INFO 2312 – Midterm 2

Date: Mar 31st, 2023

Time: 2 Hours 30 Minutes

Marks: 35 Marks

- **The Midterm 2 consists of 2 sections.**
- **One is creating the Database with 2 tables and the another is answering queries based on the created table. Submit the file after finishing the question on Moodle link for Midterm 2.**
- **The exam is open book and slides. Laptop with PostgreSQL is allowed.**
- **No copying code from the internet is allowed. Postgres documentation can be used.**
- **Separate spaces provided for query and screenshots of the answer.**
- **Not all questions need screenshots. Only answer what is asked in the question. If you do not find the box to paste the screenshot, then it is not needed.**

Consider a Superstore Database which consists of 3 tables, Orders, Returns and Managers. The CSV files have been provided along with this DOC file in the Midterm 2 Link in the Moodle. Answer the questions as below:

1. Create the two tables with their SQL queries in a newly created Database Superstore and paste the code in text format below. Make sure to assign primary and foreign keys as needed. Be careful while using the Unique and NOT NULL constraints. Use them only when required. Use correct datatype for each attribute in an Entity. Use PG-ADMIN to create these tables, as the tables will be needed to answer questions in the question 2.

Here is the Orders Table Query:

```
CREATE TABLE orders(
    rowid int PRIMARY KEY,
    orderpriority varchar(100),
    discount REAL,
    unitprice REAL,
    shippingcost REAL,
    customerid INT,
    customername varchar(100),
    shipmode varchar(100),
    customersegment varchar(100),
    productcategory varchar(100),
    productsubcategory TEXT,
    productcontainer varchar(100),
    productbasemargin REAL,
    region varchar(100) REFERENCES managers(region),
    province varchar(100),
    city varchar(100),
    postalcode varchar(100),
    orderdate DATE,
    shipdate DATE,
    profit REAL,
    quantity INT,
    sales REAL,
    orderid INT
);
```

Write Create Returns Table Query: **[2 Marks]** create table returns(OrderID Int, Status VARCHAR(127));
Write Create Managers Table Query: **[2 Marks]** create table managers(Region varchar(127) primary key , Manager varchar(127));

2. Use the created table as in Question 1, solve the problems as mentioned below. You will have to import the respective CSV files of the above created tables as without them, it is impossible to solve the questions below. If you are not able to upload the files successfully, do not leave the query questions. Just write the query to the best of your knowledge. Do not copy. To be graded for the screenshot answer, you must upload the CSV properly and paste the resulting screenshot of the queries as asked.

(a) Write Query to Find out which Product Sub-Category has a sum of Shipping Cost to sum of Sales ratio > 0.03.

Write the Query in box below. **[4 Marks]**

```
select productsubcategory,
sum(shippingcost)/sum(sales) as ratio
from orders
group by productsubcategory
having sum(shippingcost)/sum(sales)> 0.03 ;
```

Paste the screenshot of a portion of the answer below. **[1 Marks]**

Data Output Messages Notifications

	productsubcategory text	ratio real
1	Paper	0.034859788

(b) Write Queries to add 2 columns ‘Return Status’ & ‘Days to Ship’ to the Orders table.

