Finance and Supply Chain Analytics

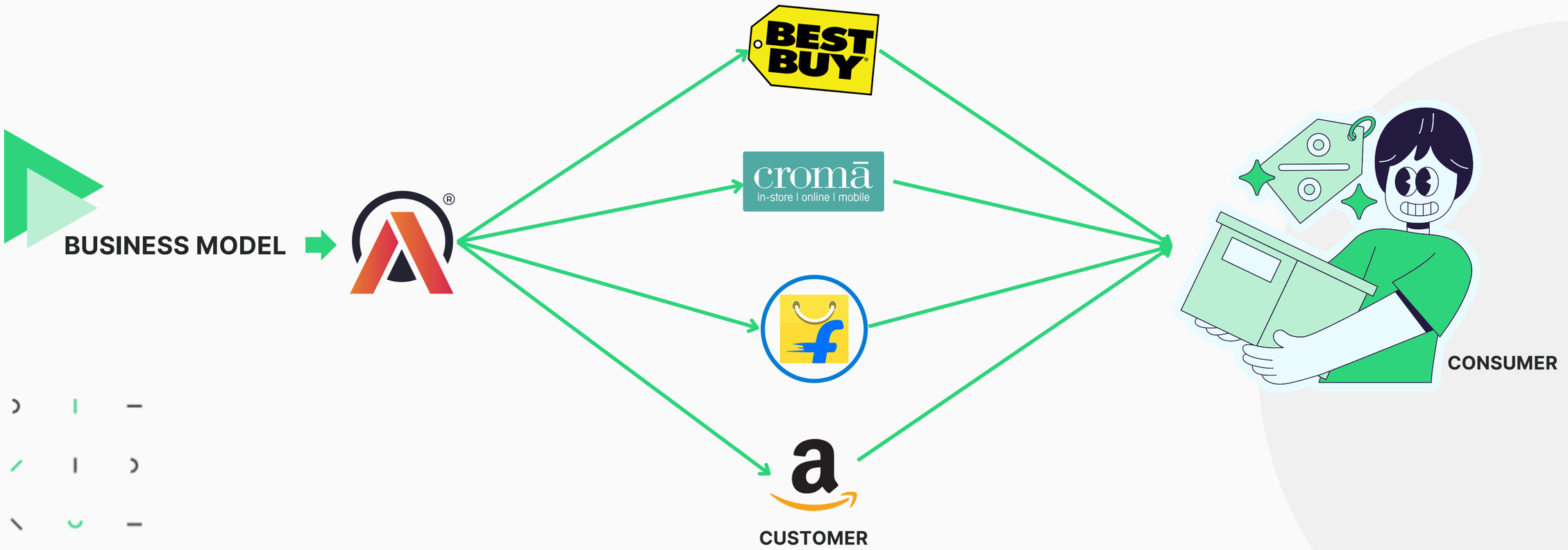
01

Outline

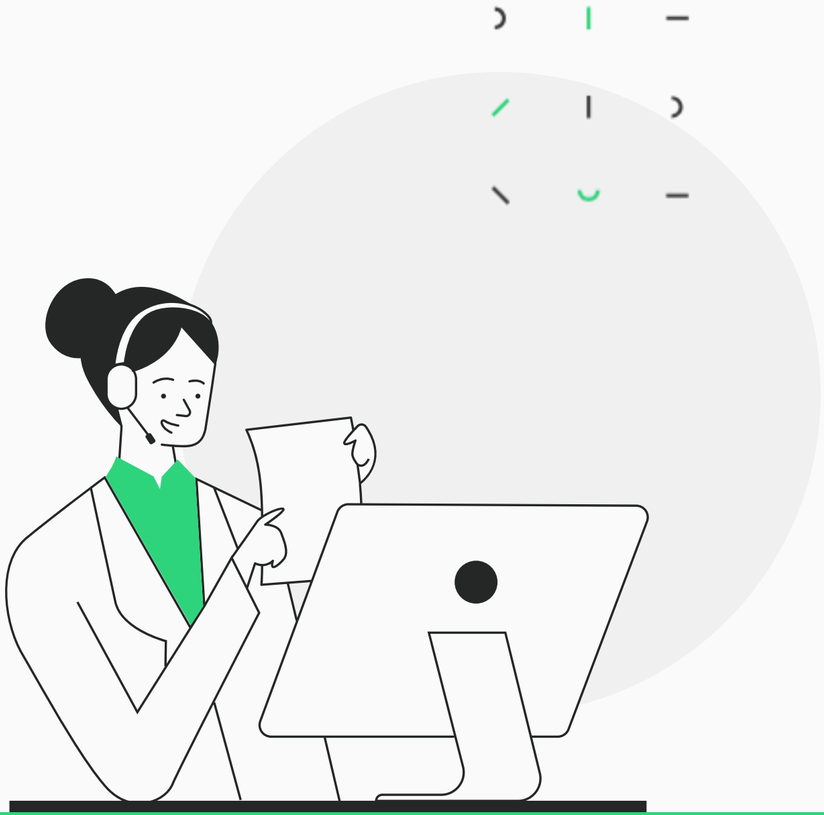
- Company details
- Problem Statement
- Objective of the Project
- Understanding Database
- Reports
- Key Insights

Company Details

AtliQ Hardware is a leading electronics manufacturing company that sells hardware's like PCs, Storage devices, Network peripherals, keyboards, mouse, printers etc... across the country



Platforms



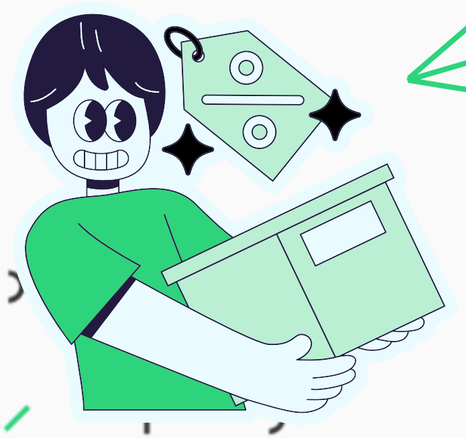
BRICK & MOTOR

BEST BUY

cromā
in-store | online | mobile

E-COMMERCE

Customer Channels



RETAILER

DIRECT

e store

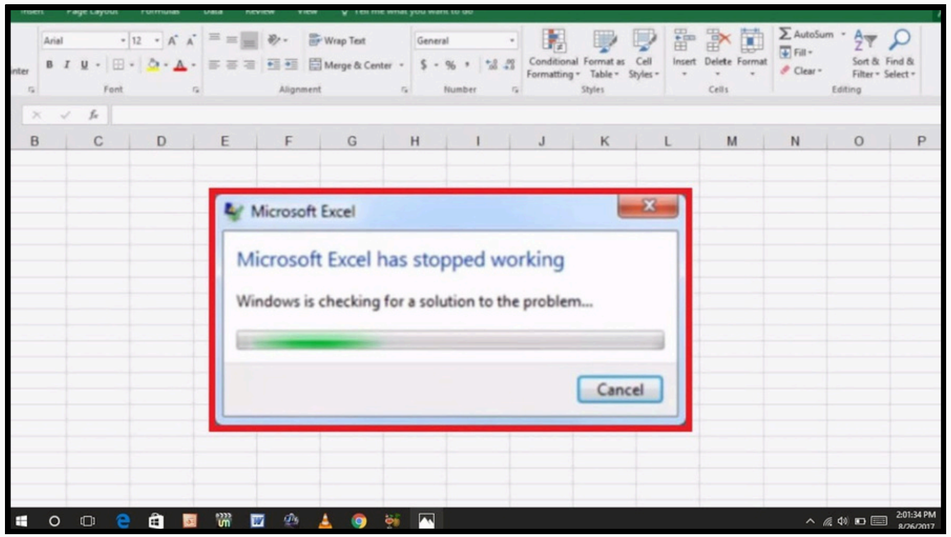
Exclusive

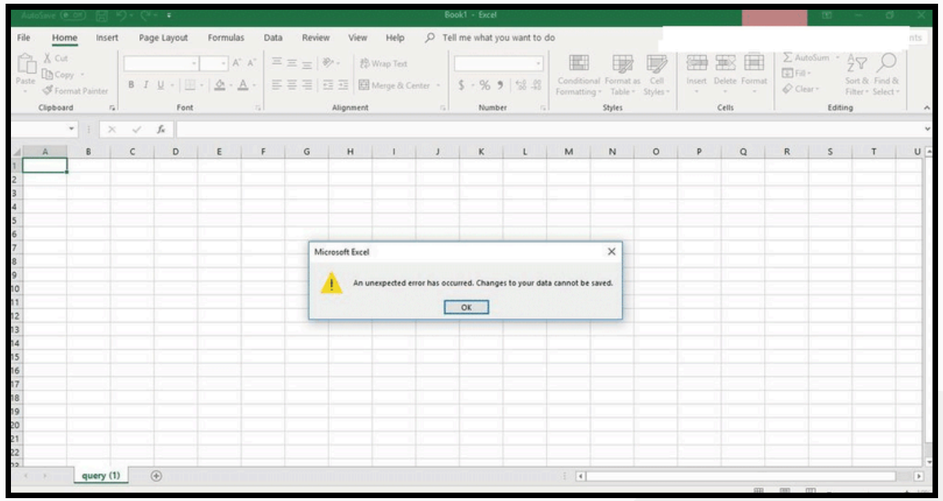
INDIRECT

NEPTUNE

Problem Statement:

The performance of Excel has decreased due to the growing size of Excel files, leaving it in a fragile and non-functional state. In order to solve this problem, AtliQ Hardware has employed data specialists who will evaluate and draw conclusions from the data using MYSQL as their database.





