Codebasics SQL Challenge

Here I have wrote complex queries for varies Business requirements
Requests:

1. Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region.

```
SELECT market,customer,region FROM gdb023.dim_customer
where customer="Atliq Exclusive" and region="APAC"
```

2. What is the percentage of unique product increase in 2021 vs. 2020? The final output contains these fields,
   unique_products_2020
   unique_products_2021
   percentage_chg

```
with cte1 as
(SELECT
count(DISTINCT CASE WHEN fiscal_year = 2020 THEN product_code END)
as unique_products_2020,
count(DISTINCT CASE WHEN fiscal_year = 2021 THEN product_code END) as
unique_products_2021
FROM
fact_sales_monthly)

SELECT
unique_products_2020,
unique_products_2021,
ROUND(((unique_products_2021 - unique_products_2020) / unique_products_2020) *
100, 2) as percentage_chg
FROM
CTE1;
```

3. Provide a report with all the unique product counts for each segment and sort them in descending order of product counts. The final output contains 2 fields,

   segment
   product_count

```
SELECT  segment,count(distinct(product_code)) as product_count from dim_product
group by segment
order by product_count desc
```

4. Follow-up: Which segment had the most increase in unique products in 2021 vs 2020? The final output contains these fields,

   segment
   product_count_2020
   product_count_2021
   difference

```
with cte1 as(
select p.segment,
count(distinct case when s.fiscal_year=2021 then s.product_code end) as
prod_count_2021,
count(distinct case when s.fiscal_year=2020 then s.product_code end) as
prod_count_2020

from fact_sales_monthly s
join dim_product p
on p.product_code=s.product_code
group by p.segment)

select *,(prod_count_2021-prod_count_2020) as diff
from cte1
```

5. Get the products that have the highest and lowest manufacturing costs. The final output should contain these fields,

   product_code product

manufacturing_cost

```sql
select m.product_code,p.product,m.manufacturing_cost
from fact_manufacturing_cost m
join dim_product p
on m.product_code=p.product_code
where m.manufacturing_cost=
(select max(manufacturing_cost)
from fact_manufacturing_cost)
or
m.manufacturing_cost=
(select min(manufacturing_cost)
from fact_manufacturing_cost)
order by manufacturing_cost desc
```

6. Generate a report which contains the top 5 customers who received an average high pre_invoice_discount_pct for the fiscal year 2021 and in the Indian market. The final output contains these fields,

   customer_code
   customer
   average_discount_percentage

```
select c.market, d.customer_code,avg(d.pre_invoice_discount_pct) as avg
FROM gdb023.fact_pre_invoice_deductions d
join dim_customer c
on c.customer_code=d.customer_code
where market="india" and fiscal_year=2021
group by c.market,d.customer_code
order by avg desc
limit 5
```

7.Get the complete report of the Gross sales amount for the customer **“Atliq Exclusive”** for each month**.** This analysis helps to get an idea of low and high-performing months and take strategic decisions.

The final report contains these columns:

Month
Year
Gross sales Amount

```
SELECT month(date) as Month,year(date) as Year,round(f.sold_quantity* g.gross_price,2)
as gross_sales_amt
FROM gdb023.fact_sales_monthly f
join fact_gross_price g
on g.product_code= f.product_code and
g.fiscal_year = f.fiscal_year
join dim_customer c
on c.customer_code=f.customer_code
where customer='Atliq Exclusive'
```

8.In which quarter of 2020, got the maximum total_sold_quantity? The final output contains these fields sorted by the total_sold_quantity,

Quarter
total_sold_quantity

```
with cte1 as

 (SELECT *,
  CASE
    WHEN MONTH(date) IN (9, 10, 11) THEN "Q1"
   WHEN MONTH(date) IN (12, 1, 2) THEN "Q2"
   WHEN MONTH(date) IN (3, 4, 5) THEN "Q3"
   WHEN MONTH(date) IN (6, 7, 8) THEN "Q4"
  END as quarters
FROM fact_sales_monthly)


select quarters,sum(sold_quantity) as total_sold_qty
from cte1
where fiscal_year=2020
group by quarters
order by total_sold_qty desc
limit 1
```

