**Bahria University, Lahore Campus**

Department of Computer Sciences

Lab Journal 09

**(Fall 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Database Management System Lab** |  |
| Course Code: | CSL 220 | Max Marks: 25 |
| Faculty’s Name: | Maryam Munawar | Lab Engineer: |

Name: \_affan ahmad\_\_\_\_\_\_\_\_\_ Enroll No: \_\_03-134221-003\_\_\_\_\_

## Lab Task1

**Create the following table**

CREATE TABLE writer (

poet varchar(50) default NULL,

anthology varchar(40) default NULL,

copies\_in\_stock int null

default NULL

);



**Task 2**

**Insert the following values**

INSERT INTO writer VALUES

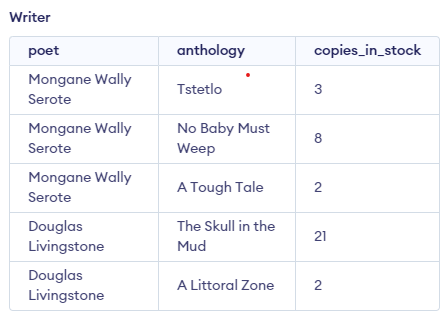
('Mongane Wally Serote','Tstetlo',3),

('Mongane Wally Serote', 'No Baby Must Weep',8),

('Mongane Wally Serote','A Tough Tale',2),

('Douglas Livingstone', 'The Skull in the Mud',21),

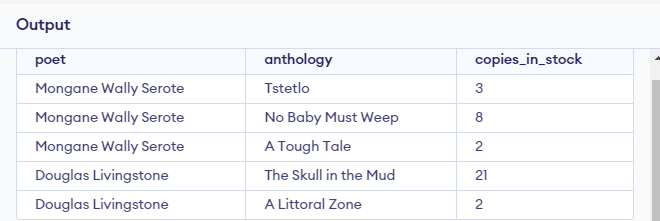
('Douglas Livingstone','A Littoral Zone',2);



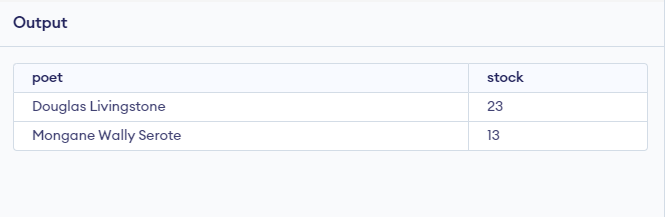
**Task 3**

**Practice the following queries**

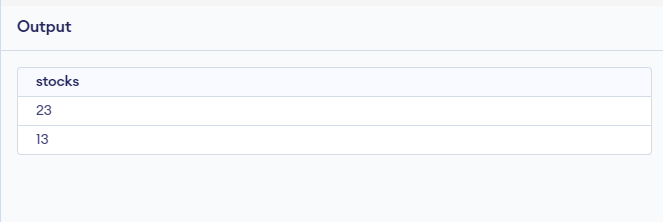
1. SELECT \* FROM writer;

v

1. SELECT poet,SUM(copies\_in\_stock) as stock FROM writer GROUP BY poet;



1. SELECT SUM(copies\_in\_stock) FROM writer GROUP BY poet;



1. SELECT poet,

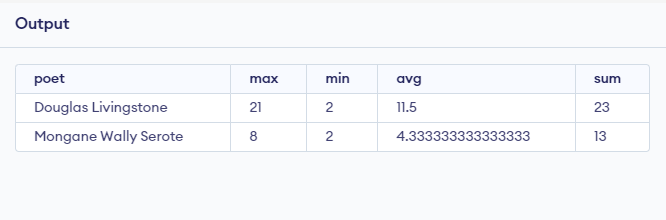
MAX(copies\_in\_stock) max,

MIN(copies\_in\_stock) min,

AVG(copies\_in\_stock) avg,

SUM(copies\_in\_stock) sum

FROM writer GROUP BY poet;



1. SELECT poet,

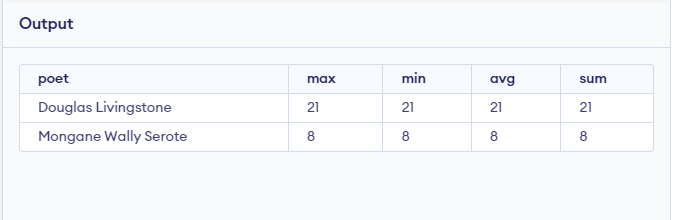
MAX(copies\_in\_stock) AS max,

MIN(copies\_in\_stock) AS min,

AVG(copies\_in\_stock) AS avg,

SUM(copies\_in\_stock) AS sum

FROM writer WHERE copies\_in\_stock > 5 GROUP BY poet;



1. SELECT poet,

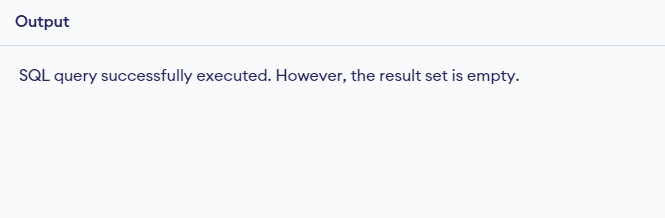
MAX(copies\_in\_stock) AS max,

MIN(copies\_in\_stock) AS min,

AVG(copies\_in\_stock) AS avg,

SUM(copies\_in\_stock) AS sum

FROM writer GROUP BY poet HAVING (copies\_in\_stock) > 5;

