



FINANCE & SUPPLY CHAIN ANALYTIC USING SQL



AtliQ Hardware

Gowhar Ahmad Dar

FINANCE ANALYTICS



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

Query

```
1  ### Gross Sales Report: Monthly Product Transactions
2  •  SELECT
3      s.date,
4      s.product_code,
5      p.product,
6      p.variant,
7      s.sold_quantity,
8      g.gross_price,
9      ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total
10 FROM fact_sales_monthly s
11 JOIN dim_product p
12     ON s.product_code=p.product_code
13 JOIN fact_gross_price g
14     ON g.fiscal_year=get_fiscal_year(s.date)
15     AND g.product_code=s.product_code
16 WHERE
17     customer_code=90002002 AND
18     get_fiscal_year(s.date)=2021
19 LIMIT 1000000;
```

Result

Result Grid							
Filter Rows:		Export:		Wrap Cell Content:		Fetch rows:	
	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
	2020-10-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	95	19.0573	1810.44
	2020-12-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	113	19.0573	2153.47
	2021-01-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	182	19.0573	3468.43
	2021-02-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	208	19.0573	3963.92
	2021-04-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	199	19.0573	3792.40
	2021-05-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	58	19.0573	1105.32
	2021-06-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	205	19.0573	3906.75
	2021-08-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	88	19.0573	1677.04
	2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
	2020-10-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	237	21.4565	5085.19
	2020-12-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	172	21.4565	3690.52
	2021-01-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	121	21.4565	2596.24
	2021-02-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	77	21.4565	1652.15
	2021-04-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	247	21.4565	5299.76
	2021-05-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	135	21.4565	2896.63
	2021-06-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	78	21.4565	1673.61
	2021-08-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	196	21.4565	4205.47
	2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
	2020-10-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	102	21.7795	2221.51
	2020-12-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	383	21.7795	8341.55
	2021-01-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	142	21.7795	3092.69



Gross Sales Report 2: Monthly Product Transactions for Croma Customer in FY

Query

```
1  ## monthly gross sales report for Croma India for all the years
2  SELECT
3      s.date,
4      SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
5  FROM fact_sales_monthly s
6  JOIN fact_gross_price g
7      ON g.fiscal_year=get_fiscal_year(s.date) AND g.product_code=s.product_code
8  WHERE
9      customer_code=90002002
10 GROUP BY date;
11
```

Result

Result Grid		Filter Rows:
	date	monthly_sales
▶	2017-09-01	122407.57
	2017-10-01	162687.56
	2017-12-01	245673.84
	2018-01-01	127574.73
	2018-02-01	144799.54
	2018-04-01	130643.92
	2018-05-01	139165.06
	2018-06-01	125735.36
	2018-08-01	125409.90
	2018-09-01	343337.14
	2018-10-01	440562.10
	2018-12-01	653944.72
	2019-01-01	359025.06
	2019-02-01	356607.19
	2019-04-01	379549.74
	2019-05-01	340152.29
	2019-06-01	343792.08
	2019-08-01	338108.87
	2019-09-01	808250.42



Generate monthly gross sales report for any customer using stored procedure

Query for Store Procedure

```
1  -- Generate monthly gross sales report for any customer using stored procedure
2
3  DELIMITER $
4  CREATE PROCEDURE `get_monthly_gross_sales_for_customer`(
5      in_customer_codes TEXT
6  )
7  BEGIN
8      SELECT
9          s.date,
10         SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
11     FROM fact_sales_monthly s
12     JOIN fact_gross_price g
13         ON g.fiscal_year=get_fiscal_year(s.date)
14         AND g.product_code=s.product_code
15     WHERE
16         FIND_IN_SET(s.customer_code, in_customer_codes) > 0
17     GROUP BY s.date
18     ORDER BY s.date DESC;
19 END$
20 DELIMITER ;
21
22 call gdb0041.get_monthly_gross_sales_for_customer('90002002');
```

Result

Result Grid		Filter Rows:
	date	monthly_sales
▶	2017-09-01	122407.57
	2017-10-01	162687.56
	2017-12-01	245673.84
	2018-01-01	127574.73
	2018-02-01	144799.54
	2018-04-01	130643.92
	2018-05-01	139165.06
	2018-06-01	125735.36
	2018-08-01	125409.90
	2018-09-01	343337.14
	2018-10-01	440562.10
	2018-12-01	653944.72
	2019-01-01	359025.06
	2019-02-01	356607.19
	2019-04-01	379549.74
	2019-05-01	340152.29
	2019-06-01	343792.08
	2019-08-01	338108.87
	2019-09-01	808250.42
	2019-10-01	1092622.30
	2019-12-01	1488174.01