9.Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution? The final output contains these fields,

channel
gross_sales_mln
percentage

```
with cte1 as
  (select c.channel,round(sum(gross_sales)/1000000,2) as Gross_sales_mln
  from gross_sales g
  join dim_customer c
  on c.customer_code=g.customer_code
  where g.fiscal_year=2021
  group by c.channel
  order by  Gross_sales_mln desc)

 select *,round((Gross_sales_mln/(select sum(Gross_sales_mln) from cte1)) *100,2) as   pct
  from cte1
  group by channel
```

10. Get the Top 3 products in each division that have a high total_sold_quantity in the fiscal_year 2021? The final output contains these fields,

division product_code

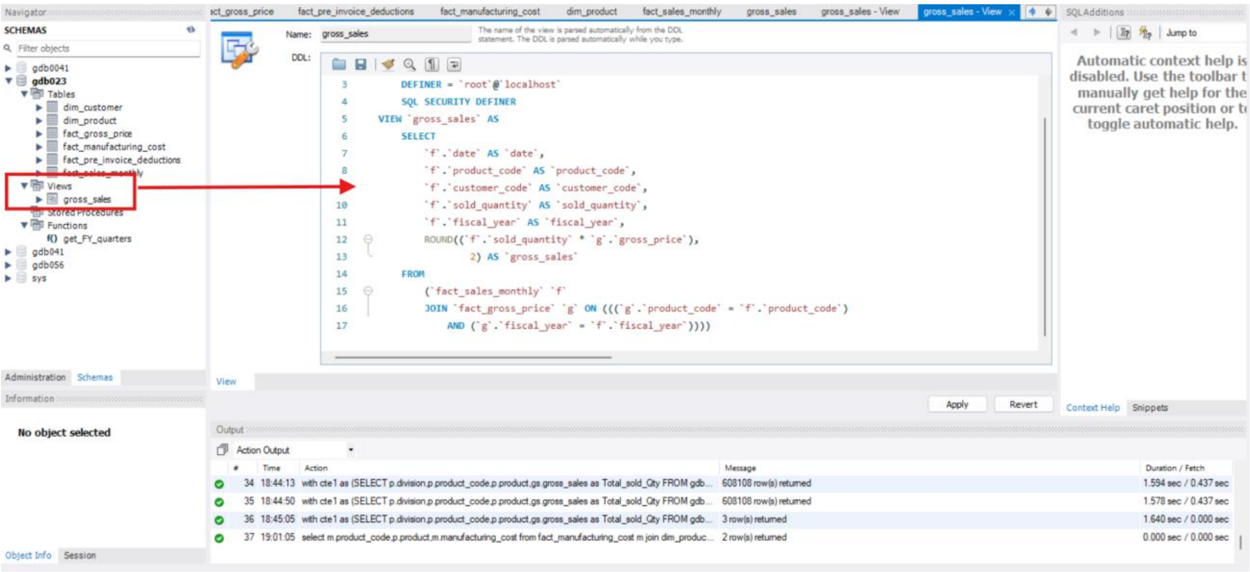
product
total_sold_quantity
rank_order

```
with cte1 as
(select
p.division,p.product_code,p.product,gs.gro
ss_sales as Total_sold_Qty
FROM gdb023.dim_product p
join gross_sales gs
on p.product_code=gs.product_code
where gs.fiscal_year=2021)
```

```
select * , dense_rank() over(partition by division order by Total_sold_Qty desc)
as rank_order
from cte1
limit 3
```

Created Views

(For query re-usability for Gross_sold_price).