Project Objectives

To solve queries related to sales, finance, Market, regional analysis, customer behavior and supply chain forecasting

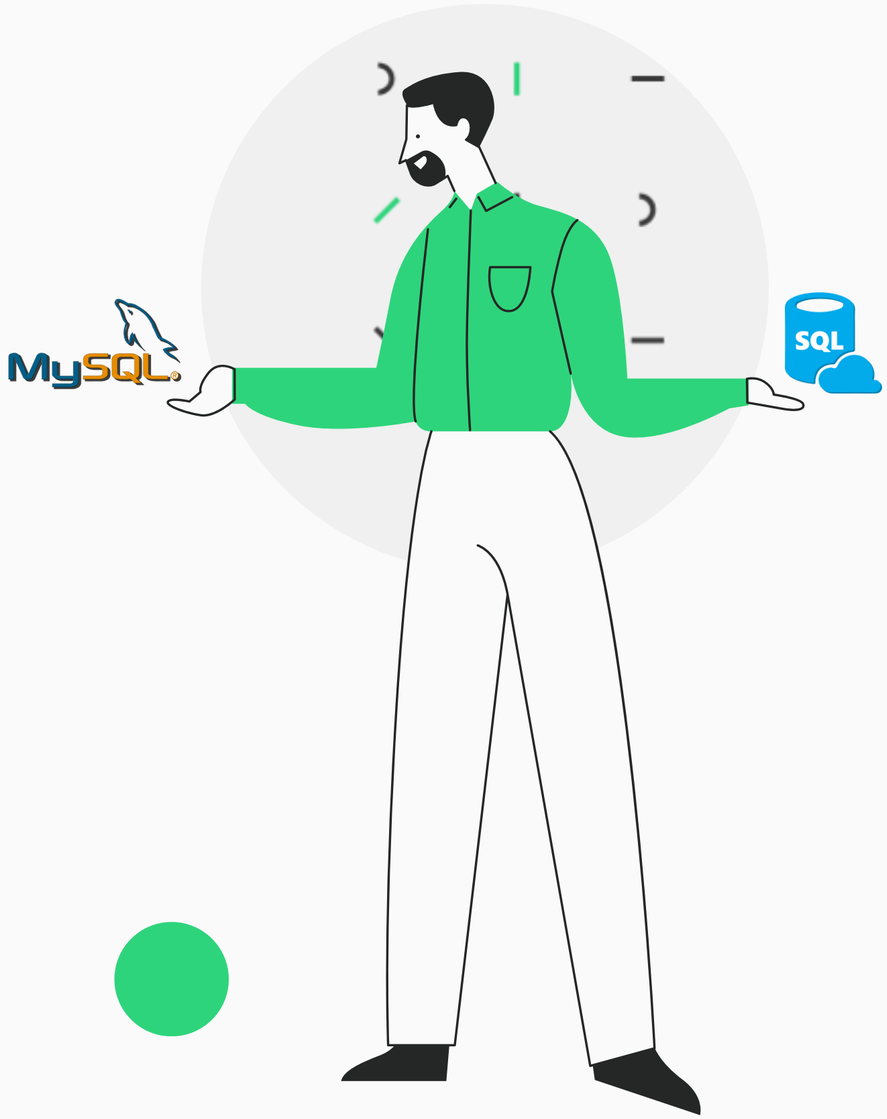
Database

GDB0041

- FACT TABLES
  - FACT_SALES_MONTHLY
  - FACT_PRE_INVOICE_DEDUCTIONS
  - FACT_POST_INVOICE_DEDUCTIONS
  - FACT_MANUFACTURING_COST
  - FACT_GROSS_PRICE
  - FACT_FREIGHT_COST
- DIM TABLES
  - DIM_CUSTOMER
  - DIM_PRODUCT
  - DIM_FORECAST_MONTHLY

Generating Insights

Generating important insights as per the requirements given by the product owner or stakeholders of the company.



Report Generation

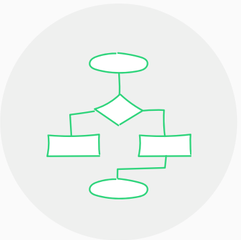
To create a stunning reports, presentation, it's best to simplify your thoughts. Start with an outline of topics and identify highlights, which can be applied to creating presentations

REPORTS

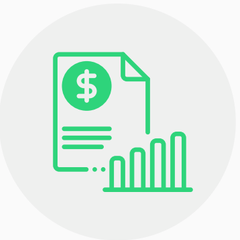
user defined functions

Stored procedures

Views



Queries



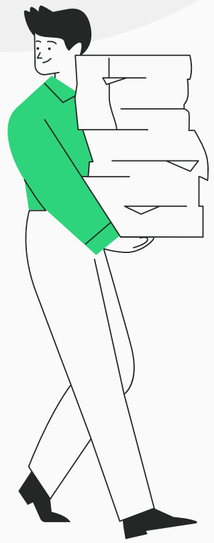
Forecasting

Generate a report by creating a user defined function get_fiscal_year to get individual product sales for croma india customer of FY 2021