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

```
1  ##Write a stored proc that can retrieve market badge. i.e. if total sold quantity > 5 million that market is considered "Gold" else "Silver"
2  DELIMITER $
3  CREATE PROCEDURE `get_market_badge` (
4      IN in_market VARCHAR(45),
5      IN in_fiscal_year YEAR,
6      OUT out_level VARCHAR(45)
7  )
8  BEGIN
9      DECLARE qty INT DEFAULT 0;
10     ## Default market is India
11     IF in_market = "" THEN
12         SET in_market="India";
13     END IF;
14     ## Retrieve total sold quantity for a given market in a given year
15     SELECT
16         SUM(s.sold_quantity) INTO qty
17     FROM fact_sales_monthly s
18     JOIN dim_customer c
19     ON s.customer_code=c.customer_code
20     WHERE
21         get_fiscal_year(s.date)=in_fiscal_year AND
22         c.market=in_market;
23     ## Determine Gold vs Silver status
24     IF qty > 5000000 THEN
25         SET out_level = 'Gold';
26     ELSE
27         SET out_level = 'Silver';
28     END IF;
29     END$
```

```
7  END$
8  DELIMITER ;
9
10  set @out_level = '0';
11  call gdb0041.get_market_badge('india', 2021, @out_level);
12  select @out_level;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
@out_level			
Gold			





SUPPLY CHAIN ANALYTICS



Creating table fact_actuals_est which contains sold_qty and forecast qty

Query

```
1 • SET GLOBAL max_allowed_packet = 1073741824; -- Example to set 1GB limit
2 • SET GLOBAL wait_timeout = 28800;
3 • SET GLOBAL interactive_timeout = 28800;
4
5
6 -- Create fact_act_est table
7 • drop table if exists fact_act_est;
8
9 • create table fact_act_est
10 (
11     select
12         s.date as date,
13         s.fiscal_year as fiscal_year,
14         s.product_code as product_code,
15         s.customer_code as customer_code,
16         s.sold_quantity as sold_quantity,
17         f.forecast_quantity as forecast_quantity
18     from
19         fact_sales_monthly s
20     left join fact_forecast_monthly f
21         using (date, customer_code, product_code)
22 )
23 union
24 (
25     select
26         f.date as date,
27         f.fiscal_year as fiscal_year,
28         f.product_code as product_code,
29         f.customer code as customer code,
```

```
30         s.sold_quantity as sold_quantity,
31         f.forecast_quantity as forecast_quantity
32     from
33         fact_forecast_monthly f
34     left join fact_sales_monthly s
35         using (date, customer_code, product_code)
36 );
37
38 • SET SQL_SAFE_UPDATES = 0;
39
40 • update fact_act_est
41     set sold_quantity = 0
42     where sold_quantity is null;
43
44 • update fact_act_est
45     set forecast_quantity = 0
46     where forecast_quantity is null;
47
48 • SELECT * FROM gdb0041.fact_act_est;
49
```



Creating table fact_actuals_est which contains sold_qty and forecast qty

Result

Result Grid						
		Filter Rows:			Edit:	Export/Import:
	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
	2017-09-01	2018	A0118150101	70013126	1	2
	2017-09-01	2018	A0118150101	70016178	1	0
	2017-09-01	2018	A0118150101	70022085	20	12
	2017-09-01	2018	A0118150101	70023031	4	1
	2017-09-01	2018	A0118150101	80001019	10	7
	2017-09-01	2018	A0118150101	80006154	10	21
	2017-09-01	2018	A0118150101	80006155	28	21
	2017-09-01	2018	A0118150101	80007195	80	62
	2017-09-01	2018	A0118150101	90001021	1	2
	2017-09-01	2018	A0118150101	90002001	42	52



Forecast Accuracy Report

Query

