Finance and Supply Chain Analytics

by Ng Hoi Yee

Project overview

AtliQ Hardware is a computer hardware company that manufactures various computer parts and peripherals. AtliQ operates in APAC (Asia Pacific), EU (Europe), NA (North America), and LATAM (Latin America). They sell their products at Croma and Best Buy stores as well as online on Amazon and Flipkart.

Problem statement

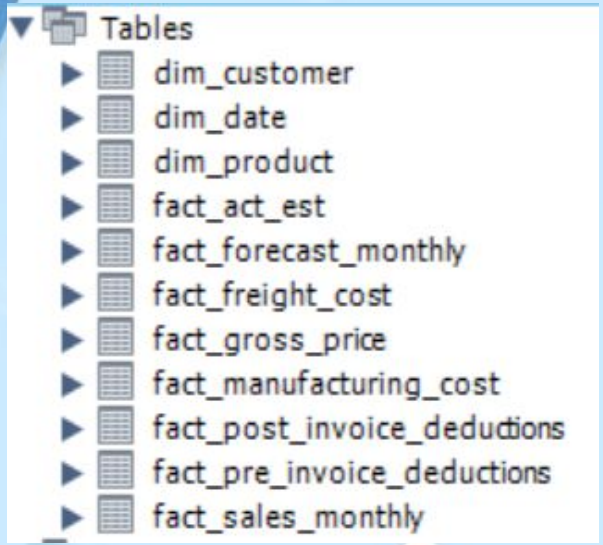
- Excel files getting bigger is causing performance issues, making things slow and inefficient.
- AtliQ Hardware is addressing the problem with a project that involves growing their data analytics team by hiring junior data analysts.
- They are using MySQL as their database management system to analyze data, understand the company's finanical position over the years, and extract valuable insights.
- The goal is to improve decision-making, optimize operations, and enhance overall performance.

Task

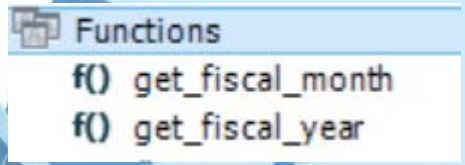
- Generate monthly gross sales report
- identify the financial position of the company over the various fiscal years
- identify the top products, markets and customers

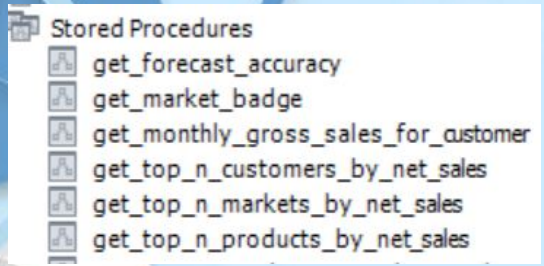
Dataset overview

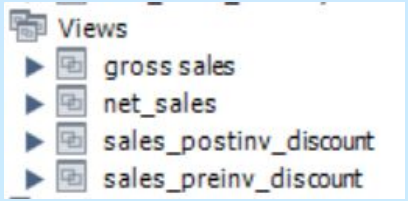
- loaded a million rows of data into a MySQL database and executed multiple queries.
- analyzed the following tables to derive the SQL concepts presented



- 2 user-defined SQL functions
- 5 stored procedures
- 4 views







Finance Analytics

create a function for fiscal year

```
CREATE DEFINER=`root`@`localhost` 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
```

create a function for fiscal quarter

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_month`(
CALENDAR_DATE DATE
) RETURNS char(2) CHARSET utf8mb4
    DETERMINISTIC
BEGIN

    declare m tinyint;
    declare qtr char(2);
    set m= month(CALENDAR_DATE);

case
when m in (9,10,11) then
set qtr="Q1";
when m in (12,1,2) then
set qtr="Q2";
when m in (3,4,5) then
set qtr="Q3";
    else
set qtr="Q4";

end case;
RETURN qtr;
END
```

Finance Analytics

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

```
SELECT
s.date, s.product_code,
p.product, p.variant, s.sold_quantity , g.gross_price,
round(g.gross_price* s.sold_quantity,2) as gross_price_total

FROM fact_sales_monthly s
join dim_product p
on p.product_code=s.product_code
join fact_gross_price g

on g.product_code =s.product_code and
g.fiscal_year=get_fiscal_year(s.date)
where
customer_code=90002002 and
get_fiscal_year(date)=2021
order by date asc
limit 100000000
```

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

Finance Analytics

Gross Sales Report 2: Monthly Product Transactions for all FY

