuracy_2020 f_2020
119 JOIN forecast_accuracy_2021 f_2021
120 ON f_2020.customer_code = f_2021.customer_code
121 WHERE f_2021.forecast_accuracy < f_2020.forecast_accuracy
122 ORDER BY f_2020.forecast_accuracy DESC
123 ;
```

Result

customer_code	customer	market	forecast_accuracy_2020	forecast_accuracy_2021
70006158	Atliq e Store	Philippines	42.65	24.49
70008170	Atliq e Store	Australia	40.96	38.74
90005161	Zone	Pakistan	40.08	37.10
90014140	Radio Popular	Netherlands	38.53	0.00
90008166	Sound	Australia	38.51	36.79
70014143	Atliq e Store	Netherlands	38.32	0.00
90004062	Flawless Stores	Japan	38.22	32.56
90014137	Media Markt	Netherlands	37.85	0.00
90014138	Mbit	Netherlands	37.83	0.00
70004069	Atliq Exclusive	Japan	37.62	32.09



Thank You!