SQL: Finance Analytics

1.first grab customer codes for Crom

```
SELECT * FROM dim_customer
where
customer like "%croma%" and market = "india"
```

Output

customer_code	customer	platform	channel	market	sub_zone	region
90002002	Croma	Brick & Mortar	Retailer	India	India	APAC
NULL	NULL	NULL	NULL	NULL	NULL	NULL

USER DEFINED FUNCTIONS

2. create a function 'get_fiscal_year' to get fiscal year(begins in September) by passing the date

```
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
```

3.create another function' get_fiscal_quater' to generate quarter wise sales

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_quarter`(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
```

4.Croma India product wise sales report for fiscal year 2021

```
SELECT
s. date, s.product_code, p.product, p. variant,
 s. sold_quantity, round( g. gross_price,2),
  round( g. gross_price * s. sold_quantity,2) as gross_price_total
 FROM fact_sales_monthly s
join dim_product p
on s. product_code=p. product_code
join fact_gross_price g
on s. product_code = g.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 1000000;
```

Output

date	product_code	product	variant	sold_quantity	round(g. gross_price,2)	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.06	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.46	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.78	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.97	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.70	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107	24.73	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123	23.62	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.72	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.10	6396.24
2020-09-01	A0321150303	AQ Zion Saga	Premium	137	28.01	3836.81
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	23	19.52	449.04
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	82	19.92	1633.76

5.Gross Monthly total sales report for Croma

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

Output

date	total_gross_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
2018-05-01	139165.10
2018-06-01	125735.38
2018-08-01	125409.88
2018-09-01	343337.17

Created by Manju

6.Get Yearly Gross sales report for Croma India

```
SELECT
      get_fiscal_year (date) as Fiscal_year,
      round(sum(s. sold_quantity* g. gross_price),2) as total_gross_price
FROM fact_sales_monthly s
join fact_gross_price g
on  s. product_code= g. product_code and
     g. fiscal_year= get_fiscal_year(s. date)
where customer_code= 90002002
group by fiscal_year
order by fiscal_year asc
```

Output

Fiscal_year	total_gross_price
2018	1324097.44
2019	3555079.02
2020	6502181.91
2021	23216512.22
2022	44638198.92

STORED PROCEDURE

7.Create a stored procedure to get monthly gross sales for any customer

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customer`(c_code int)
BEGIN
      SELECT s. date,
         round(sum(s. sold_quantity * g. gross_price),2) as monthly_sales
FROM fact_sales_monthly s
join fact_gross_price g
on s. product_code = g. product_code and
   g. fiscal_year = get_fiscal_year(s.date)
where customer_code= c_code
group by date;
END
```

8.Create a stored procedure to get monthly gross sales for customers having 2 customer codes

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customers_having_two_codes`
                  (in_customer_code text)
BEGIN
      SELECT s. date,
        round(sum(s. sold_quantity * g. gross_price),2) as monthly_sales
FROM fact_sales_monthly s
join fact_gross_price g
on s. product_code = g. product_code and
   g. fiscal_year = get_fiscal_year(s.date)
where
       find_in_set( s. customer_code, in_customer_code)>0
group by date;
END
```

9.Create a stored procedure to get market badge

#If total sold quantity >5million that market is considered Gold else it is Silver

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`( in in_market varchar(45),
                    in in_fiscal_year year,
                   out out_badge varchar(45))
BEGIN
     declare qty int default 0;
     #set default market to be india
     if in_market= " " then
     set in_market= "india";
     end if;
  #retrive total qty for a given market in a given year
  select
        sum( sold_quenty) into qty
        from fact_sales_monthly s
        join dim_customer c
        on s. customer_code = c. csutomer_code
  where get_fiscal_year(s.date)= in_fiscal_year and
        c. market= in_market
  group by c. market;
  #determine market badge
    if qty > 5000000 then
    set out_badge = "gold";
    else
    set out_badge = "silver";
    end if;END
```

Top Market, Customers and Products Analysis

PERFORMANCE IMPROVEMENT

10.Added the fiscal year in the fact_sales_monthly table itself to get pre invoice discount pct

```
SELECT
s. date, s.product_code, p.product, p. variant,
 s. sold_quantity, round( g. gross_price,2) as gross_price_per_item,
  round( g. gross_price * s. sold_quantity,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 s. product_code = g.product_code and
   g. fiscal_year= s. fiscal_year
join fact_pre_invoice_deductions pre
on pre. customer_code= s. customer_code and
   pre. fiscal_year= s. fiscal_year
where
      s. fiscal_year = 2021
limit 1000000;
```