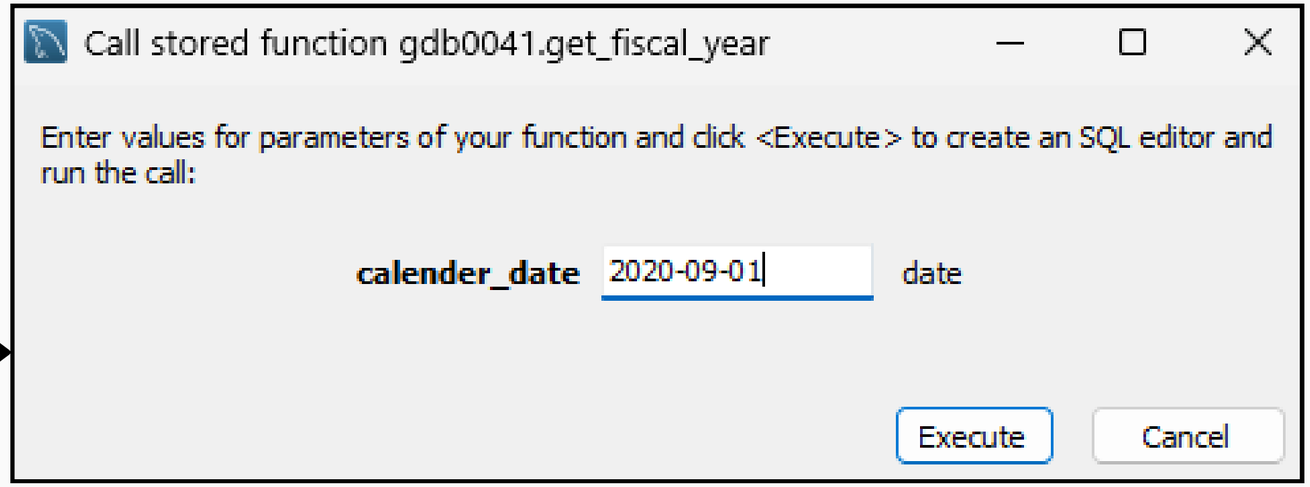
USER DEFINED FUNCTION

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_year`(
     calender_date date
) RETURNS int
    DETERMINISTIC
BEGIN
    declare fiscal_year int;
    set fiscal_year = year(date_add(calender_date, interval 4 month));
return fiscal_year;
END
```

SQL QUERY



OUTPUT

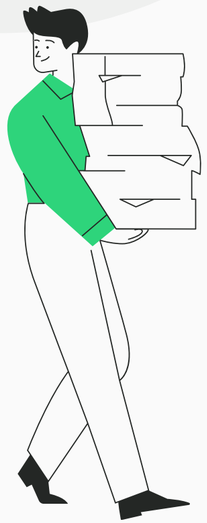


USER DEFINED FUNCTION

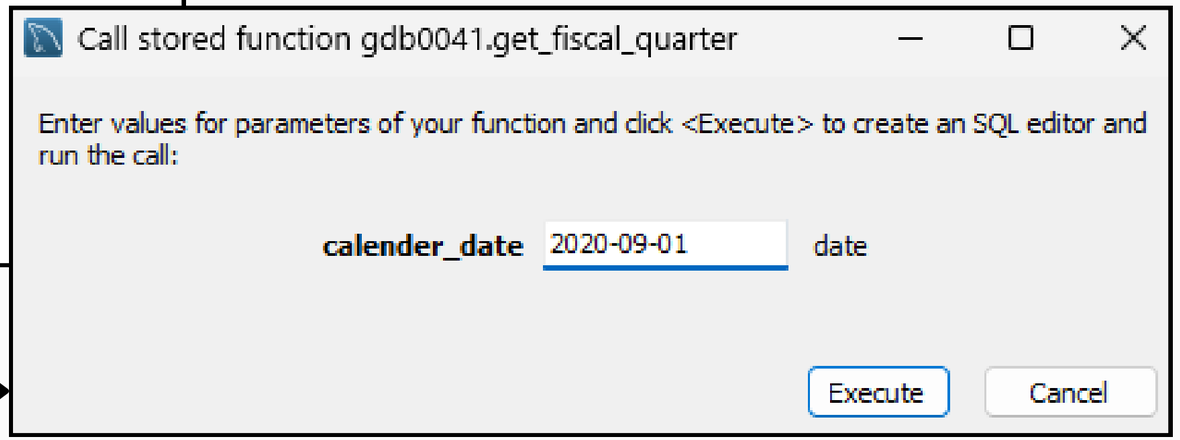
Generate a report by creating a user defined function get_fiscal_quarter for fiscal year to get report of any fiscal year by quarter

SQL QUERY

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_quarter`(
    calender_date date
) RETURNS char(2) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
     declare m tinyint;
     declare qtr char(2);
     set m=month(calender_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
```



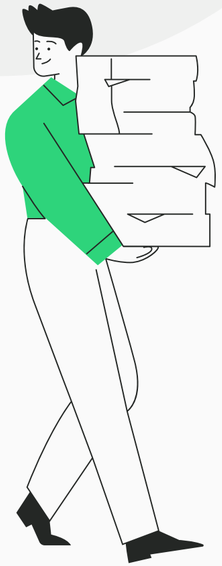
OUTPUT



Generate a report of individual product sales for croma india customer of FY 2021

SQL QUERY

```sql
SELECT
    s.date,
    s.product_code,
    p.product_code,p.variant,
    s.sold_quantity,
    g.gross_price,
    round(s.sold_quantity*g.gross_price,2) as total_gross_price
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;
```



OUTPUT

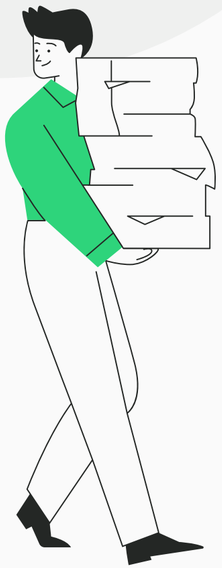
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

date	product_code	product_code	variant	sold_quantity	gross_price	total_gross_price
2020-09-01	A0118150101	A0118150101	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	A0118150102	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	A0118150103	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	A0118150104	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	A0219150201	Standard	149	23.6987	3531.11
2020-09-01	A0219150202	A0219150202	Plus	107	24.7312	2646.24
2020-09-01	A0220150203	A0220150203	Premium	123	23.6154	2904.69

Generate an aggregate monthly gross sales report for Croma India customers.

```sql
select
    s.date,
    sum(round(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
    customer_code = 90002002
group by s.date
order by s.date;
```

SQL QUERY



OUTPUT

Result Grid | Filter Rows:

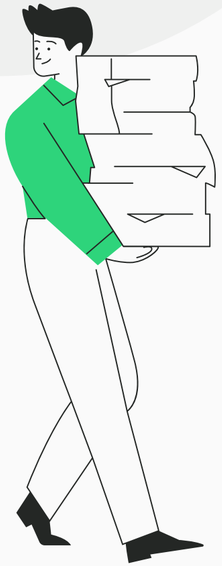
date	gross_price_total
2017-09-01	122407.57
2017-10-01	162687.56
2017-12-01	245673.84
2018-01-01	127574.73
2018-02-01	144799.54
2018-04-01	130643.92
2018-05-01	139165.06
2018-06-01	125735.36
2018-08-01	125409.90

STORED PROCEDURE

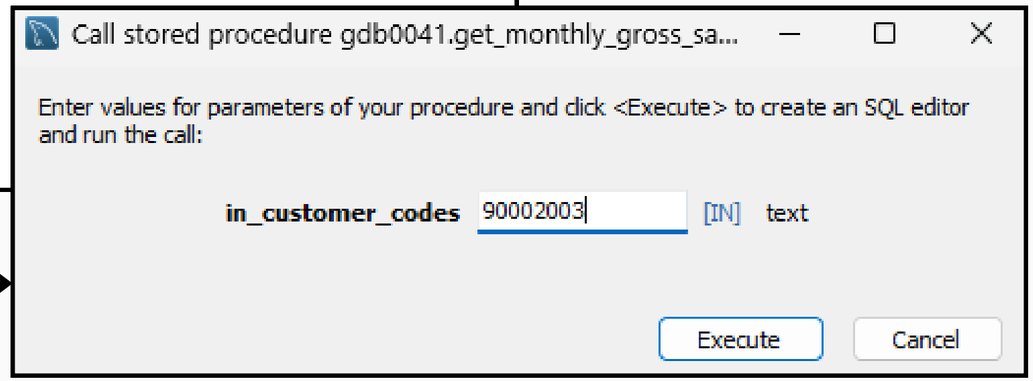
Create an stored procedure get_monthly_gross_sales_customer to generate aggregate monthly gross sales report for any customers.

SQL QUERY

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customers`(
     in_customer_codes text
)
BEGIN
select
      s.date,
      sum(round(g.gross_price*s.sold_quantity,2)) as gross_price_total
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;
END
```



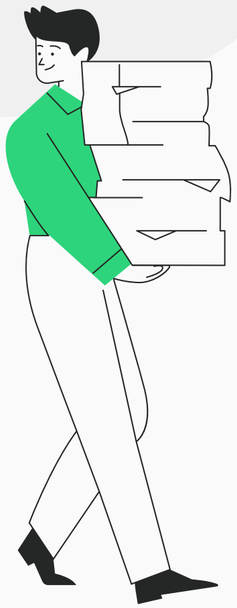
OUTPUT



Create a yearly gross sales report for Croma India customers