```
SQL File 7*  SQL File 8* x
[Icons] Don't Limit

1  -- Forecast accuracy report
2  with forecast_err_table as (
3      select
4          s.customer_code as customer_code,
5          c.customer as customer_name,
6          c.market as market,
7          sum(s.sold_quantity) as total_sold_qty,
8          sum(s.forecast_quantity) as total_forecast_qty,
9          sum(s.forecast_quantity-s.sold_quantity) as net_error,
10         round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
11         sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
12         round(sum(abs(s.forecast_quantity-s.sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
13     from fact_act_est s
14     join dim_customer c
15     on s.customer_code = c.customer_code
16     where s.fiscal_year=2021
17     group by customer_code
18 )
19 select
20     *,
21     if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
22 from forecast_err_table
23     order by forecast_accuracy desc;
```



Forecast Accuracy Report

Result

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
▶	90013120	Coolblue	Italy	109547	133532	23985	18.0	70467	52.77	47.23
	70010048	Atliq e Store	Bangladesh	119439	142010	22571	15.9	75711	53.31	46.69
	90023027	Costco	Canada	236189	279962	43773	15.6	149303	53.33	46.67
	90023026	Relief	Canada	228988	273492	44504	16.3	146948	53.73	46.27
	90017051	Forward Stores	Portugal	86823	118067	31244	26.5	63568	53.84	46.16
	90017058	Mbit	Portugal	86860	110195	23335	21.2	59473	53.97	46.03
	90023028	walmart	Canada	239081	283323	44242	15.6	153058	54.02	45.98
	90023024	Sage	Canada	246397	287233	40836	14.2	155610	54.18	45.82
	90015146	Mbit	Norway	147152	210507	63355	30.1	114189	54.24	45.76
	90013124	Amazon	Italy	110898	136116	25218	18.5	73826	54.24	45.76
	90017054	Flawless Stores	Portugal	84371	114698	30327	26.4	62483	54.48	45.52
	70027208	Atliq e Store	Brazil	33713	47321	13608	28.8	25784	54.49	45.51
	90015147	Chiptec	Norway	154897	223867	68970	30.8	122100	54.54	45.46
	80001019	Neptune	China	1113979	1275248	161269	12.6	695779	54.56	45.44
	90015144	Sound	Norway	160074	225637	65563	29.1	123257	54.63	45.37
	90009130	Logic Stores	Newzealand	103290	110175	6885	6.2	60225	54.66	45.34
	90015149	UniEuro	Norway	142086	212500	70414	33.1	116172	54.67	45.33
	90021088	Electricalslytical	United Kin...	224350	323689	99339	30.7	176975	54.67	45.33
	90017050	Electricalsara S...	Portugal	85272	114688	29416	25.6	62760	54.72	45.28
	70013125	Atliq Exclusive	Italy	101658	123428	21770	17.6	67546	54.73	45.27
	90021094	Coolblue	United Kin...	208512	301367	92855	30.8	165043	54.76	45.24
	70009134	Atliq e Store	Newzealand	103747	110791	7044	6.4	60726	54.81	45.19
	90013118	Fnac-Darty	Italy	101847	126289	24442	19.4	69242	54.83	45.17
	70023031	Atliq Exclusive	Canada	234114	286297	52183	18.2	157171	54.90	45.10
	90023025	Premium Stores	Canada	220808	266351	45543	17.1	146235	54.90	45.10



Store Procedure Forecast Accuracy Report

Query

```
SQL File 7*  SQL File 8* x  get_forecast_accuracy
-- Forecast accuracy report by store procedure
DELIMITER $
CREATE 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-s.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$
```



Store Procedure Forecast Accuracy Report

Query and Result

```
27 FROM forecast_err_table
28 order by forecast_accuracy desc;
29 END$
30
31 DELIMITER ;
32
33 • call gdb0041.get_forecast_accuracy(2021);
```

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
▶	90013120	Coolblue	Italy	109547	133532	23985	18.0	70467	52.77	47.23
	70010048	Atiq e Store	Bangladesh	119439	142010	22571	15.9	75711	53.31	46.69
	90023027	Costco	Canada	236189	279962	43773	15.6	149303	53.33	46.67
	90023026	Relief	Canada	228988	273492	44504	16.3	146948	53.73	46.27
	90017051	Forward Stores	Portugal	86823	118067	31244	26.5	63568	53.84	46.16
	90017058	Mbit	Portugal	86860	110195	23335	21.2	59473	53.97	46.03
	90023028	walmart	Canada	239081	283323	44242	15.6	153058	54.02	45.98
	90023024	Sage	Canada	246397	287233	40836	14.2	155610	54.18	45.82
	90015146	Mbit	Norway	147152	210507	63355	30.1	114189	54.24	45.76
	90013124	Amazon	Italy	110898	136116	25218	18.5	73826	54.24	45.76
	90017054	Flawless Stores	Portugal	84371	114698	30327	26.4	62483	54.48	45.52
	70027208	Atiq e Store	Brazil	33713	47321	13608	28.8	25784	54.49	45.51
	90015147	Chiptec	Norway	154897	223867	68970	30.8	122100	54.54	45.46
	80001019	Neptune	China	1113979	1275248	161269	12.6	695779	54.56	45.44
	90015144	Sound	Norway	160074	225637	65563	29.1	123257	54.63	45.37
	90001000

Result 1

Result 1 x



Forecast Accuracy 2020 vs 2021

Query