```
select
s.date, sum(round(s.sold_quantity*g.gross_price,2)) as monthly_sales
FROM fact_sales_monthly s
join fact_gross_price g
on g.fiscal_year= get_fiscal_year(s.date)
and g.product_code=s.product_code
where customer_code =90002002
group by date;
```

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

Finance Analytics

Generate Monthly Gross Sales Report for any customer using stored procedure

```
ATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customer`(
        in_customer_codes TEXT
 )
IN
        SELECT
                s.date,
                SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
        FROM fact_sales_monthly s
        JOIN fact_gross_price g
                ON g.fiscal_year=get_fiscal_year(s.date)
                AND g.product_code=s.product_code
        WHERE
                FIND_IN_SET(s.customer_code, in_customer_codes) > 0
        GROUP BY s.date
        ORDER BY s.date DESC;
 END
```

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

Finance Analytics

Stored procedure for market badge (total sold quantity> 5 million = "Gold", else: "Silver"

```
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;
```

```
set @out_level = '0';
call gdb0041.get_market_badge('india', 2021, @out_level);
select @out_level;
```

	@out_level
▶	Gold

Supply Chain Analytics

Forecast Accuracy Report

```
SELECT
s.customer_code,
sum(s.sold_quantity) as total_sold_qty,
sum(s.forecast_quantity) as total_forecast_qty,
sum((forecast_quantity-sold_quantity)) as net_err,
sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity) as pct_net_err,
sum(abs(forecast_quantity-sold_quantity)) as abs_err,
sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity) as abs_err_pct

FROM gdb0041.fact_act_est s
where s.fiscal_year=2021
group by customer_code)

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
```

Supply Chain Analytics

Forecast Accuracy Report

customer_code	total_sold_qty	total_forecast_qty	net_err	pct_net_err	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

Supply Chain Analytics

Forecast Accuracy 2020 vs 2021

```
create temporary table forecast_accuracy_2021
with forecast_err_table as
(
SELECT
s.customer_code,
c.customer,
c.market,

sum(s.sold_quantity) as total_sold_qty,
sum(s.forecast_quantity) as total_forecast_qty,
sum((forecast_quantity-sold_quantity)) as net_err,
round(sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as pct_net_err,
sum(abs(forecast_quantity-sold_quantity)) as abs_err,
round(sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as abs_err_pct

FROM gdb0041.fact_act_est s

join dim_customer c
on s.customer_code =c.customer_code
where s.fiscal_year=2021
group by customer_code
)
 SELECT *,
if (abs_err_pct >100, 0,100-abs_err_pct) as forecast_accuracy
from forecast_err_table
order by forecast_accuracy desc;

drop table if exists forecast_accuracy_2020;
create temporary table forecast_accuracy_2020
with forecast_err_table as
```

Supply Chain Analytics

Forecast Accuracy 2020 vs 2021

```
(
select
s.customer_code,
c.customer,
c.market,

sum(s.sold_quantity) as total_sold_qty,
sum(s.forecast_quantity) as total_forecast_qty,
sum((forecast_quantity-sold_quantity)) as net_err,
round(sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as pct_net_err,
sum(abs(forecast_quantity-sold_quantity)) as abs_err,
round(sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as abs_err_pct
FROM gdb0041.fact_act_est s

join dim_customer c
on s.customer_code =c.customer_code
where s.fiscal_year=2020
group by customer_code
)

select *,
if (abs_err_pct >100, 0,100-abs_err_pct) as forecast_accuracy
from forecast_err_table
order by forecast_accuracy desc;
```

Supply Chain Analytics

Forecast Accuracy 2020 vs 2021

```
select
f_2020.customer_code,
f_2020.customer,
f_2020.market,
f_2020.forecast_accuracy as forecast_accuracy_2020,
f_2021.forecast_accuracy as forecast_accuracy_2021


from forecast_accuracy_2020 f_2020
join forecast_accuracy_2021 f_2021
on f_2020.customer_code = f_2021.customer_code
where f_2021.forecast_accuracy < f_2020.forecast_accuracy
order by f_2020.forecast_accuracy desc
;
```

customer_code	customer	market	forecast_accuracy_2020	forecast_accuracy_2021
70006158	Atliq e Store	Philiphines	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

Thank you