FINANCIAL ANALYTICS for ATLIQ Hardwares using MySQL

task:1

objective: retrieve customer information from the dim_customer table(i.e customer code for customer croma in india)
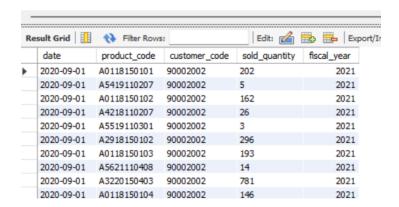
QUERY:

```
1 •    select *
2      from dim_customer
3      where customer like "%croma%"
4      and market ="india"
```

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 FUCTIONS

TASK:2

"get fiscal year" function for getting fiscal year(september=starting month) by passing calender date

```
1 • ⊖  CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_year`(
2        calendar_date DATE
3     └  ) RETURNS int
4          DETERMINISTIC
5   ⊖  BEGIN
6        declare fiscal_year int;
7        set fiscal_year=year(date_add(calendar_date,interval 4 month));
8     └  RETURN fiscal_year;
```

QUERY using this user defined function and output (for croma india sales during fiscal year 2021)

```
1 ●    select * from fact_sales_monthly
2      where
3      customer_code=90002002 and
4      get_fiscal_year(date)= 2021
5      order by date asc;
```

date	product_code	customer_code	sold_quantity	fiscal_year
2020-09-01	A0118150101	90002002	202	2021
2020-09-01	A5419110207	90002002	5	2021
2020-09-01	A0118150102	90002002	162	2021
2020-09-01	A4218110207	90002002	26	2021
2020-09-01	A5519110301	90002002	3	2021
2020-09-01	A2918150102	90002002	296	2021
2020-09-01	A0118150103	90002002	193	2021
2020-09-01	A5621110408	90002002	14	2021
2020-09-01	A3220150403	90002002	781	2021
2020-09-01	A0118150104	90002002	146	2021

TASK:3

"get_fiscal_quarter" user defined function to get quarter of year for which we want we want sales

```
1 ● ⊖  CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_quarter`(
2        calendar_date DATE
3      ⌐ ) RETURNS char(2) CHARSET utf8mb4
4            DETERMINISTIC
5   ⊖  BEGIN
6        declare qtr CHAR(2);
7        declare M tinyint;
8        SET M=month(calendar_date);
9   ⊖    case
10         when M in (9,10,11) then set qtr="q1";
11         when M in (12,01,02) then set qtr="q2";
12         when M in (03,04,05) then set qtr="q3";
13         when M in (06,07,08) then set qtr="q4";
14       end case;
15
16       RETURN qtr;
17       END
```

QUERY and Output:

```
•    SELECT s.date,s.product_code,s.sold_quantity,p.product,p.variant
     FROM gdb0041.fact_sales_monthly s
     join dim_product p
     on p.product_code=s.product_code
     where
     customer_code=90002002 and
     get_fiscal_year(date)=2021 and
     get_fiscal_quarter(date)="q4"
     order by date asc;
```

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

date	product_code	sold_quantity	product	variant
▶ 2021-06-01	A0118150101	205	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard
2021-06-01	A0118150102	78	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus
2021-06-01	A0118150103	48	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium
2021-06-01	A0118150104	126	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus
2021-06-01	A0219150201	40	AO WereWolf NAS Internal Hard Drive HDD – 8....	Standard

TASK 4:

Gross sales report: monthly product transactions

query

```
1 •  SELECT s.date,s.product_code,s.sold_quantity,p.product,p.variant,g.gross_price,
2    round(gross_price*s.sold_quantity,2) as gross_price_total
3    FROM gdb0041.fact_sales_monthly s
4    join dim_product p
5    on p.product_code=s.product_code
6    join fact_gross_price g
7    on g.product_code=s.product_code and g.fiscal_year = get_fiscal_year(s.date)
8
9    where
10   s.customer_code=90002002 and
11   get_fiscal_year(s.date)=2021 and
12   get_fiscal_quarter(s.date)="q4"
13   order by s.date asc;
```

output:

date	product_code	sold_quantity	product	variant	gross_price	gross_price_total
▶ 2021-06-01	A0118150101	205	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	19.0573	3906.75
2021-06-01	A0118150102	78	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	21.4565	1673.61
2021-06-01	A0118150103	48	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	21.7795	1045.42
2021-06-01	A0118150104	126	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	22.9729	2894.59
2021-06-01	A0219150201	40	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	23.6987	947.95
2021-06-01	A0219150202	102	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	24.7312	2522.58
2021-06-01	A0220150203	31	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	23.6154	732.08
2021-06-01	A0320150301	91	AQ Zion Saga	Standard	23.7223	2158.73
2021-06-01	A0321150302	70	AQ Zion Saga	Plus	27.1027	1897.19
2021-06-01	A0321150303	145	AQ Zion Saga	Premium	28.0059	4060.86
2021-06-01	A0418150103	108	AQ Mforce Gen X	Standard 3	19.5235	2108.54

TASK 5:

Gross Sales report: total sales amount

```
1 •   SELECT s.date,
2         round(sum(g.gross_price * s.sold_quantity),2) as gross_price_total
3         FROM gdb0041.fact_sales_monthly s
4         join dim_product p
5         on p.product_code=s.product_code
6         join fact_gross_price g
7         on g.product_code=s.product_code and g.fiscal_year = get_fiscal_year(s.date)
8
9         where
10        s.customer_code=90002002
11
```

date	gross_price_total
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
2018-10-01	440562.08
2018-12-01	653944.75
2019-01-01	359025.02

TASK6: Creating Stored Procedure("get monthly gross sales for any customer")

```
1 •⊖  CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customer`(
2             in_customer_codes TEXT
3         )
4  ⊖  BEGIN
5             SELECT
6                 s.date,
7                 SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
8             FROM fact_sales_monthly s
9             JOIN fact_gross_price g
10                ON g.fiscal_year=get_fiscal_year(s.date)
11                AND g.product_code=s.product_code
12            WHERE
13                FIND_IN_SET(s.customer_code, in_customer_codes) > 0
14            GROUP BY s.date
15            ORDER BY s.date DESC;
16      END
```

entering the customer code for which we want report

```
Call stored procedure gdb0041.get_monthly_gross_sales_...   —   ☐   ✕

Enter values for parameters of your procedure and click <Execute> to create an SQL editor
and run the call:

    in_customer_codes  90002002        [IN]  TEXT

                                              [Execute]   [Cancel]
```

```
1 •   call gdb0041.get_monthly_gross_sales_for_customer('90002002');
```

output:

date	monthly_sales
2021-12-01	19537146.58
2021-10-01	13908229.35
2021-09-01	11192823.18
2021-08-01	2349478.81
2021-06-01	2288587.49
2021-05-01	2181587.87
2021-04-01	2253575.01
2021-02-01	2355170.55
2021-01-01	2303086.42
2020-12-01	4078790.04
2020-10-01	3109316.96

TASK 7:

Creating Stored Procedure("MARKET BADGE for customer")

```sql
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;
# Determine Gold vs Silver status
        IF qty > 5000000 THEN
                SET out_level = 'Gold';
        ELSE
                SET out_level = 'Silver';
        END IF;
    END
```

entering the market and fiscal year for which we want to see performance of customer and his market badge

output:

```
1 •  set @out_level = '0';
2 •  call gdb0041.get_market_badge('india', 2019, @out_level);
3 •  select @out_level;
4
```


@out_level
Silver

TASK :8

Creating VIEW "Sales pre_invoice_discount "

```
1 •   CREATE        ALGORITHM = UNDEFINED        DEFINER = `root`@`localhost`
2         SQL SECURITY DEFINER
3     VIEW `sales pre_invoice discount 2` AS
4         SELECT
5             `s`.`date` AS `date`,
6             `s`.`fiscal_year` AS `fiscal_year`,
7             `s`.`customer_code` AS `customer_code`,
8             `p`.`product_code` AS `product_code`,
9             `p`.`product` AS `product`,
10            `p`.`variant` AS `variant`,
11            `g`.`gross_price` AS `gross_price`,
12            `s`.`sold_quantity` AS `sold_quantity`,
13            ROUND((`g`.`gross_price` * `s`.`sold_quantity`),
14                2) AS `gross_price_total`,
15            `pre`.`pre_invoice_discount_pct` AS `pre_invoice_discount_pct`
16        FROM
17            (((((`fact_sales_monthly` `s`
18            JOIN `dim_date` `dt` ON ((`dt`.`Calender_date` = `s`.`date`)))
19            JOIN `dim_product` `p` ON ((`p`.`product_code` = `s`.`product_code`)))
20            JOIN `fact_gross_price` `g` ON (((`g`.`product_code` = `s`.`product_code`)
21                AND (`g`.`fiscal_year` = `s`.`fiscal_year`))))
22            JOIN `fact_pre_invoice_deductions` `pre` ON (((`pre`.`customer_code` = `s`.`customer_code`)
23                AND (`pre`.`fiscal_year` = `s`.`fiscal_year`))))
```

using view for QUERY:

```
1 •   SELECT * FROM gdb0041.`sales pre_invoice discount 2`;
```