```sql
select
     g.fiscal_year as fiscal_year,
     sum(round(s.sold_quantity*g.gross_price,2)) as yearly_sales
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 customer_code = 90002002
group by fiscal_year
order by fiscal_year
```

SQL QUERY

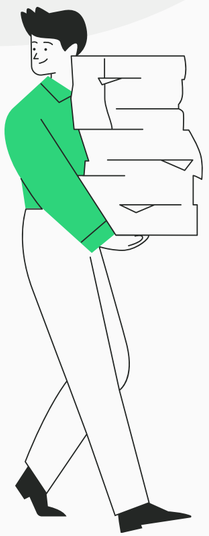


OUTPUT

fiscal_year	yearly_sales
2018	1324097.48
2019	3555079.19
2020	6502182.12
2021	23216512.73
2022	44638199.11

Generate a report of top market net sales for a given financial year

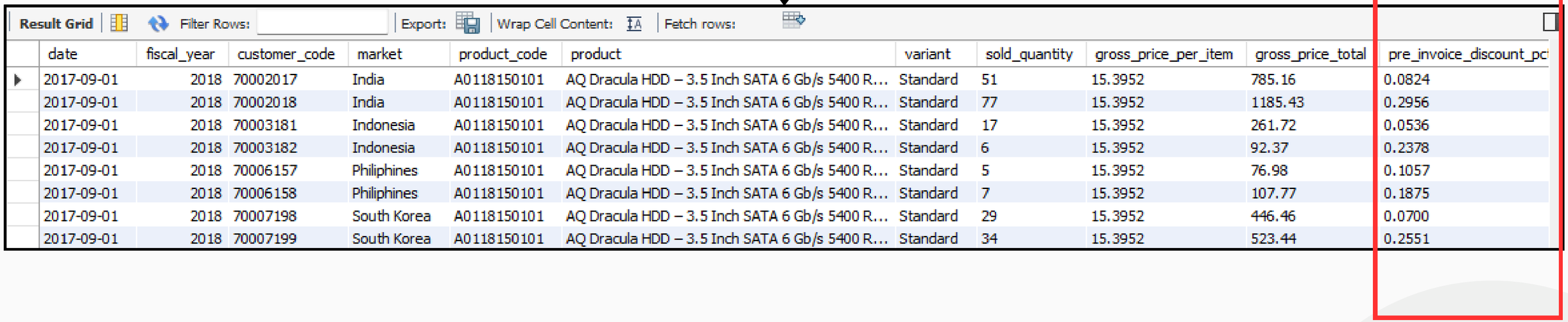
VIEWS



SQL QUERY

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `sales_preinv_discount` AS
    SELECT
        `s`.`date` AS `date`,
        `s`.`fiscal_year` AS `fiscal_year`,
        `s`.`customer_code` AS `customer_code`,
        `c`.`market` AS `market`,
        `s`.`product_code` AS `product_code`,
        `p`.`product` AS `product`,
        `p`.`variant` AS `variant`,
        `s`.`sold_quantity` AS `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` AS `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` `pre` ON (((`pre`.`customer_code` = `s`.`customer_code`)
            AND (`pre`.`fiscal_year` = `s`.`fiscal_year`))))
```

OUTPUT



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

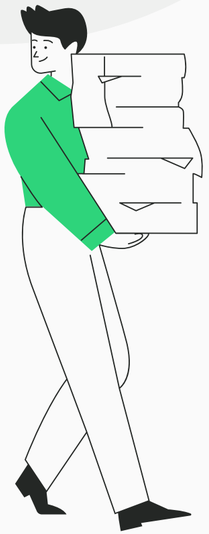
date	fiscal_year	customer_code	market	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pc
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16	0.0824
2017-09-01	2018	70002018	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43	0.2956
2017-09-01	2018	70003181	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72	0.0536
2017-09-01	2018	70003182	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37	0.2378
2017-09-01	2018	70006157	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1057
2017-09-01	2018	70006158	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77	0.1875
2017-09-01	2018	70007198	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46	0.0700
2017-09-01	2018	70007199	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44	0.2551

Generate a report of top market net sales for a given financial year

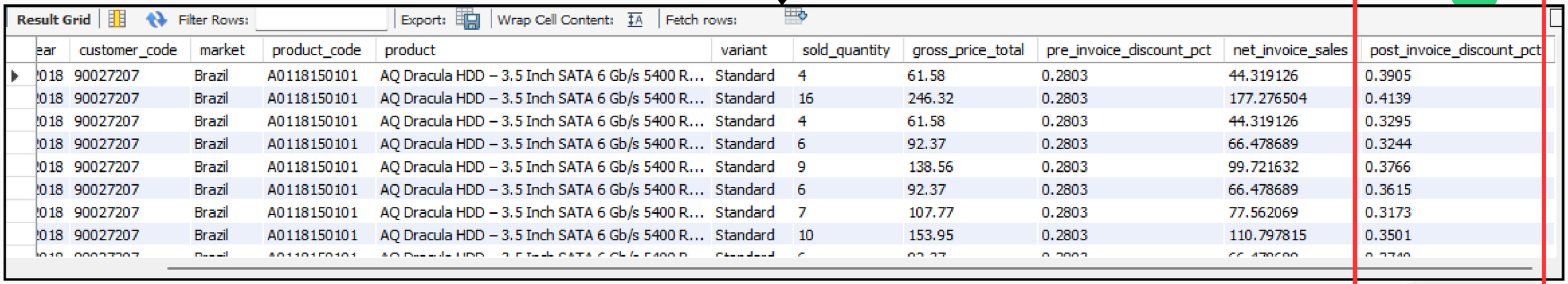
VIEWS

SQL QUERY

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `sales_postinv_discount` AS
    SELECT
        `s`.`date` AS `date`,
        `s`.`fiscal_year` AS `fiscal_year`,
        `s`.`customer_code` AS `customer_code`,
        `s`.`market` AS `market`,
        `s`.`product_code` AS `product_code`,
        `s`.`product` AS `product`,
        `s`.`variant` AS `variant`,
        `s`.`sold_quantity` AS `sold_quantity`,
        `s`.`gross_price_total` AS `gross_price_total`,
        `s`.`pre_invoice_discount_pct` AS `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`))))
```



OUTPUT



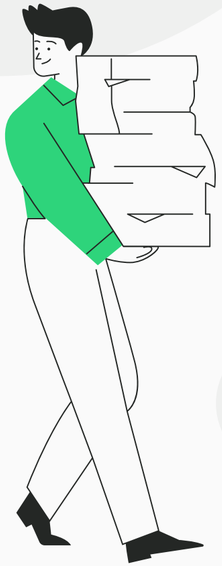
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ear	customer_code	market	product_code	product	variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3905
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	16	246.32	0.2803	177.276504	0.4139
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3295
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3244
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	9	138.56	0.2803	99.721632	0.3766
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3615
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	107.77	0.2803	77.562069	0.3173
2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	153.95	0.2803	110.797815	0.3501

Generate a report of top market net sales for a given financial year

VIEWS

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `net_sales` AS
    SELECT
        `sales_postinv_discount`.`date` AS `date`,
        `sales_postinv_discount`.`fiscal_year` AS `fiscal_year`,
        `sales_postinv_discount`.`customer_code` AS `customer_code`,
        `sales_postinv_discount`.`market` AS `market`,
        `sales_postinv_discount`.`product_code` AS `product_code`,
        `sales_postinv_discount`.`product` AS `product`,
        `sales_postinv_discount`.`variant` AS `variant`,
        `sales_postinv_discount`.`sold_quantity` AS `sold_quantity`,
        `sales_postinv_discount`.`gross_price_total` AS `gross_price_total`,
        `sales_postinv_discount`.`pre_invoice_discount_pct` AS `pre_invoice_discount_pct`,
        `sales_postinv_discount`.`net_invoice_sales` AS `net_invoice_sales`,
        `sales_postinv_discount`.`post_invoice_discount_pct` AS `post_invoice_discount_pct`,
        ((1 - `sales_postinv_discount`.`post_invoice_discount_pct`) * `sales_postinv_discount`.`net_invoice_sales`) AS `net_sales`
    FROM
        `sales_postinv_discount`
```



SQL QUERY

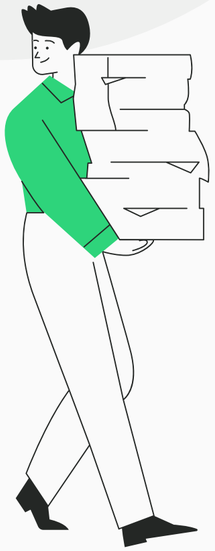
OUTPUT

product_code	product	variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct	net_sales
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3905	27.0125072970
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	16	246.32	0.2803	177.276504	0.4139	103.9017589944
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3295	29.7159739830
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3244	44.9130022884
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	9	138.56	0.2803	99.721632	0.3766	62.1664653888
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3615	42.4466429265
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	107.77	0.2803	77.562069	0.3173	52.9516245063
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	153.95	0.2803	110.797815	0.3501	72.0074999685

Generate a report for top 5 markets for fiscal year 2021 by using net sales

```sql
select
  market,
  round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales
where fiscal_year = 2021
group by market
order by net_sales_mln desc
limit 5;
```

SQL QUERY



OUTPUT

Result Grid | Filter Rows:

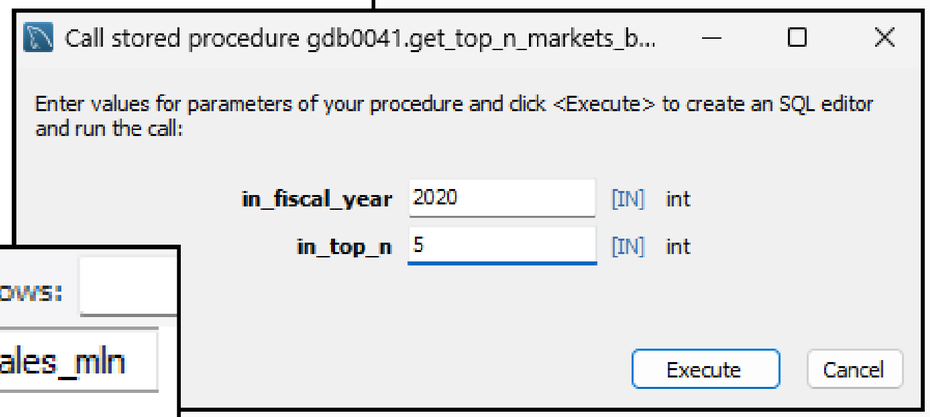
market	net_sales_mln
India	210.67
USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

Create a stored procedure for top_n_markets_by_net_sales for top_n market of any given fiscal year

STORED PROCEDURE

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_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
gdb0041.net_sales
where fiscal_year = in_fiscal_year
group by market
order by net_sales_mln desc
limit in_top_n;
END
```

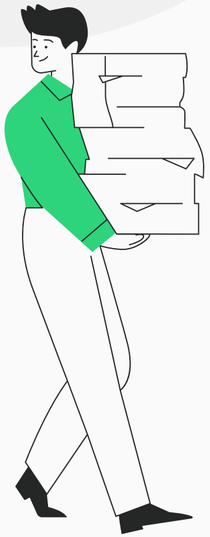
SQL QUERY



OUTPUT

Result Grid | Filter Rows:

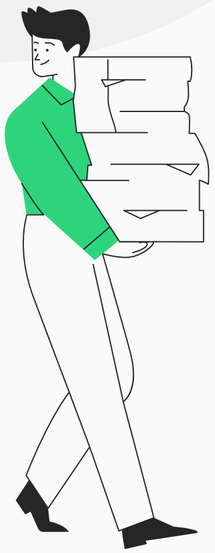
market	net_sales_mln
India	64.73
USA	46.35
South Korea	22.38
Philiphines	17.45
Canada	15.87



Create a report of op 5 customers for the fiscal year 2021

```sql
select
    c.customer,
    round(sum(net_sales)/1000000,2) as net_sales_mln
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_mln desc
limit 5;
```

SQL QUERY



OUTPUT

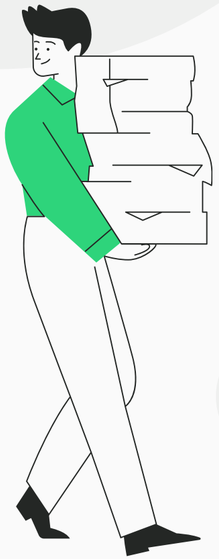
Result Grid | Filter Rows:

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

Generate a report of top product net sales for a given financial year

```sql
select
    product,
    round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales
where fiscal_year = 2021
group by product
order by net_sales_mln desc
limit 5;
```

SQL QUERY



OUTPUT

Result Grid | Filter Rows:

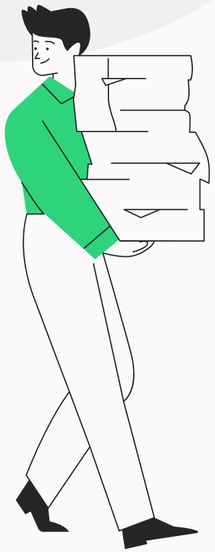
product	net_sales_mln
AQ BZ Allin1	33.75
AQ Qwerty	27.84
AQ Trigger	26.95
AQ Gen Y	23.58
AQ Maxima	22.32

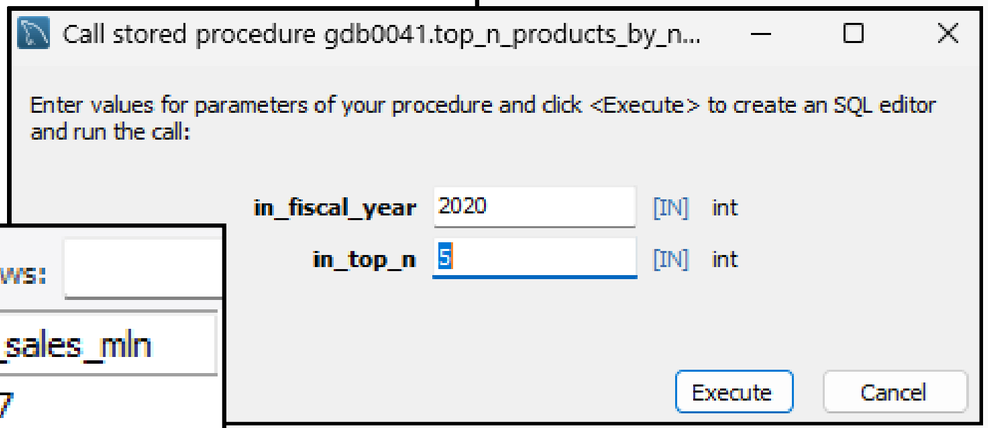
STORED PROCEDURE

Create a stored procedure to generate ‘top_n_products_by_net_sales’ of any market to extract top_n_products for any fiscal year

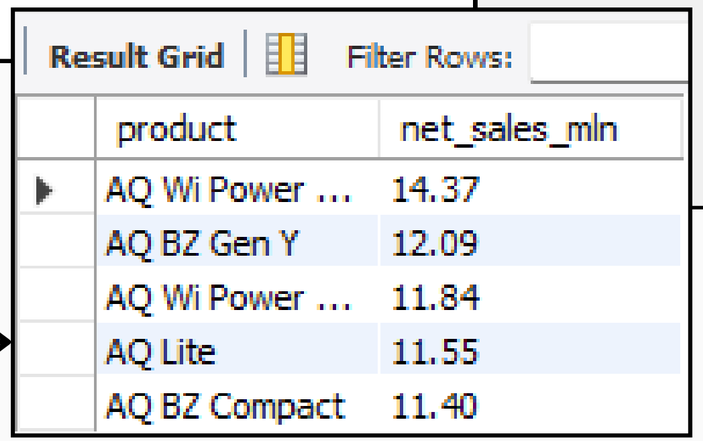
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `top_n_products_by_net_sales`(
    in_fiscal_year int,
    in_top_n int
)
BEGIN
select
    product,
    round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales
where fiscal_year = in_fiscal_year
group by product
order by net_sales_mln desc
limit in_top_n;
END
```

SQL QUERY





OUTPUT

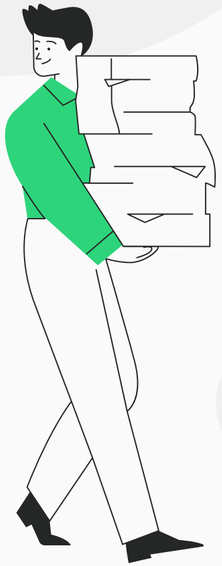


product	net_sales_mln
AQ Wi Power ...	14.37
AQ BZ Gen Y	12.09
AQ Wi Power ...	11.84
AQ Lite	11.55
AQ BZ Compact	11.40

Generate a bar chart for financial year 2021 for top 10 customers by % net sales

```sql
With cte1 as
(select
    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 s. fiscal_year = 2021
group by customer)
select
  *,
  net_sales_mln*100/sum(net_sales_mln) over() as pct_share_overall
  from cte1
order by net_sales_mln desc;
```

SQL QUERY

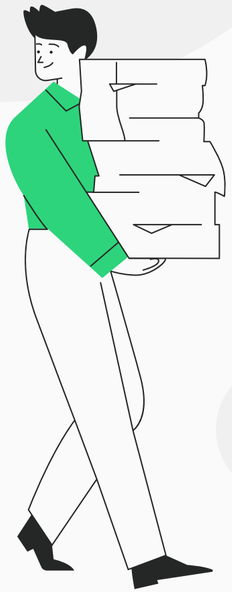


OUTPUT

Result Grid | Filter Rows: | Export:

customer	net_sales_mln	pct_share_overall
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

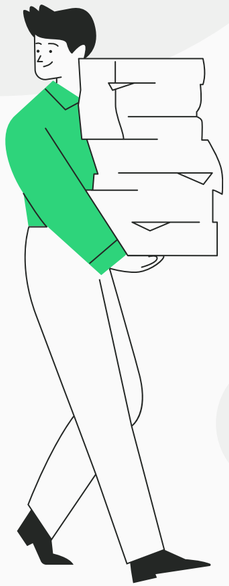
Generate a bar chart for financial year 2021 for top 10 customers by % net sales