Name: gross_sales

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

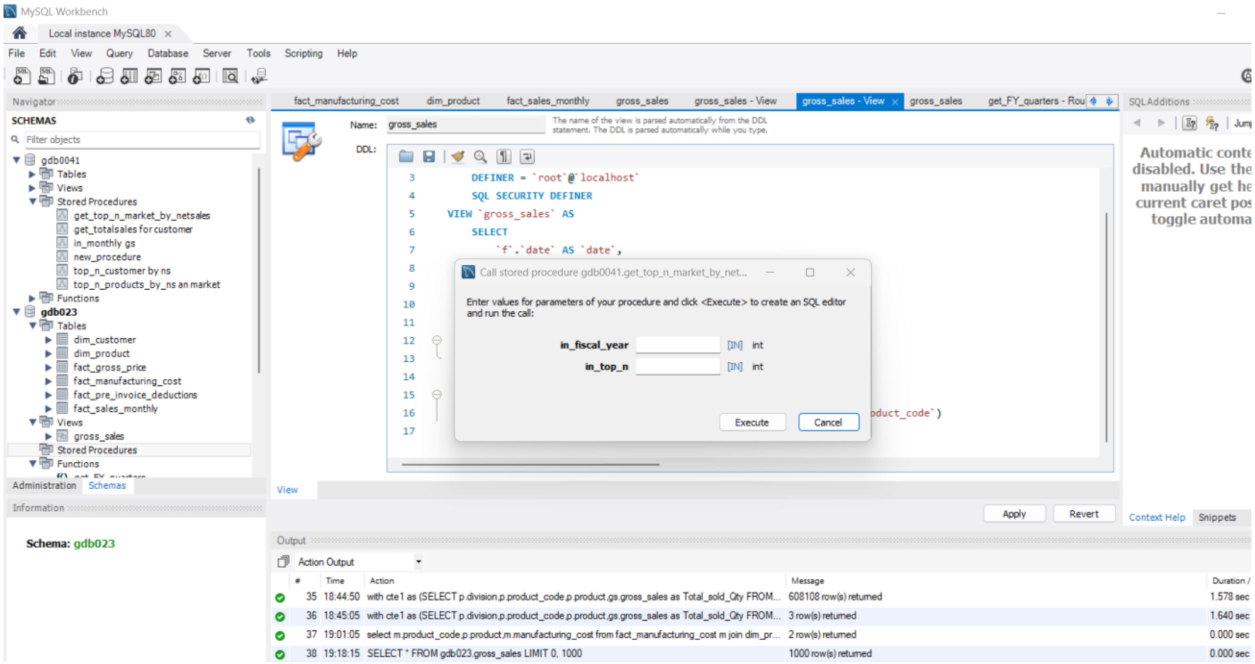
```
        DEFINER = `root`@`localhost`
        SQL SECURITY DEFINER
    VIEW `gross_sales` AS
        SELECT
            `f`.`date` AS `date`,
            `f`.`product_code` AS `product_code`,
            `f`.`customer_code` AS `customer_code`,
            `f`.`sold_quantity` AS `sold_quantity`,
            `f`.`fiscal_year` AS `fiscal_year`,
            ROUND((`f`.`sold_quantity` * `g`.`gross_price`),
                    2) AS `gross_sales`
        FROM
            (`fact_sales_monthly` `f`
            JOIN `fact_gross_price` `g` ON (((`g`.`product_code` = `f`.`product_code`)
                AND (`g`.`fiscal_year` = `f`.`fiscal_year`))))
```

Output

Action Output

```
    #   Time      Action                                                                              Message                  Duration / Fetch
✓   34  18:44:13  with cte1 as (SELECT p.division.p.product_code.p.product.gs.gross_sales as Total_sold_Qty FROM gdb...  608108 row(s) returned   1.594 sec / 0.437 sec
✓   35  18:44:50  with cte1 as (SELECT p.division.p.product_code.p.product.gs.gross_sales as Total_sold_Qty FROM gdb...  608108 row(s) returned   1.578 sec / 0.437 sec
✓   36  18:45:05  with cte1 as (SELECT p.division.p.product_code.p.product.gs.gross_sales as Total_sold_Qty FROM gdb...  3 row(s) returned        1.640 sec / 0.000 sec
✓   37  19:01:05  select m.product_code.p.product,m.manufacturing_cost from fact_manufacturing_cost m join dim_produc...  2 row(s) returned        0.000 sec / 0.000 sec
```