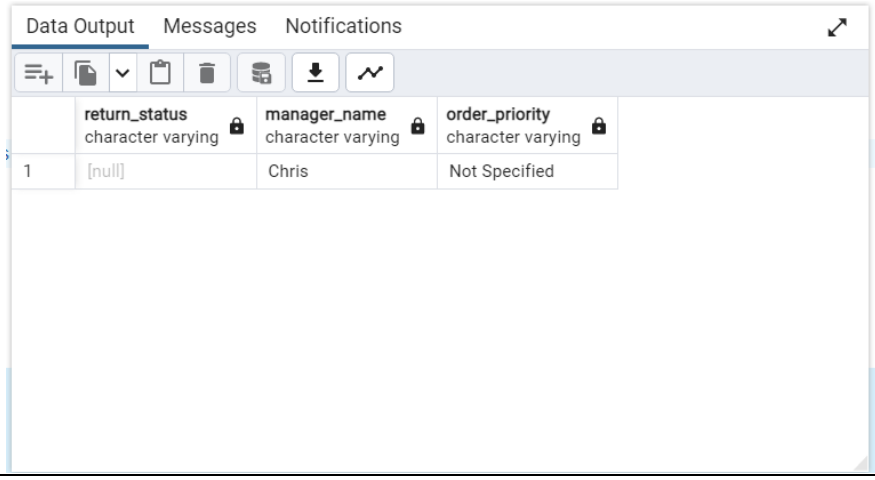
Write the Add column Queries in box below. **[3 Marks]**
alter table orders add column return_status varchar(127); alter table orders add column days_to_ship INT;
‘Return Status’ data comes from the Returns table, after adding the table use Update/CASE/WHEN/THEN to append a Boolean value True or False based on return status from the Returns table. Write Query below. **[3 Marks]**
update orders set return_status = case when returns.status = 'Returned' then orders.return_status = 'true' else 'false' end from returns where orders.orderid = returns.orderid;
‘Days to Ship’ is the difference between Order date and ship date column in the Orders table. Again, use Update to add data to the ‘Days to Ship’ Column **[2 Marks]**

```
update orders
set days_to_ship = shipdate - orderdate ;
```

(c) Create a Function which takes in OrderID column of Orders table as the input and returns back a table with columns return status of the order (True or False), name of the manager handling that Order and order priority of that order.

Write the Query in box below. **[5 Marks]**

```
create or replace function custom_function(order_id int)
returns table(
        return_status varchar(127),
        manager_name varchar(127),
        order_priority varchar(100)
)
AS $$
BEGIN
RETURN QUERY
select o.return_status, m.manager, o.orderpriority from orders o left join
managers m
on o.region = m.region
where o.orderid = order_id;
END; $$
LANGUAGE 'plpgsql';
```

Paste the screenshot for a sample OrderID input to the function. **[1 Marks]**
SELECT * FROM custom_function (88525); 

(d) Write a query to find out how much greater were the sales for the East region than for the South region?

Write the Query in box below. **[2 Marks]**
select sum(o.sales), m.region from orders o right join managers m on o.region = m.region group by m.region
Paste the screenshot of a portion of the answer below. **[1 Marks]**

Data Output Messages Notifications

	sum real	region [PK] character varying (127)
1	2.4228038e+06	East
2	1.5973472e+06	South
3	2.3914388e+06	West
4	2.5403458e+06	Central

(e) Write a query to find the contribution of total Sales by the 'Home Office' Customer Segment in the year 2012? For example, if in 2012, the total sum of sales across all Customer segments is 100 and 'Home Office' contributes 30 to the sum of sales. Then the answer would be 30/100, which is 0.3. *Hint: There are altogether 4 distinct Customer segments.*
The best way to solve this would be using subqueries in your SELECT.

Write the Query in box below. **[4 Marks]**

```
select sum(sales) / (select sum(sales) from orders where extract(year
from orderdate) = 2012) as contribution
,customersegment
from
orders
where extract(year from orderdate) = 2012
and customersegment = 'Home Office'
group by customersegment ;
```

Paste the screenshot of a portion of the answer below. **[1 Marks]**

Data Output Messages Notifications

	contribution real	customersegment character varying (100)
1	0.22937308	Home Office

(f) Write a query to find the total sales amount by Month for all the orders which are not returned?

Write the Query in box below. **[3 Marks]**

```
select sum(o.sales), extract(month from orderdate)
from orders o
where o.orderid
not in(select orderid from returns)
group by extract(month from orderdate);
```

Paste the screenshot of a portion of the answer below. **[1 Marks]**

Data Output Messages Notifications

	sum real	extract numeric
1	442572.6	2
2	787198.3	9
3	477970.16	3
4	574098.56	7
5	1.4000312e+06	11
6	462713.06	4
7	539531.9	6
8	1.0748128e+06	10
9	1.03141444e+06	12