Output

date	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	248	19.06	4726.21	0.0703
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	240	19.06	4573.75	0.2061
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	31	19.06	590.78	0.0974
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	37	19.06	705.12	0.2065
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.06	133.40	0.1068
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	12	19.06	228.69	0.2612
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	19.06	323.97	0.2471
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	60	19.06	1143.44	0.0858
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	19.06	647.95	0.2450
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	24	19.06	457.38	0.0736

DATABASE VIEWA

11. Creating the view `sales pre invoice discount` and store all the data in like a virtual table

```
CREATE  VIEW `sales_preinv_discount` AS
    SELECT s.date, s.fiscal_year,s.customer_code,c.market,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_customer c
        ON s.customer_code = c.customer_code
    JOIN dim_product p
            ON s.product_code=p.product_code
    JOIN fact_gross_price g
            ON g.fiscal_year=s.fiscal_year
            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=s.fiscal_year
```

Now generate net_invoice_sales using the above created view "sales_pre_invoice_discount"

```
SELECT
	*,
	(1-pre_invoice_discount_pct)*gross_price_total as net_invoice_sales
FROM sales_pre_invoice_discount
```

Output

date	fiscal_year	customer_code	market	product_code	product	variant	sold_quantity	gross_price_p	gross_price_tota	pre_invoice_discount_pct	net_invoice_sales
2017-09-01	2018	70002017	India	A0118150101	AQ Drac...	Standard	51	15.40	785.16	0.0824	720.462816
2017-09-01	2018	70002018	India	A0118150101	AQ Drac...	Standard	77	15.40	1185.43	0.2956	835.016892
2017-09-01	2018	70003181	Indonesia	A0118150101	AQ Drac...	Standard	17	15.40	261.72	0.0536	247.691808
2017-09-01	2018	70003182	Indonesia	A0118150101	AQ Drac...	Standard	6	15.40	92.37	0.2378	70.404414
2017-09-01	2018	70006157	Philiphines	A0118150101	AQ Drac...	Standard	5	15.40	76.98	0.1057	68.843214
2017-09-01	2018	70006158	Philiphines	A0118150101	AQ Drac...	Standard	7	15.40	107.77	0.1875	87.563125
2017-09-01	2018	70007198	South Korea	A0118150101	AQ Drac...	Standard	29	15.40	446.46	0.0700	415.207800
2017-09-01	2018	70007199	South Korea	A0118150101	AQ Drac...	Standard	34	15.40	523.44	0.2551	389.910456
2017-09-01	2018	70008169	Australia	A0118150101	AQ Drac...	Standard	22	15.40	338.69	0.0953	306.412843

12. Create a view for post invoice deductions: `sales_postinv_discount`

```
CREATE VIEW `sales_postinv_discount` AS
    SELECT
            s.date, s.fiscal_year,
            s.customer_code, s.market,
            s.product_code, s.product, s.variant,
            s.sold_quantity, s.gross_price_total,
            s.pre_invoice_discount_pct,
            (s.gross_price_total-s.pre_invoice_discount_pct*s.gross_price_total) as net_invoice_sales,
            (po.discounts_pct+po.other_deductions_pct) as post_invoice_discount_pct
    FROM sales_preinv_discount s
    JOIN fact_post_invoice_deductions po
        ON po.customer_code = s.customer_code AND
        po.product_code = s.product_code AND
        po.date = s.date;
```

Now generate net_sales using the above created view "sales_postinv_discount"

```
SELECT
	*,
	(1-post_invoice_discount_pct)*net_invoice_sales as net_sales
from sales_post_invoice_discount
```

Output

date	fiscal_year	customer_code	market	product_code	product	variant	sold_quantit	gross_price_	pre_invoice_dis	net_invoice_sal	post_invoice_discount_	net_sales
2017-09-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	4	61.58	0.2803	44.319126	0.3905	27.0125072970
2017-11-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	16	246.32	0.2803	177.276504	0.4139	103.9017589944
2017-12-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	4	61.58	0.2803	44.319126	0.3295	29.7159739830
2018-01-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	6	92.37	0.2803	66.478689	0.3244	44.9130022884
2018-03-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	9	138.56	0.2803	99.721632	0.3766	62.1664653888
2018-04-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	6	92.37	0.2803	66.478689	0.3615	42.4466429265
2018-05-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	7	107.77	0.2803	77.562069	0.3173	52.9516245063
2018-07-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	10	153.95	0.2803	110.797815	0.3501	72.0074999685
2018-08-01	2018	90027207	Brazil	A0118150101	AQ Dra...	Standard	6	92.37	0.2803	66.478689	0.3740	41.6156593140

