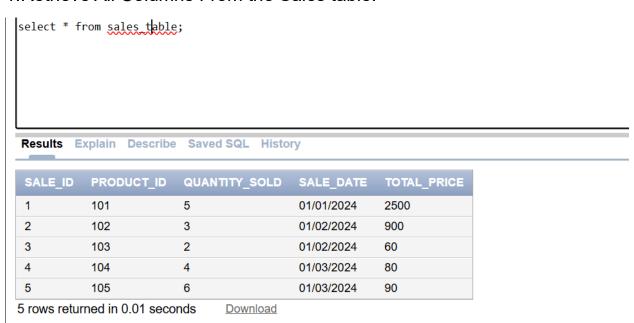
SQL LAB PRACTICE - 2

1.Retrieve All Columns From the Sales table.



2.Retrieve sale_id and quantity_sold from sales table;

select sale id quantity sold from sales table;

Results Explain Describe Saved SQL History

SALE_ID	QUANTITY_SOLD
1	5
2	3
3	2
4	4
5	6

5 rows returned in 0.00 seconds

Download

3.Retrieve the sale_id and sale_date from the sales table.

select <u>sale id.to char(sale date,'YYYY</u>-MM-DD') as <u>sale date</u> from <u>sales table</u>;

SALE_ID	SALE_DATE
1	2024-01-01

Results Explain Describe Saved SQL History

1	2024-01-01
2	2024-01-02
3	2024-01-02
4	2024-01-03
5	2024-01-03

5 rows returned in 0.00 seconds

Download

4. Filter the sales table to show only sales with a total_price greater than \$100.

select * from sales table where total price>100;

Results	Explain	Describe	Saved SQL	History

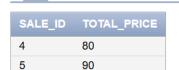
SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	01/01/2024	2500
2	102	3	01/02/2024	900

2 rows returned in 0.00 seconds Download

Results Explain Describe Saved SQL History

5.Retrieve the sale_id and total_price from the sales_table for sales made on January 3,2024.

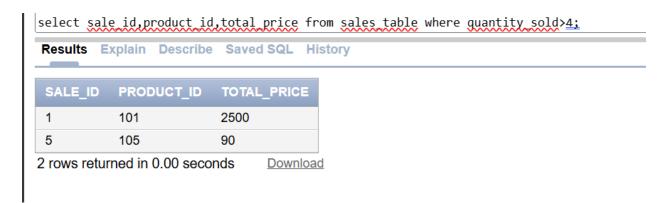
select sale id, total price from sales table where sale date = to date('2024-01-03','YYYY-MM-DD');



2 rows returned in 0.00 seconds

Download

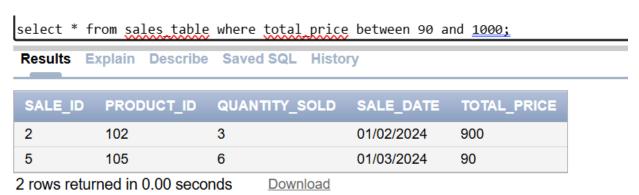
6.Retrieve the Sale_id,product_id and total_price from the sales table for sales with a quantity _sold greater than 4.



7.Retrieve all columns from the sales table those sale_id are 1,3,5.

select * f	rom sales tabl	e where sale id <u>in</u>	<u>(</u> 1,3,5 <u>);</u>	
Results E	Explain Describ	e Saved SQL Histo	ory	
SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	01/01/2024	2500
3	103	2	01/02/2024	60
5	105	6	01/03/2024	90
3 rows retu	rned in 0.00 sec	onds <u>Download</u>		

8.Retrieve all columns from the sales table those total_price between 90 and 1000.



9.Retrieve all coloumns from the sales table those total_price not between 90 and 100;

select * from sales table where total price not between 90 and 1000;

Results Explain Describe Saved SQL History

SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	01/01/2024	2500
3	103	2	01/02/2024	60
4	104	4	01/03/2024	80

³ rows returned in 0.00 seconds Download

10. Retrieve all columns from the sales table those sale_id are not in 1,3,5.

select * from sales table where sale id not in(1,3,5);

Results Explain Describe Saved SQL History

SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
2	102	3	01/02/2024	900
4	104	4	01/03/2024	80

² rows returned in 0.00 seconds Download

11. Update total price as 500 in the sales table those sale id are 1,3,5.

update sales table set total price = 500 where sale id in(1,3,5);

Results	ved SQL Hi	History
ıs	ived SQL mi	пізі

3 row(s) updated.

0.00 seconds

SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	2024-01-01	500
2	102	3	2024-01-02	900
3	103	2	2024-01-02	500
4	104	4	2024-01-03	80
5	105	6	2024-01-03	500

5 rows returned in 0.00 seconds Download

12.Delete from the sales table those total_price not between 90 and 1000

delete from sales table where total price not between 90 and 1000;

Results Explain Describe Saved SQL History

1 row(s) deleted.

0.00 seconds

SALE_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	2024-01-01	500
2	102	3	2024-01-02	900
3	103	2	2024-01-02	500
5	105	6	2024-01-03	500

4 rows returned in 0.00 seconds

Results Explain Describe Saved SQL History

Results Explain Describe Saved SQL History

Download

13. Sort all the records using sale_id column in ascending order.

select sale id product id quantity sold to char(sale date, 'YYYY-MM-DD') as sale date total price from sales table order by sale id;

Results	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	2024-01-01	500
2	102	3	2024-01-02	900
3	103	2	2024-01-02	500
5	105	6	2024-01-03	500
4 rows retu	rned in 0.00 seco	nds <u>Download</u>		

14. Sort all the records using sale_id column in descending order.

select sale id product id quantity sold to char(sale date, 'YYY' -MM-DD') as sale date, total price from sales table order by sale id desc:

SALE_ID PRODUCT_ID QUANTITY_SOLD SALE_DATE TOTAL_PRICE 6 5 105 2024-01-03 3 103 2024-01-02 500 2 102 2024-01-02 900 101 2024-01-01 4 rows returned in 0.00 seconds Download

15.Rename the sale_id Column as sales_id.

alter table sales table rename column sale id to sales id;

Results Explain Describe Saved SQL History

Table altered.

0.01 seconds

Results Explain Describe Saved SQL History

SALES_ID	PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
1	101	5	2024-01-01	500
2	102	3	2024-01-02	900
3	103	2	2024-01-02	500
5	105	6	2024-01-03	500

⁴ rows returned in 0.00 seconds Download

16.Drop the column Sales id.

alter table sales table drop column sales id;

Results Explain Describe Saved SQL History

Table altered.

0.08 seconds

Results	Explain	Describe	Saved SQL	History

PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
101	5	2024-01-01	500
102	3	2024-01-02	900
103	2	2024-01-02	500
105	6	2024-01-03	500

⁴ rows returned in 0.00 seconds Download

17.Rename The Table as tbl_sales.

rename sales table to tbl sales;

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

PRODUCT_ID	QUANTITY_SOLD	SALE_DATE	TOTAL_PRICE
101	5	2024-01-01	500
102	3	2024-01-02	900
103	2	2024-01-02	500
105	6	2024-01-03	500

⁴ rows returned in 0.00 seconds Download

18.Drop the table.

drop table tbl sales;

Results Explain Describe Save

Table dropped.

0.01 seconds