**Part 1**

**Task 2: Adding New Data to orders and sales Tables**

1. **Sql Query Syntax Add New Data**

**Add New Data to Orders Table**

-- Add New Data Orders

INSERT INTO public.orders (order\_id,customer\_id,payment,order\_date,delivery\_date,payment\_status)

    VALUES (1006,1002,900,'2023-02-10','2023-02-10','Paid');

INSERT INTO public.orders (order\_id,customer\_id,payment,order\_date,delivery\_date,payment\_status)

    VALUES (1007,1005,900,'2023-02-15','2023-02-15','Paid');

INSERT INTO public.orders (order\_id,customer\_id,payment,order\_date,delivery\_date,payment\_status)

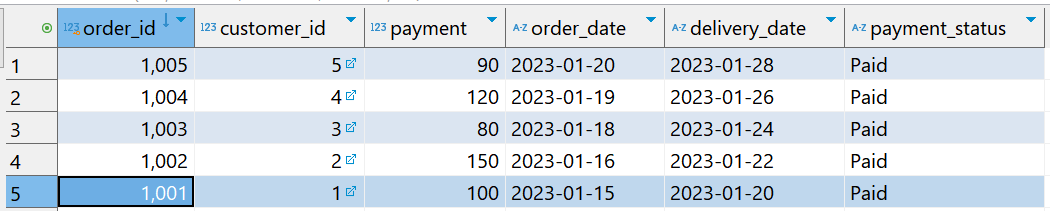
    VALUES (1008,1002,900,'2023-02-20','2023-02-20','Paid');

INSERT INTO public.orders (order\_id,customer\_id,payment,order\_date,delivery\_date,payment\_status)

    VALUES (1009,1003,900,'2023-02-25','2023-02-25','Paid');

INSERT INTO public.orders (order\_id,customer\_id,payment,order\_date,delivery\_date,payment\_status)

    VALUES (1010,1004,900,'2023-02-25','2023-02-25','Paid');



**Add New Data to Sales Table**

-- Add New Data Sales

INSERT INTO public.sales (sales\_id,order\_id,product\_id,price\_per\_unit,quantity,total\_price)

    VALUES (5005,1006,1260,90,10,900);

INSERT INTO public.sales (sales\_id,order\_id,product\_id,price\_per\_unit,quantity,total\_price)

    VALUES (5006,1007,1260,90,10,900);

INSERT INTO public.sales (sales\_id,order\_id,product\_id,price\_per\_unit,quantity,total\_price)

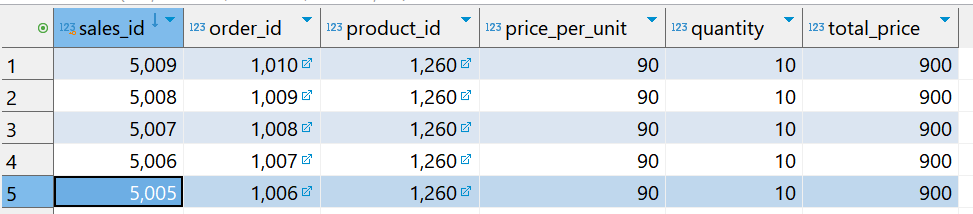
    VALUES (5007,1008,1260,90,10,900);

INSERT INTO public.sales (sales\_id,order\_id,product\_id,price\_per\_unit,quantity,total\_price)

    VALUES (5008,1090,1260,90,10,900);