13. Finally creating the view `net_sales` which inbuilt use/include all the previous created view and gives the final result

```
CREATE VIEW `net_sales` AS
    SELECT
            *,
            net_invoice_sales*(1-post_invoice_discount_pct) as net_sales
    FROM gdb041.sales_postinv_discount;
```

Created by Manju

14. Get top 5 market by net sales in fiscal year 2021

```
SELECT
      market,
      round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales
where fiscal_year= 2021
group by market
order by net_sales_million desc
limit 5
```

Output

market	net_sales_million
India	210.67
USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

15. Stored procedure to get top n markets by net sales for a given year

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_markets_by_net_sales`(in_fiscal_year year,
                in_top_n int)
BEGIN
     SELECT
      market,
      round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales
where fiscal_year= in_fiscal_year
group by market
order by net_sales_million desc
limit in_top_n;
END
```

16. Get top 5 Customers by net sales in fiscal year 2021

```
SELECT
      c.customer,
      round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales n
join dim_customer c
on n.customer_code= c.customer_code
where fiscal_year= 2021
group by c.customer
order by net_sales_million desc
limit 5
```

Output

customer	net_sales_million
Amazon	109.03
Atliq Exclusive	79.92
Atliq e Store	70.31
Sage	27.07
Flipkart	25.25

17. Stored procedure to get top n customers by net sales for a given year

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_customers_by_net_sales`(in_market varchar(45),
                    in_fiscal_year year,
                     in_top_n int)
BEGIN
      SELECT
       c.customer,
       round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales n
join dim_customer c
on n.customer_code= c.customer_code
where s.fiscal_year= in_fiscal_year and
      s.market= in_market
group by c.customer
order by net_sales_million desc
limit in_top_n;
END
```

18. Get top 5 Products by net sales in fiscal year 2021

```
SELECT
        product,
        round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales
where fiscal_year = 2021
group by product
order by net_sales_million
limit 5
```

Output

product	net_sales_million
AQ LION x1	1.15
AQ Pen Drive 2 IN 1	1.54
AQ LION x2	1.65
AQ Marquee P4	1.76
AQ LION x3	2.11

19. Stored procedure to get top n products by net sales for a given year

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_products_by_net_sales`(in_fiscal_year year,
                    in_top_n int)
BEGIN
     SELECT
       product,
       round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales
where fiscal_year = in_fiscal_year
group by product
order by net_sales_million
limit in_top_n;
END
```

WINDOW FUNCTION OVER CLAUSE

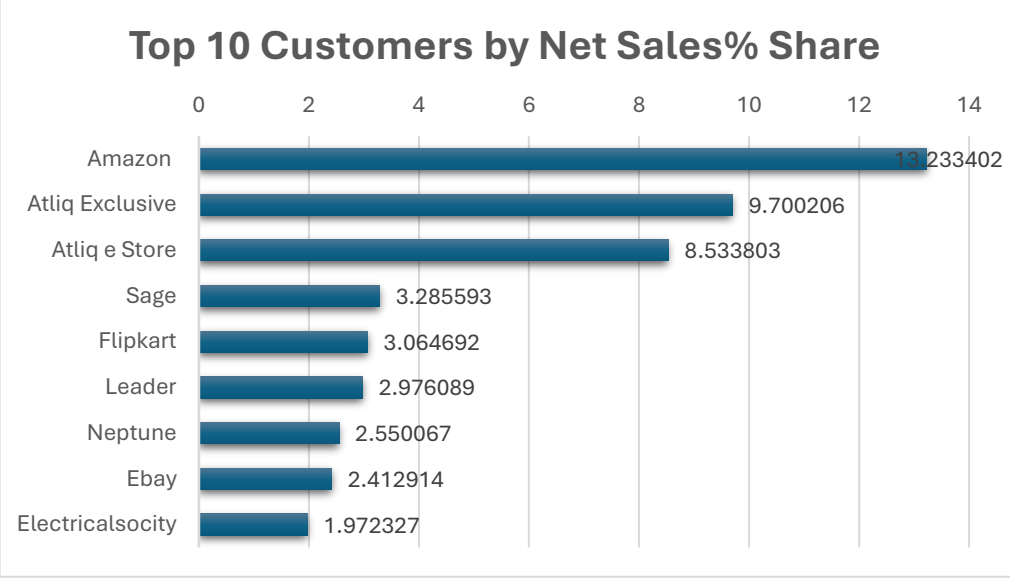
20. find out customer wise net sales percentage contribution