Generate a report for financial year-2021 for top 10 customers by region wise % net sales

SQL QUERY

```sql
With cte1 as
(select
   c.customer,
   c.region,
   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 s. fiscal_year = 2021
group by c.customer, c.region)
select
  *,
  net_sales_mln*100/sum(net_sales_mln) over(partition by region) as pct_share_overall
  from cte1
order by region,net_sales_mln desc;
```

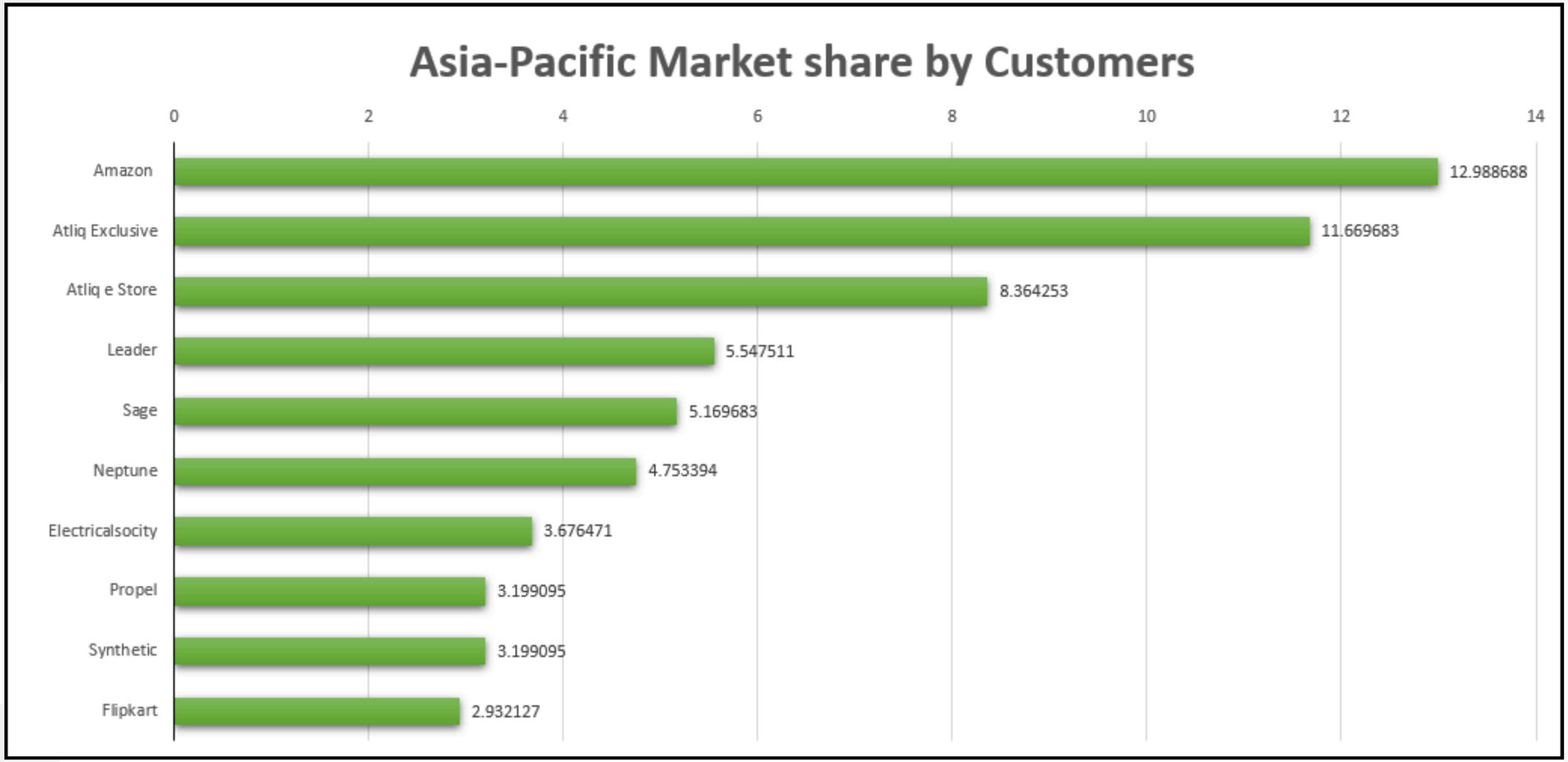


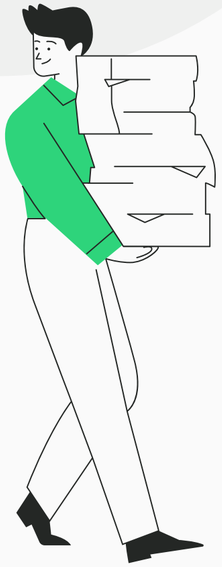
OUTPUT

Result Grid | Filter Rows: | Export:

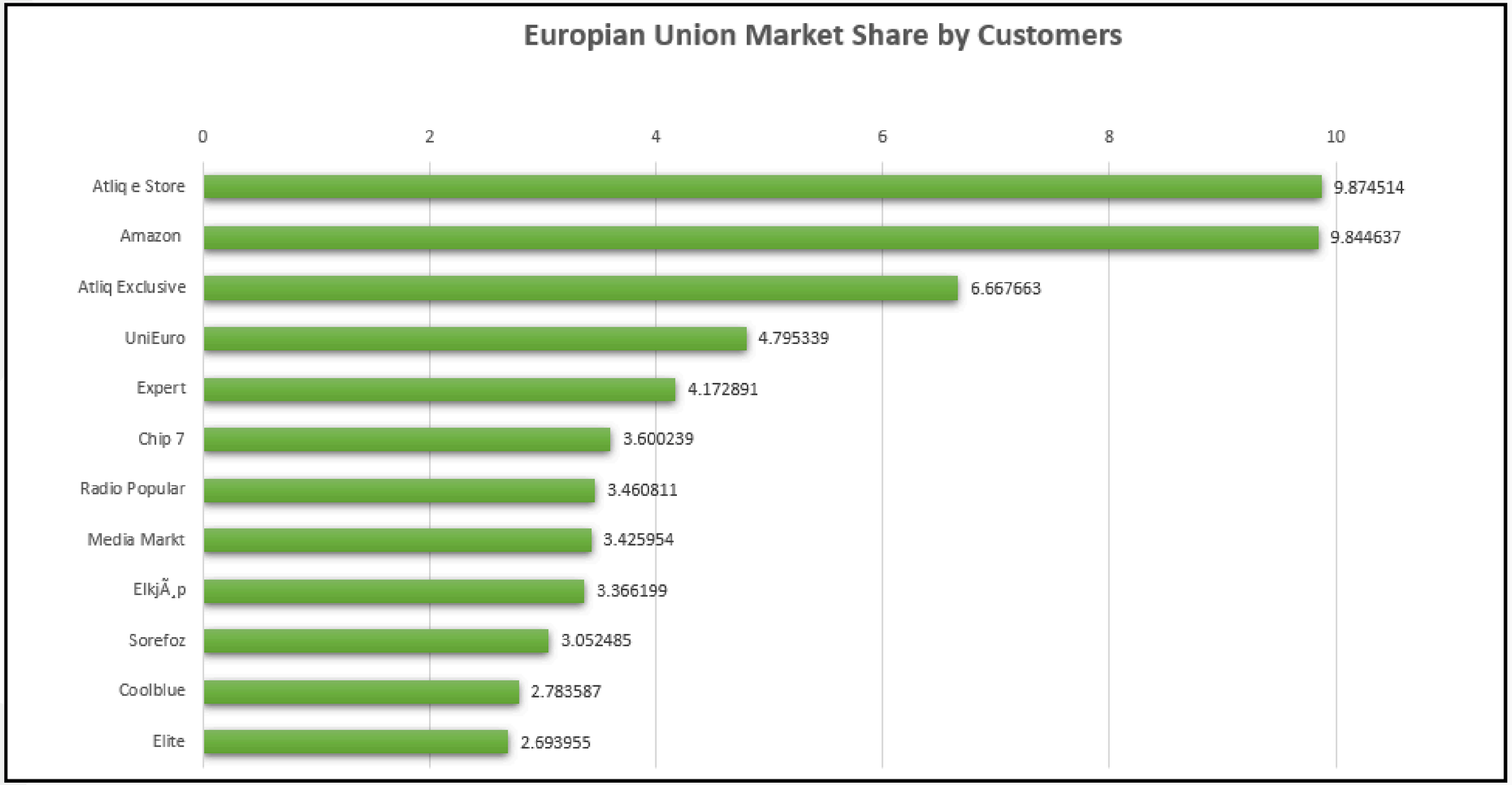
customer	region	net_sales_mln	pct_share_overall
Amazon	APAC	57.41	12.988688
Atliq Exclusive	APAC	51.58	11.669683
Atliq e Store	APAC	36.97	8.364253
Leader	APAC	24.52	5.547511
Sage	APAC	22.85	5.169683
Neptune	APAC	21.01	4.753394
Electricalsocity	APAC	16.25	3.676471
Propel	APAC	14.14	3.199095
Synthetic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