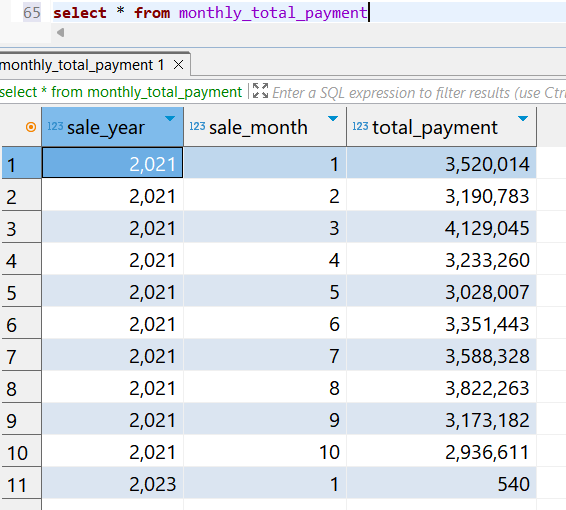
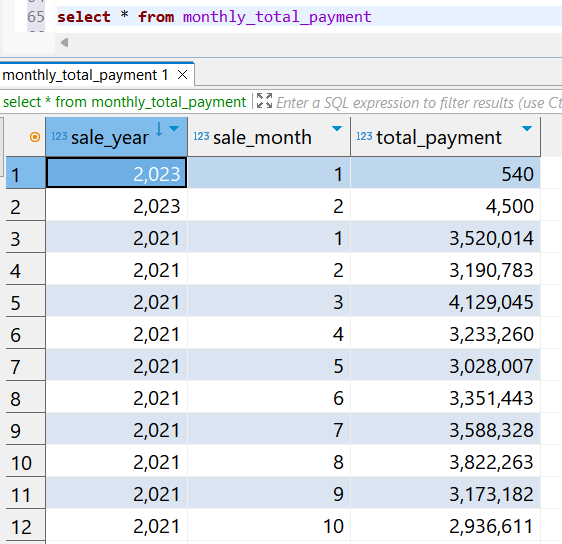
INSERT INTO public.sales (sales\_id,order\_id,product\_id,price\_per\_unit,quantity,total\_price)

    VALUES (5009,1010,1260,90,10,900);

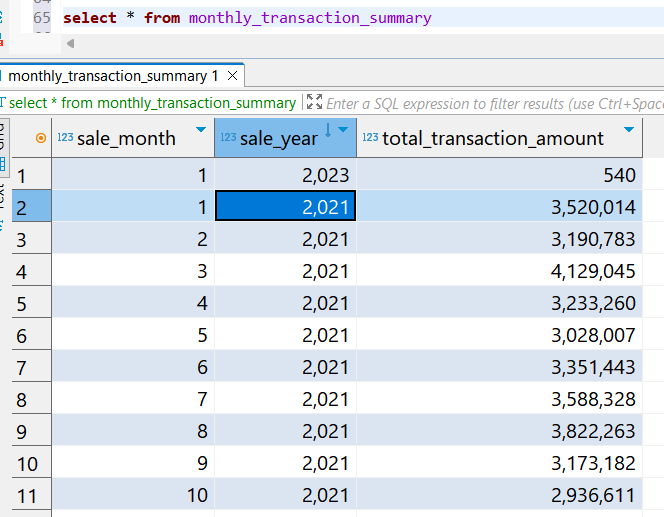
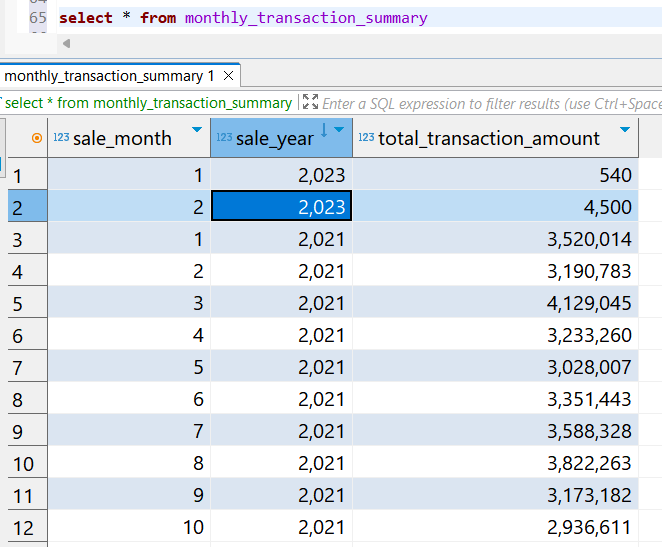


1. **Comparing montly total pament**

**select** \* **from** monthly\_total\_payment



1. **Description**
2. Run the monthly\_total\_payment query before adding new data.
3. Add 5 New data to order table and sales table with data orders for February 2023. Use the insert into query because the table already exists, just to insert additional data.
4. If new data has been added, run the monthly\_total\_payment checking query. Compare the results before and after adding the data.
5. It is known that there has been a change in data, where transactions for February 2023 have been entered into the monthly\_total\_payment record.