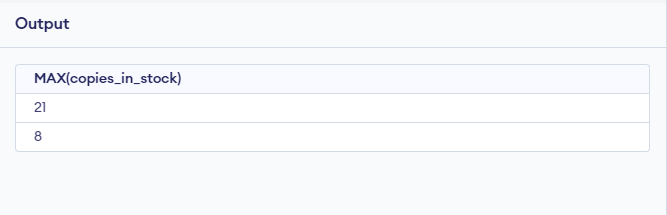


1. SELECT MAX(copies\_in\_stock)

FROM writer

GROUP BY poet

HAVING COUNT(copies\_in\_stock) > 1;



1. SELECT poet,

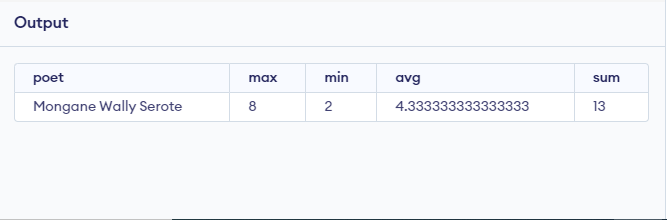
MAX(copies\_in\_stock) AS max,

MIN(copies\_in\_stock) AS min,

AVG(copies\_in\_stock) AS avg,

SUM(copies\_in\_stock) AS sum

FROM writer GROUP BY poet HAVING poet > 'E';



**Task 4**

**SQL CREATE VIEW Syntax**

CREATE VIEW view\_name AS

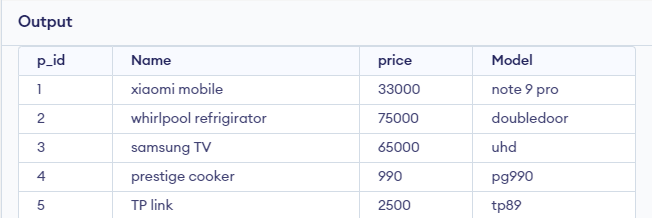
SELECT column\_name(s)

FROM table\_name WHERE condition

**SQL CREATE VIEW Examples**

If you have the Northwind database you can see that it has several views installed bydefault.

The view "Current Product List" lists all active products (products that are not Discontinued) from the "Products" table.



**The view is created with the following SQL:**

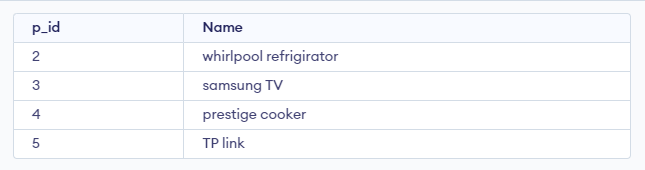
CREATE VIEW [Current Product List] AS

SELECT ProductID,ProductName

FROM Products WHERE Discontinued=No

**We can query the view above as follows:**

SELECT \* FROM [Current Product List]



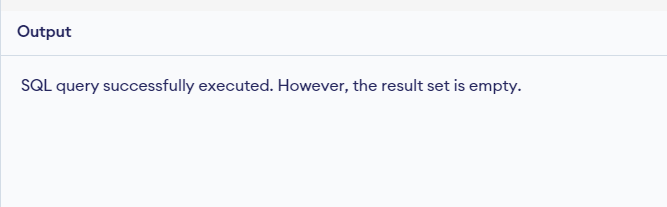
Another view in the Northwind sample database selects every product in the"Products" table with a unit price higher than the average unit price:

CREATE VIEW [Products Above Average Price] AS

SELECT ProductName,UnitPrice

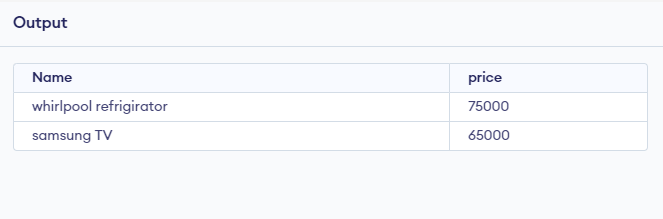
FROM Products

WHERE UnitPrice>(SELECT AVG(UnitPrice) FROM Products)



**We can query the view above as follows:**

SELECT \* FROM [Products Above Average Price]



Another view in the Northwind database calculates the total sale for each category in1997. Note that this view selects its data from another view called "Product Sales for1997":

CREATE VIEW [Category Sales For 1997] AS

SELECT DISTINCT CategoryName,Sum(ProductSales) AS CategorySales

FROM [Product Sales for 1997]

GROUP BY CategoryName

**We can query the view above as follows:**

SELECT \* FROM [Category Sales For 1997]

We can also add a condition to the query. Now we want to see the total sale only for the category "Beverages":

SELECT \* FROM [Category Sales For 1997]

WHERE CategoryName='Beverages'

**Task 5**

**Write a short note on this lecture. What did you learn ?**

In this journal I learn the group by and view , the purpose of group by is to  divides a table into sets and the purpose of  Views can be used as security mechanisms by letting users access data through the view, without granting the users permissions to directly access the underlying base tables of the view.

**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 5 |  |  |
| 2. | 5 |  |  |
| 3. | 5 |  |  |
| 4. | 5 |  |  |
| 5. | 5 |  |  |
| **Total** | **25** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Lab. Instructor**