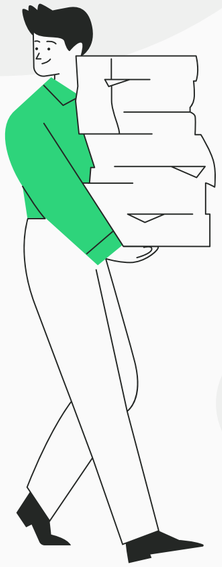
Generate a report for financial year-2021 for top 10 customers by region wise % net sales



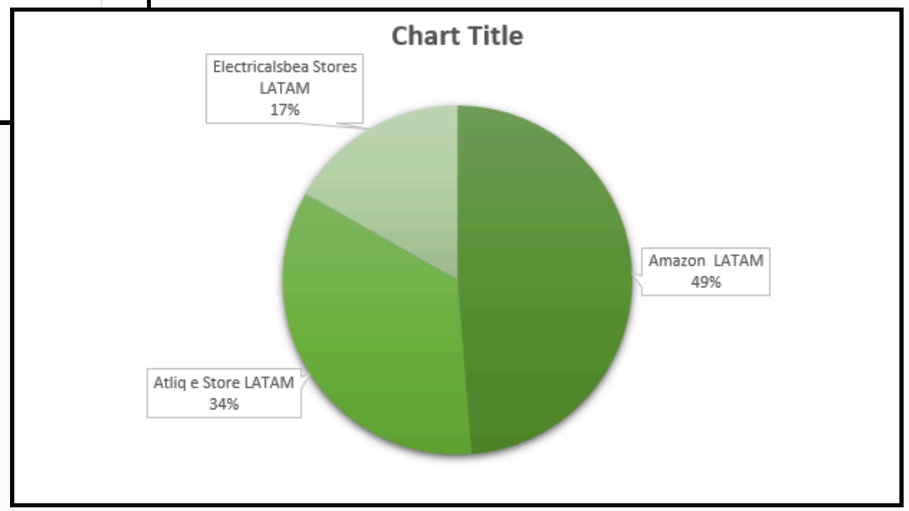


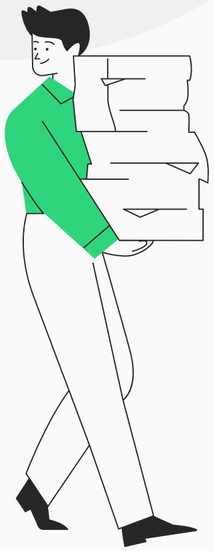
Generate a report for financial year-2021 for top 10 customers by region wise % net sales





Generate a report for financial year-2021 for top 10 customers by region wise % net sales

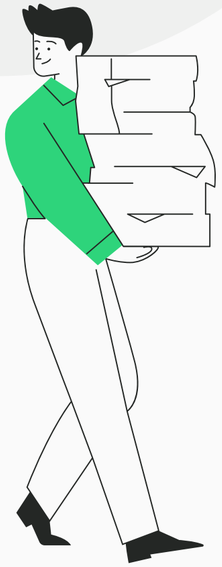




Generate forecast accuracy report for all the customers for a given fiscal year

```sql
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-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=2021
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;
```

SQL QUERY

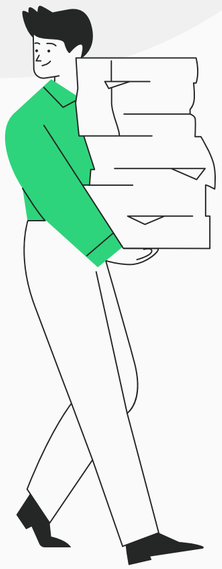


Generate forecast accuracy report for all the customers for a given fiscal year

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	23920	17.9	70402	52.72	47.28
70010048	Atliq e Store	Bangladesh	119439	142010	22526	15.9	75666	53.28	46.72
90025209	Electricalsbea Stores	Columbia	13178	15428	1892	12.3	8224	53.31	46.69
90023027	Costco	Canada	236189	279962	43752	15.6	149282	53.32	46.68
90023026	Relief	Canada	228988	273492	44486	16.3	146930	53.72	46.28
90017051	Forward Stores	Portugal	86823	118067	31138	26.4	63462	53.75	46.25
70027208	Atliq e Store	Brazil	33713	47321	13282	28.1	25458	53.80	46.20
90017058	Mbit	Portugal	86860	110195	23226	21.1	59364	53.87	46.13
90023028	walmart	Canada	239081	283323	44228	15.6	153044	54.02	45.98
90023024	Sage	Canada	246397	287233	40823	14.2	155597	54.17	45.83
90013124	Amazon	Italy	110898	136116	25168	18.5	73776	54.20	45.80
90015146	Mbit	Norway	147152	210507	63320	30.1	114154	54.23	45.77
90017054	Flawless Stores	Portugal	84371	114698	30241	26.4	62397	54.40	45.60
90015147	Chiptec	Norway	154897	223867	68934	30.8	122064	54.53	45.47
80001019	Neptune	China	1113979	1275248	161269	12.6	695779	54.56	45.44
90015144	Sound	Norway	160074	225637	65522	29.0	123216	54.61	45.39
90009130	Logic Stores	Newzealand	103290	110175	6837	6.2	60177	54.62	45.38
90017050	Electricalsara Stores	Portugal	85272	114688	29321	25.6	62665	54.64	45.36
70013125	Atliq Exclusive	Italy	101658	123428	21689	17.6	67465	54.66	45.34
90015149	UniEuro	Norway	142086	212500	70391	33.1	116149	54.66	45.34
90021088	Electricalslytical	United Kin...	224350	323689	99319	30.7	176955	54.67	45.33
90021094	Coolblue	United Kin...	208512	301367	92816	30.8	165004	54.75	45.25
70009134	Atliq e Store	Newzealand	103747	110791	6992	6.3	60674	54.76	45.24
90013118	Fnac-Darty	Italy	101847	126289	24362	19.3	69162	54.76	45.24
70017060	Atliq e Store	Portugal	89925	120744	30723	25.4	66189	54.82	45.18
90013122	Radio Popular	Italy	100746	123516	22692	18.4	67744	54.85	45.15
90017053	Info Stores	Portugal	84149	111740	27521	24.6	61303	54.86	45.14
90017059	Amazon	Portugal	87828	114154	26239	23.0	62633	54.87	45.13
70023031	Atliq Exclusive	Canada	234114	286297	52162	18.2	157150	54.89	45.11
90023025	Premium Stores	Canada	220808	266351	45523	17.1	146215	54.90	45.10
90023022	Nomad Stores	Canada	225182	264886	39686	15.0	145526	54.94	45.06
90002007	Girias	India	746226	778757	32531	4.2	427909	54.95	45.05
90001021	Taobao	China	259715	297960	38234	12.8	163912	55.01	44.99
90017049	Premium Stores	Portugal	82557	112228	29605	26.4	61759	55.03	44.97

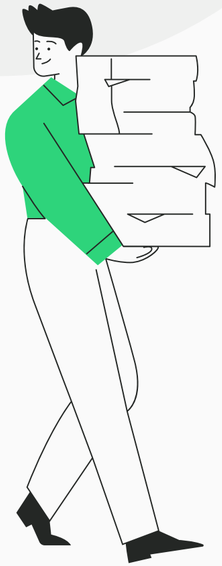
OUTPUT

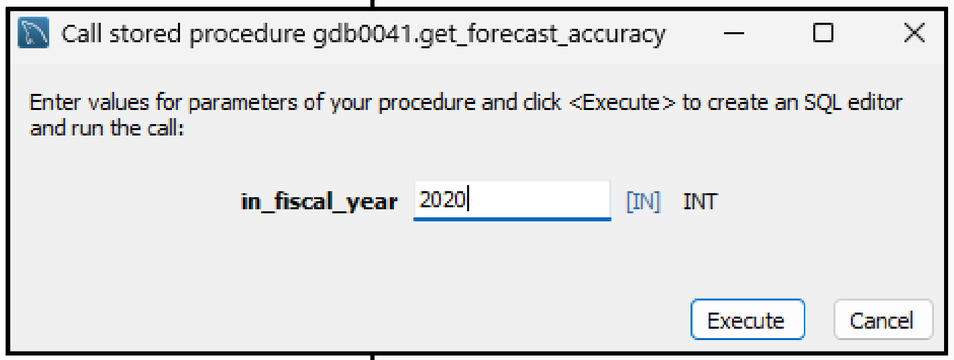


Generate a stored procedure to get forecast accuracy report for all the customers for any fiscal year

```sql
CREATE DEFINER=`root`@`localhost` 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-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
```