```
1  # Get forecast accuracy of FY 2021 and store that in a temporary table
2  • drop table if exists forecast_accuracy_2021;
3  • create temporary table forecast_accuracy_2021
4  with forecast_err_table as (
5      select
6          s.customer_code as customer_code,
7          c.customer as customer_name,
8          c.market as market,
9          sum(s.sold_quantity) as total_sold_qty,
10         sum(s.forecast_quantity) as total_forecast_qty,
11         sum(s.forecast_quantity-s.sold_quantity) as net_error,
12         round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
13         sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
14         round(sum(abs(s.forecast_quantity-s.sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
15     from fact_act_est s
16     join dim_customer c
17     on s.customer_code = c.customer_code
18     where s.fiscal_year=2021
19     group by customer_code
20 )
21 select
22     *,
23     if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
24 from
25     forecast_err_table
26 order by forecast_accuracy desc;
```



Forecast Accuracy 2020 vs 2021

Query

```
28 # Get forecast accuracy of FY 2020 and store that also in a temporary table
29 • drop table if exists forecast_accuracy_2020;
30 • create temporary table forecast_accuracy_2020
31 with forecast_err_table as (
32     select
33         s.customer_code as customer_code,
34         c.customer as customer_name,
35         c.market as market,
36         sum(s.sold_quantity) as total_sold_qty,
37         sum(s.forecast_quantity) as total_forecast_qty,
38         sum(s.forecast_quantity-s.sold_quantity) as net_error,
39         round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
40         sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
41         round(sum(abs(s.forecast_quantity-s.sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
42     from fact_act_est s
43     join dim_customer c
44     on s.customer_code = c.customer_code
45     where s.fiscal_year=2020
46     group by customer_code
47 )
48 select
49     *,
50     if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
51 from
52     forecast_err_table
53 order by forecast_accuracy desc;
```



Forecast Accuracy 2020 vs 2021

Query

```
54
55  -- | Join forecast accuracy tables for 2020 and 2021 using a customer_code
56  • select
57      f_2020.customer_code,
58      f_2020.customer_name,
59      f_2020.market,
60      f_2020.forecast_accuracy as forecast_acc_2020,
61      f_2021.forecast_accuracy as forecast_acc_2021
62  from forecast_accuracy_2020 f_2020
63  join forecast_accuracy_2021 f_2021
64  on f_2020.customer_code = f_2021.customer_code
65  where f_2021.forecast_accuracy < f_2020.forecast_accuracy
66  order by forecast_acc_2020 desc;
67
```



Forecast Accuracy 2020 vs 2021

Result

Result Grid					
		Filter Rows:	Export:	Wrap Cell Content:	
	customer_code	customer_name	market	forecast_acc_2020	forecast_acc_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
	90014136	Reliance Digital	Netherlands	37.59	0.00
	80006154	Synthetic	Philippines	37.49	24.63
	70014142	Atliq Exclusive	Netherlands	37.43	0.00
	90014141	Amazon	Netherlands	37.39	0.00
	90005160	Nomad Stores	Pakistan	37.30	37.29
	90006156	Amazon	Philippines	37.21	27.94
	90008164	Digimarket	Australia	37.15	36.01
	90006153	Insight	Philippines	37.11	26.88
	90016173	Expert	Poland	36.76	35.26
	80006155	Novus	Philippines	36.59	25.28
	90019200	Sorefoz	Sweden	36.41	24.14
	70016177	Atliq Exclusive	Poland	36.22	34.87
	90004066	Surface Stores	Japan	36.05	33.51
	70006157	Atliq Exclusive	Philippines	35.84	25.20
	90004065	Neptune	Japan	35.67	30.27



**THANK
YOU...**

Gowhar Ahmad Dar