OUTPUT:

date	fiscal_year	customer_code	product_code	product	variant	gross_price	sold_quantity	gross_price_total	pre_invoice_discount_pct
2017-09-01	2018	70002017	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	51	785.16	0.0824
2017-09-01	2018	70002018	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	77	1185.43	0.2956
2017-09-01	2018	70003181	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	17	261.72	0.0536
2017-09-01	2018	70003182	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	6	92.37	0.2378
2017-09-01	2018	70006157	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	5	76.98	0.1057
2017-09-01	2018	70006158	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	7	107.77	0.1875
2017-09-01	2018	70007198	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	29	446.46	0.0700
2017-09-01	2018	70007199	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	34	523.44	0.2551
2017-09-01	2018	70008169	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	22	338.69	0.0953
2017-09-01	2018	70008170	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	5	76.98	0.1896
2017-09-01	2018	70011193	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	10	153.95	0.0521
2017-09-01	2018	70011194	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	4	61.58	0.2046
2017-09-01	2018	70012042	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	0	0.00	0.0984
2017-09-01	2018	70012043	A0118150101	AQ Dracula HDD – 3.5 Inch...	Standard	15.3952	0	0.00	0.2620

e discount 2 1 ×

TASK:9

NET Invoice sales using previous view

```
1 •    SELECT *,
2      (1-pre_invoice_discount_pct)*gross_price_total as net_invoice_sales
3        FROM gdb0041.`sales pre_invoice discount 2`
4
```

date	fiscal_year	customer_code	product_code	product	variant	gross_price	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
2017-09-01	2018	70002017	A0118150101	AQ Dracula ...	Standard	15.3952	51	785.16	0.0824	720.462816
2017-09-01	2018	70002018	A0118150101	AQ Dracula ...	Standard	15.3952	77	1185.43	0.2956	835.016892
2017-09-01	2018	70003181	A0118150101	AQ Dracula ...	Standard	15.3952	17	261.72	0.0536	247.691808
2017-09-01	2018	70003182	A0118150101	AQ Dracula ...	Standard	15.3952	6	92.37	0.2378	70.404414
2017-09-01	2018	70006157	A0118150101	AQ Dracula ...	Standard	15.3952	5	76.98	0.1057	68.843214
2017-09-01	2018	70006158	A0118150101	AQ Dracula ...	Standard	15.3952	7	107.77	0.1875	87.563125
2017-09-01	2018	70007198	A0118150101	AQ Dracula ...	Standard	15.3952	29	446.46	0.0700	415.207800
2017-09-01	2018	70007199	A0118150101	AQ Dracula ...	Standard	15.3952	34	523.44	0.2551	389.910456

TASK:10

Creating VIEW"Sales_post_invoice_discount"

```
1 •    CREATE             ALGORITHM = UNDEFINED        DEFINER = `root`@`localhost`
2          SQL SECURITY DEFINER
3      VIEW `sales_post_invoice_discount` AS
4          SELECT
5              `s`.`date` AS `date`,
6              `s`.`fiscal_year` AS `fiscal_year`,
7              `s`.`customer_code` AS `customer_code`,
8              `s`.`product_code` AS `product_code`,
9              `s`.`product` AS `product`,
10             `s`.`variant` AS `variant`,
11             `s`.`gross_price` AS `gross_price`,
12             `s`.`sold_quantity` AS `sold_quantity`,
13             `s`.`gross_price_total` AS `gross_price_total`,
14             `s`.`pre_invoice_discount_pct` AS `pre_invoice_discount_pct`,
15             ((1 - `s`.`pre_invoice_discount_pct`) * `s`.`gross_price_total`) AS `net_invoice_sales`,
16             (`po`.`discounts_pct` + `po`.`other_deductions_pct`) AS `post_invoice_discount_pct`
17         FROM
18             (`sales pre_invoice discount 2` `s`
19             JOIN `fact_post_invoice_deductions` `po` ON (((`s`.`date` = `po`.`date`)
20                 AND (`s`.`product_code` = `po`.`product_code`)
21                 AND (`s`.`customer_code` = `po`.`customer_code`)))))
```