SQL QUERY



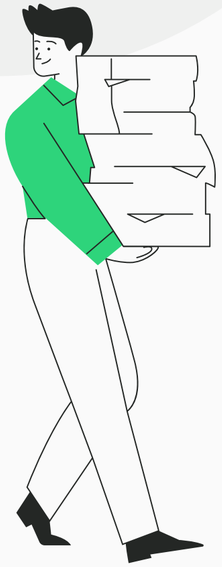


Generate a stored procedure to get forecast accuracy report for all the customers for any fiscal year

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

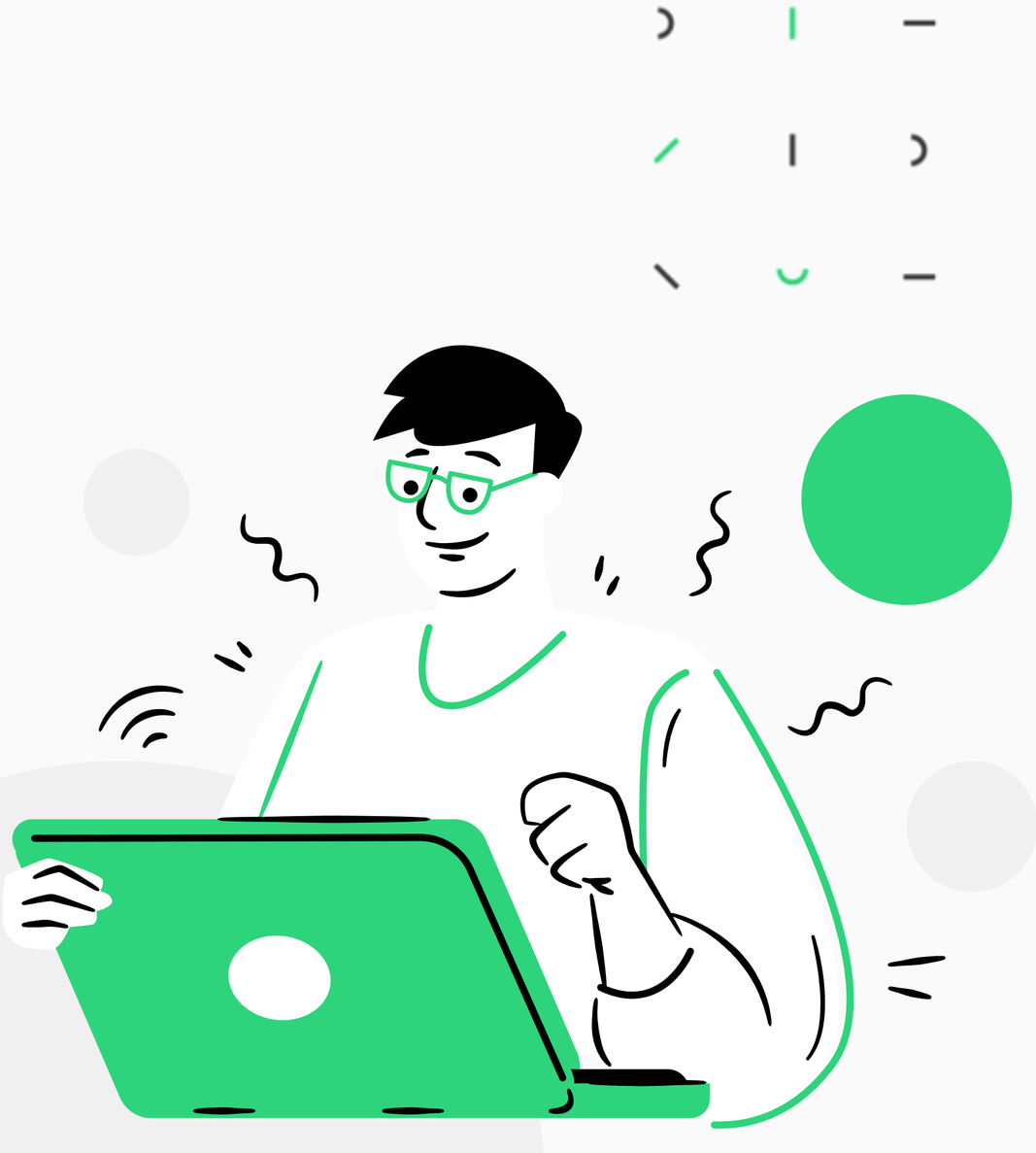
customer_code	customer_name	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_accuracy
90020099	Integration Stores	Austria	1843	2961	541	18.3	1343	45.36	54.64
90020100	Nova	Austria	1942	3000	502	16.7	1362	45.40	54.60
90020101	Euronics	Austria	1796	2727	417	15.3	1239	45.43	54.57
90020097	Atlas Stores	Austria	1889	2778	427	15.4	1289	46.40	53.60
90025209	Electricalsbea Stores	Columbia	1811	3312	872	26.3	1550	46.80	53.20
90020102	Fnac-Darty	Austria	2025	2866	415	14.5	1403	48.95	51.05
90020098	Electricalsquipo Stores	Austria	1880	2830	517	18.3	1405	49.65	50.35
90019200	Sorefoz	Sweden	4093	5973	1266	21.2	3184	53.31	46.69
90019201	Expert	Sweden	3708	5461	1214	22.2	3014	55.19	44.81
90019202	Argos (Sainsbury's)	Sweden	3956	5356	907	16.9	2995	55.92	44.08
90010046	Amazon	Banglad...	55532	58644	2919	5.0	33523	57.16	42.84
70006158	Atliq e Store	Philiphines	136991	155044	17933	11.6	88797	57.27	42.73
90023030	Amazon	Canada	127854	140248	12285	8.8	82479	58.81	41.19
70008170	Atliq e Store	Australia	178182	192586	14329	7.4	113633	59.00	41.00
90005161	Zone	Pakistan	68017	94196	25575	27.2	55837	59.28	40.72
90004062	Flawless Stores	Japan	16715	22166	4920	22.2	13164	59.39	40.61
90023026	Relief	Canada	85944	143041	56495	39.5	84953	59.39	40.61
70004069	Atliq Exclusive	Japan	15449	21225	5179	24.4	12643	59.57	40.43
90014140	Radio Popular	Netherl...	36344	62794	25571	40.7	37723	60.07	39.93
70011194	Atliq e Store	France	106020	115390	9197	8.0	69423	60.16	39.84
90010044	Surface Stores	Banglad...	36764	63634	25983	40.8	38353	60.27	39.73
90019203	Amazon	Sweden	5896	5776	-395	-6.8	3499	60.58	39.42
70020104	Atliq e Store	Austria	3027	2922	-351	-12.0	1773	60.68	39.32
90023024	Sage	Canada	83292	145672	61369	42.1	88451	60.72	39.28
90014137	Media Markt	Netherl...	32289	58838	25736	43.7	35752	60.76	39.24
90020103	Amazon	Austria	2865	2845	-257	-9.0	1729	60.77	39.23
90014138	Mbit	Netherl...	34832	59438	23803	40.0	36151	60.82	39.18
90014136	Reliance Digital	Netherl...	35477	62399	26011	41.7	38035	60.95	39.05
70014143	Atliq e Store	Netherl...	53780	60667	6456	10.6	36990	60.97	39.03
90024183	Electricalsbea Stores	Chile	3823	4767	356	7.5	2914	61.13	38.87
90010045	Control	Banglad...	34807	57994	22403	38.6	35489	61.19	38.81
90004066	Surface Stores	Japan	16931	21242	3728	17.6	13002	61.21	38.79

OUTPUT



Key Insights

- In FY-2022 AtliQ Hardware achieved the highest sales
- India was the top market with the highest net sales for FY-2021
- Amazon contributed 13.23% to the global market share in 2021 FY
- Net Sales % of Amazon was the highest in three out of four regions in 2021 FY



Connect with me

KARTHIK C
Data enthusiast

EMAIL
karthikc.0122@gmail.com