Created Stored Procedures



Name: gross_sales

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
        DEFINER = `root`@`localhost`
        SQL SECURITY DEFINER
    VIEW `gross_sales` AS
        SELECT
            `f`.`date` AS `date`,
oduct_code`)
```

Call stored procedure gdb0041.get_top_n_market_by_net...

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

in_fiscal_year [IN] int

in_top_n [IN] int

Execute Cancel

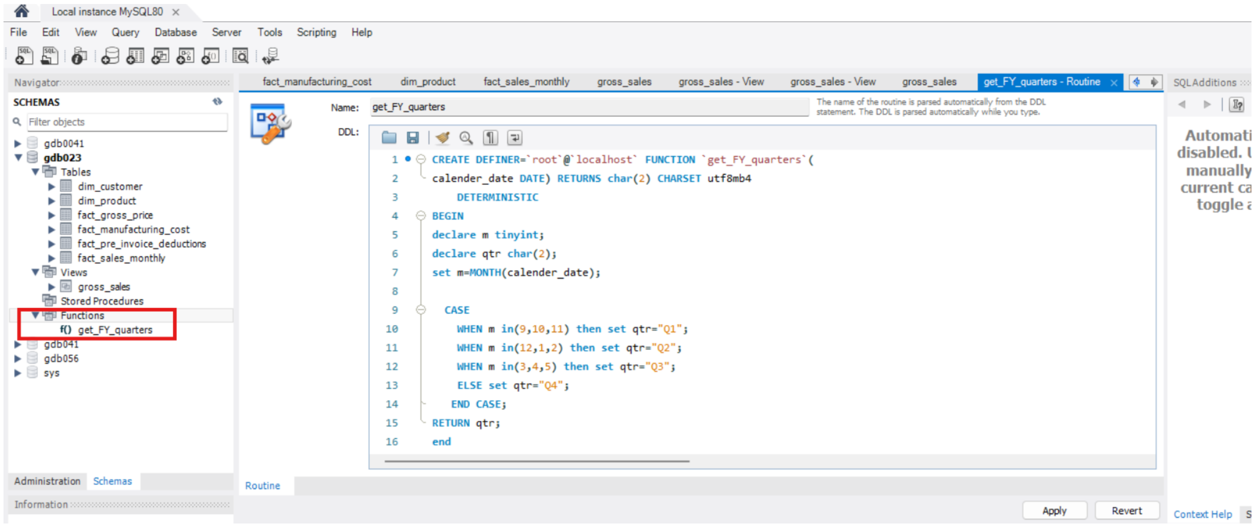
Output

Action Output

```
    #   Time      Action                                                                              Message                  Duration /
✓   35  18:44:50  with cte1 as (SELECT p.division.p.product_code.p.product.gs.gross_sales as Total_sold_Qty FROM...  608108 row(s) returned   1.578 sec
✓   36  18:45:05  with cte1 as (SELECT p.division.p.product_code.p.product.gs.gross_sales as Total_sold_Qty FROM...  3 row(s) returned        1.640 sec
✓   37  19:01:05  select m.product_code.p.product,m.manufacturing_cost from fact_manufacturing_cost m join dim_pr...  2 row(s) returned        0.000 sec
✓   38  19:18:15  SELECT * FROM gdb023.gross_sales LIMIT 0, 1000                                                    1000 row(s) returned     0.000 sec
```

Created User-Defined Functions

(For getting Quarters in Fiscal_year)



Name: get_FY_quarters

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```sql
CREATE DEFINER=`root`@`localhost` FUNCTION `get_FY_quarters`(
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
```

Thank