Data Analysis on Atliq's Hardware Dataset Using SQL.

ABOUT ATLIQ HARDWARE :

Atliq is a virtual enterprise specializing in the manufacturing and distribution of high-quality computer hardware products. Atliq delivers high-quality computer hardware solutions tailored to meet the needs of both businesses(**customers**) and individual consumers(**end users**).

Atliq offers two convenient platforms for customers to purchase its products:

1. **Brick-and-Mortar Stores** – Physical retail locations providing an in-person shopping experience.
2. **E-Commerce Platform** – A seamless online store for easy and accessible purchases.

Atliq operates through three key distribution channels to ensure efficient product availability:

1. **Retailers** – Authorized sellers providing products to end customers.
2. **Direct Sales** – Direct transactions between Atliq and consumers or businesses.
3. **Distributors** – Partners handling bulk distribution to various retail outlets.

ABOUT ATLIQ HARDWARE DATASET :

The **Atliq Hardware Dataset** consists of **nine tables** and follows the **Star Schema** model, ensuring efficient data organization and analysis. The schema is structured with a central fact table connected to multiple dimension tables, optimizing performance for analytical queries.

The Atliq Hardware Dataset follows a Star Schema and consists of two dimension tables and seven fact tables:

Dimension Tables:

1. dim_customer – Contains customer-related details.
2. dim_product – Stores product-related information.

Fact Tables:

1. fact_forecast_month – Holds monthly sales forecasts.
2. fact_freight_cost – Tracks shipping and transportation costs.
3. fact_gross_price – Contains gross product pricing data.
4. fact_manufacturing_cost – Stores production-related cost details.

5. fact_post_invoice_deduction – Captures deductions applied after invoicing.
6. fact_pre_invoice_deduction – Records deductions applied before invoicing.
7. fact_sales_monthly – Contains monthly sales transaction data.

This structure enables efficient data analysis, ensuring smooth reporting and business insights.

Task 1: Gross Sales Report : Individual Product Transaction

Filters:

- Customer: Croma
- Market: India
- Fiscal Year: 2021 (Fiscal year in Atliq's Hardware starts from September)

Report Output:

- Month
- Product Name
- Variant
- Sold Quantity
- Gross Price Per Item
- Gross Price Total

To accomplish this task four tables are used:

1. fact_sales_monthly (month, product_code, sold_quantity)
2. fact_gross_price (gross_price)
3. dim_product (product_name, variant)
4. dim_customer(customer_code, customer_name, market)

To solve this task the very first thing I did was to convert the calendar year into a fiscal year. For this I created user defined functions:

1. get_fiscal_year
2. get_fiscal_month
3. gst_fiscal_quarter

Function 1. get_fiscal_year

```
CREATE FUNCTION `get_fiscal_year`(
Calender_Date DATE)
RETURNS year
```

```
   DETERMINISTIC
BEGIN
   DECLARE fiscal_year YEAR;
   SET fiscal_year = YEAR(DATE_ADD(Calender_Date, Interval 4 month));
   RETURN fiscal_year;
END
```

Function 2. get_fiscal_month

```
CREATE FUNCTION `get_fiscal_month`(
Calender_date DATE)
RETURNS int
   DETERMINISTIC
BEGIN
  Declare fiscal_month INT;
  SET fiscal_month = MONTH(DATE_ADD(Calender_date, interval 4 month));
  RETURN fiscal_month;
END
```

Function 3. get_fiscal_quarter

```
CREATE FUNCTION `get_fiscal_quarter`(
  Calender_date DATE )
RETURNS char(2)
   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";
     when m in (6,7,8) then SET qtr =  "Q4";
 end case;
RETURN qtr;
END
```

SQL QUERY:

```
use gdb0041;
select get_fiscal_month(f.date) as months, f.product_code, p.product, p.variant,
f.sold_quantity, g.gross_price, g.gross_price * f.sold_quantity as gross_price_total
from fact_sales_monthly f
left join dim_product p on f.product_code = p.product_Code
join fact_gross_price g
on
f.product_code = g.product_code
and  g.fiscal_year = get_fiscal_year(f.date)
where
get_fiscal_year(f.date) = 2021
and
customer_code = 90002002
order by months desc;
```

OUTPUT:

Task 1.csv

Task 2: Gross Sales Report : Total Sales Amount (Monthly)

In this task total sales amount in each month for fiscal year 2018 -2022 is determined for customer croma.

Filters:

- Customer : Croma

Report Output:

- date
- gross_price_total

To accomplish this task three tables are used:

- dim_customer (to get customer code)
- fact_sales_monthly (date , sold_quantity, product_code)
- fact_gross_price (product_code, fiscal_year, gross_price)

SQL QUERY:

```
use gdb0041;
select s.date, SUM(s.sold_quantity * g.gross_price) as gross_price_total
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;
```

OUTPUT:

Task 2.csv

Task 3. Yearly Sales Report

In this task total sales amount in each year for fiscal year 2018 -2022 is determined for customer croma.

Filters:

- Customer : Croma

Report Output:

- year
- gross_price_total

To accomplish this task three tables are used:

- dim_customer (to get customer code)
- fact_sales_monthly (date , sold_quantity, product_code)
- fact_gross_price (product_code, fiscal_year, gross_price)

SQL QUERY:

```
use gdb0041;
select get_fiscal_year(s.date) as year, sum(s.sold_quantity*g.gross_price) as gross_price_total
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 s.customer_code = 90002002
group by year;
```

OUTPUT:

Task 3.csv

Task 4. Gross Monthly Sales Report Using Stored Procedure

In this gross sales for a particular customer on a monthly basis is calculated. Here, for customer Amazon in India Market gross monthly sales report is prepared.

```
CREATE PROCEDURE `get_gross_monthlysales_for_customer`(
in_customer_code text )
BEGIN
select s.date, SUM(s.sold_quantity * g.gross_price) as gross_price_total
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 s.date
order by s.date;
END
```

SQL QUERY:

```
call gdb0041.get_gross_monthlysales_for_customer("90002008,90002016");
```

OUTPUT:

Task 4.csv

Task 5. Get Market Badge based on Sold Quantity

In this task if total sold quantity > 5000000 then that market for a particular fiscal year is Gold else Silver.

Report Input:

- Market
- Fiscal Year

Report Output:

- Market badge

To accomplish this task two tables are used:

- dim_customer (market, customer_code)
- fact_sales_monthly (date , sold_quantity, customer_code)

```
CREATE PROCEDURE `get_badge` (
IN in_market varchar(45),
IN in_fiscal_year year,
OUT out_badge varchar(45))
BEGIN
        declare total_quantity int default 0;
        if in_market = “” then
             set in_market = “India”;
        end if;
        select sum(sold_quantity) into total_quantity
        from dim_customer c
        join fact_sales_monthly s
                on c.customer_code = s.customer_code
        where
                c.market = in_market and get_fiscal_year(s.date) = in_fiscal_year
        group by c.market;

if total_quantity > 5000000 then
        set out_badge = "GOLD";
else
        set out_badge = "SILVER";
end if;
END
```

SQL QUERY:

```
set @out_badge = '0';
```

```
call gdb0041.get_badge('Indonessia', 2021, @out_badge);

select @out_badge;
```

OUTPUT:

Task 5.csv