QUERY AND OUTPUT

```sql
1 ●  SELECT sa.date,
2      sa.fiscal_year,
3      sa.customer_code,
4      sa.product_code,
5      sa.product,
6      sa.variant,
7      sa.sold_quantity,sa.gross_price_total,sa.pre_invoice_discount_pct,
8      (1-pre_invoice_discount_pct)*gross_price_total as net_invoice_sales,
9      (po.discounts_pct+po.other_deductions_pct) as post_invoice_discount_pct
10      FROM gdb0041.`sales pre_invoice discount 2` sa
11     join fact_post_invoice_deductions po
12     on sa.date=po.date and
13     sa.product_code=po.product_code and
14      sa.customer_code=po.customer_code
```

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

date	fiscal_year	customer_code	product_code	product	variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct
2017-09-01	2018	70002017	A0118150101	AQ Dracula ...	Standard	51	785.16	0.0824	720.462816	0.3379
2017-09-01	2018	70002018	A0118150101	AQ Dracula ...	Standard	77	1185.43	0.2956	835.016892	0.4013
2017-09-01	2018	70003181	A0118150101	AQ Dracula ...	Standard	17	261.72	0.0536	247.691808	0.3752
2017-09-01	2018	70003182	A0118150101	AQ Dracula ...	Standard	6	92.37	0.2378	70.404414	0.3446
2017-09-01	2018	70006157	A0118150101	AQ Dracula ...	Standard	5	76.98	0.1057	68.843214	0.3065
2017-09-01	2018	70006158	A0118150101	AQ Dracula ...	Standard	7	107.77	0.1875	87.563125	0.3587
2017-09-01	2018	70007198	A0118150101	AQ Dracula ...	Standard	29	446.46	0.0700	415.207800	0.3343
2017-09-01	2018	70007199	A0118150101	AQ Dracula ...	Standard	34	523.44	0.2551	389.910456	0.4168
2017-09-01	2018	70008169	A0118150101	AQ Dracula ...	Standard	22	338.69	0.0953	306.412843	0.3129
2017-09-01	2018	70008170	A0118150101	AQ Dracula ...	Standard	5	76.98	0.1896	62.384592	0.3189
2017-09-01	2018	70011193	A0118150101	AQ Dracula ...	Standard	10	153.95	0.0521	145.929205	0.2301

FINAL TASK

[NET SALES]- query and result

```sql
1 ●  select *,
2      round((1-post_invoice_discount_pct)*net_invoice_sales,3)as Net_Sales
3      from sales_post_invoice_discount
4
```

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

	fiscal_ye	customer_code	product_code	product	variant	gross_price	sold_quantity	gross_price_total	pre_invoice_discour	net_invoice_sales	post_invoice_discount	Net_Sales
09-01	2018	70008170	A0118150101	AQ Dra...	Standard	15.3952	5	76.98	0.1896	62.384592	0.3189	42.490
09-01	2018	70011193	A0118150101	AQ Dra...	Standard	15.3952	10	153.95	0.0521	145.929205	0.2301	112.351
09-01	2018	70011194	A0118150101	AQ Dra...	Standard	15.3952	4	61.58	0.2046	48.980732	0.2888	34.835
09-01	2018	70013125	A0118150101	AQ Dra...	Standard	15.3952	1	15.40	0.0587	14.496020	0.3190	9.872
09-01	2018	70013126	A0118150101	AQ Dra...	Standard	15.3952	1	15.40	0.2501	11.548460	0.3685	7.293
09-01	2018	70016178	A0118150101	AQ Dra...	Standard	15.3952	1	15.40	0.1937	12.417020	0.2738	9.017
09-01	2018	70022085	A0118150101	AQ Dra...	Standard	15.3952	20	307.90	0.2025	245.550250	0.2476	184.752
09-01	2018	70023031	A0118150101	AQ Dra...	Standard	15.3952	4	61.58	0.1008	55.372736	0.3288	37.166