```
with cte as
    (SELECT
       c.customer,
       round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales n
join dim_customer c
on n.customer_code= c.customer_code
where n.fiscal_year= 2021
group by c.customer)
  select
        *,
        net_sales_million*100/sum(net_sales_million) over() as net_sales_pct
  from cte
order by net_sales_million desc
```

Output

customer	net_sales_million	net_sales_pct
Amazon	109.03	13.233402
Atliq Exclusive	79.92	9.700206
Atliq e Store	70.31	8.533803
Sage	27.07	3.285593
Flipkart	25.25	3.064692
Leader	24.52	2.976089
Neptune	21.01	2.550067
Ebay	19.88	2.412914
Electricalsocity	16.25	1.972327
Synthetic	16.10	1.954121
Electricalslytical	15.64	1.898289

Top 10 Customers by Net Sales% Share



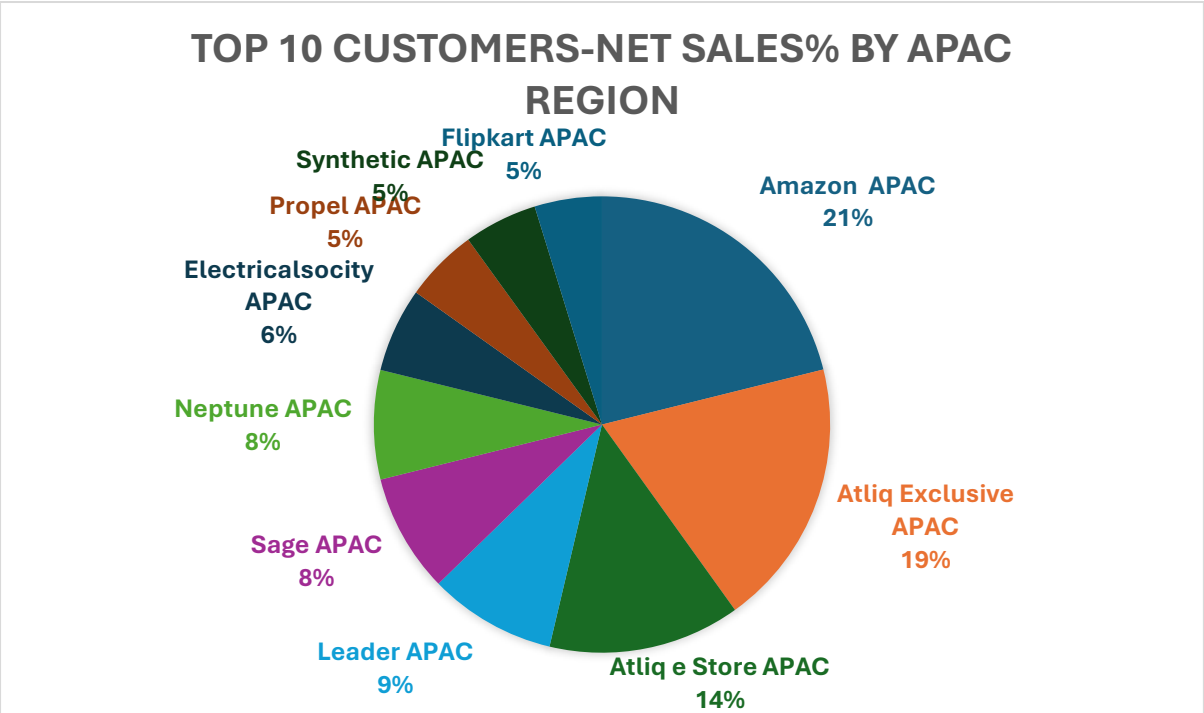
21. find out Region wise net sales percentage contribution

```
with cte as
    (SELECT
       c.customer, c.region,
       round(sum(net_sales)/1000000,2) as net_sales_million
FROM net_sales n
join dim_customer c
on n.customer_code= c.customer_code
where n.fiscal_year= 2021
group by c.customer, c.region)
  select
        *,
        net_sales_million*100/sum(net_sales_million) over(partition by region) as pct_share_region
  from cte
order by region,net_sales_million desc
```

Output

customer	region	net_sales_million	pct_share_region
Amazon	APAC	57.41	12.988688
Atliq Exclusive	APAC	51.58	11.669683
Atliq e Store	APAC	36.97	8.364253
Leader	APAC	24.52	5.547511
Sage	APAC	22.85	5.169683
Neptune	APAC	21.01	4.753394
Electricalsocity	APAC	16.25	3.676471
Propel	APAC	14.14	3.199095
Synthetic	APAC	14.14	3.199095
Flipkart	APAC	12.96	2.932127
Novus	APAC	12.91	2.920814
Expression	APAC	12.90	2.918552

Top 10 Customers for APAC-Region



Created by Manju

WINDOW FUNCTION DENSE RANK

22. Find out top 3 products from each division by total quantity sold in a given year

```
with cte1 as
    (SELECT
      p.division, p.product,sum(sold_quantity) as total_qty
FROM fact_sales_monthly s
join dim_product p
on s.product_code=p.product_code
where fiscal_year= 2021
group by p.product),
cte2 as
     (select *,dense_rank() over(partition by division
      order by total_qty desc) as drank
      from cte1)
select * from cte2
where drank <=3
```

Output

division	product	total_qty	drank
N & S	AQ Pen Drive DRC	2034569	1
N & S	AQ Digit SSD	1240149	2
N & S	AQ Clx1	1238683	3
P & A	AQ Gamers Ms	2477098	1
P & A	AQ Maxima Ms	2461991	2
P & A	AQ Master wireless x1 Ms	2448784	3
PC	AQ Digit	135092	1
PC	AQ Gen Y	135031	2
PC	AQ Elite	134431	3

23. Creating stored procedure for the above query

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_products_per_division_by_qty_sold`(in_fiscal_year year,
                    in_top_n int)
BEGIN
     with cte1 as
    (SELECT
      p.division, p.product,sum(sold_quantity) as total_qty
FROM fact_sales_monthly s
join dim_product p
on s.product_code=p.product_code
where fiscal_year= in_fiscal_year
group by p.product),
cte2 as
     (select *, dense_rank() over(partition by division
      order by total_qty desc) as drank
      from cte1)
select * from cte2
where drank <=in_top_n;
END
```

SUPPLY CHAIN ANALYSIS

CREATE A HELPER TABLE

24. create table fact_act_est

```
create table fact_act_est
     ( SELECT
           s.date as date,
           s.fiscal_year as fiscal_year,
           s.product_code as product_code,
           s.customer_code as customer_code,
           s.sold_quantity as sold_quantity,
           f.forecast_quantity as forecast_quantity
       FROM fact_sales_monthly s
       left join fact_forecast_monthly f
       using (date, customer_code, product_code)
       union
       SELECT
           f.date as date,
           f.fiscal_year as fiscal_year,
           f.product_code as product_code,
           f.customer_code as customer_code,
           s.sold_quantity as sold_quantity,
           f.forecast_quantity as forecast_quantity
       FROM fact_sales_monthly s
       right join fact_forecast_monthly f
       using (date, customer_code, product_code));
```

Output

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

25.To Replace null values with 0

```
update fact_act_est
set sold_quantity =0
where sold_quantity is null;

update fact_act_est
set forecast_quantity =0
where forecast_quantity is null;
```

Output

date	fiscal_year	product_code	customer_code	sold_quantity	forecast_quantity
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

26.Forecast Accuracy Report

```
with forecast_error_table as
   (SELECT
      s.customer_code as customer_code,
      c. customer as customer_name,
      c. 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 fiscal_year= 2021
group by customer_code)
select
      *,
 if(abs_error_pct >100, 0, 100-abs_error_pct) as forecast_accuracy
 from forecast_error_table
 order by forecast_accuracy desc;
```

Output

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

27.Now Create a stored procedure for Forecast Accuracy

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_forecast_accuracy`(in_fiscal_year int)
BEGIN
      with forecast_error_table as
   (SELECT
      s.customer_code as customer_code,
      c. customer as customer_name,
      c. 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 fiscal_year= in_fiscal_year
group by customer_code)
select
      *,
 if(abs_error_pct >100, 0, 100-abs_error_pct) as forecast_accuracy
 from forecast_error_table
 order by forecast_accuracy desc;
END
```