

JENSON USA

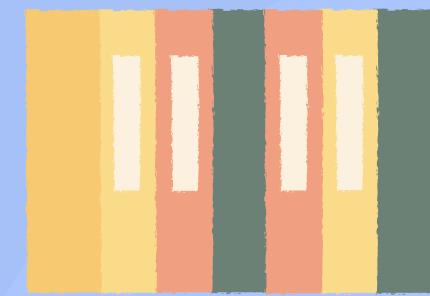
SQL PROJECT

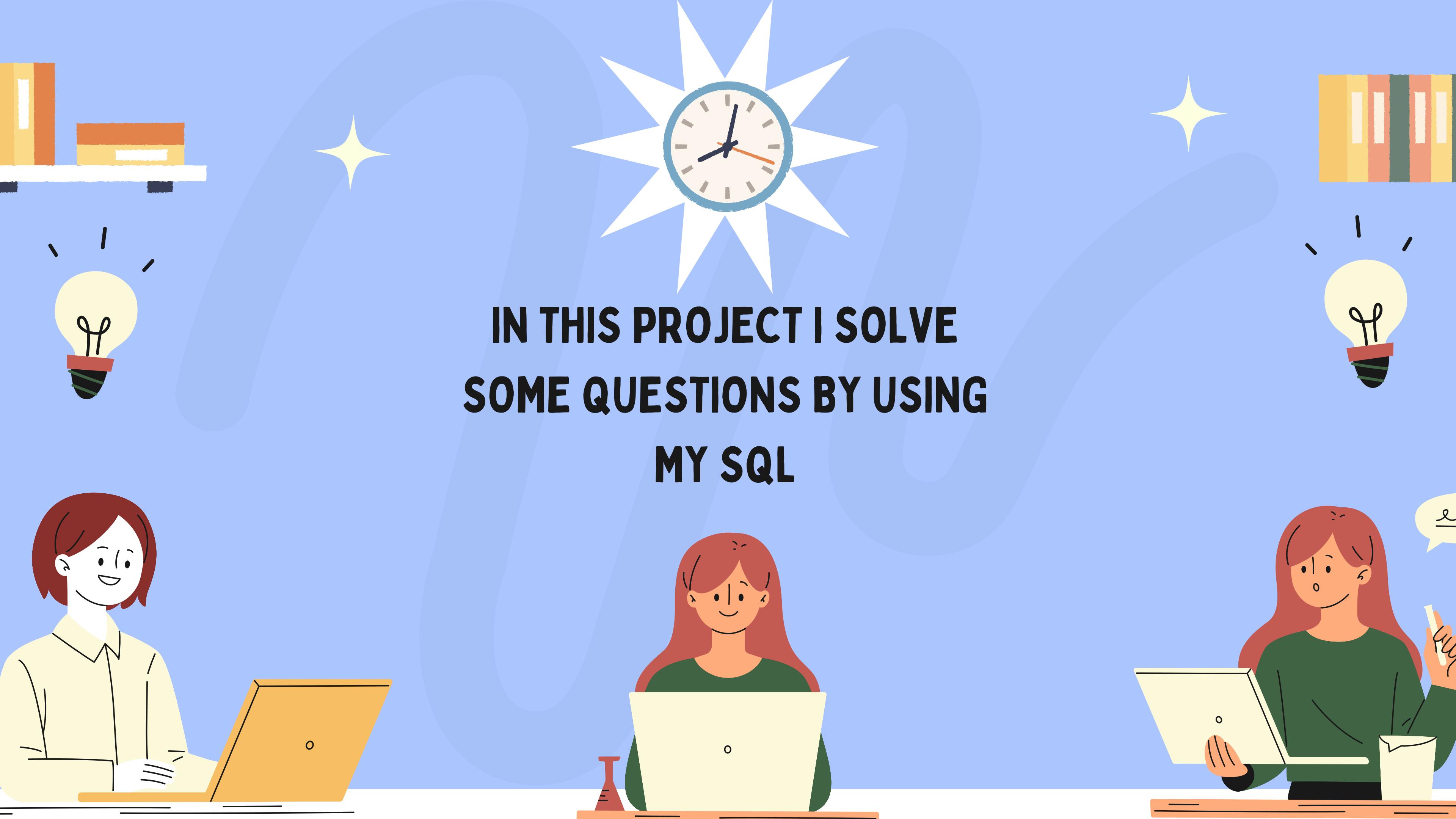
PRESENTED BY HARDIK JAIN

BACKGROUND

Michael Cachat

"These values that we hold and operate from I believe are a great reason for our success" say's founder and CEO, Michael Cachat. In 2010 Jenson USA expanded it's operations into a new 74,000 sq ft facility in Riverside California.





**IN THIS PROJECT I SOLVE
SOME QUESTIONS BY USING
MY SQL**

QUESTION 1



```
# Question 1  
#Find the total number of products sold  
#by each store along with the store name.
```

```
#Solution 1  
SELECT  
    stores.store_name,  
    SUM(order_items.quantity) AS total_quantity  
FROM  
    orders  
    JOIN  
    order_items ON order_items.order_id = orders.order_id  
    JOIN  
    stores ON stores.store_id = orders.store_id  
GROUP BY stores.store_name;
```

QUESTION 2

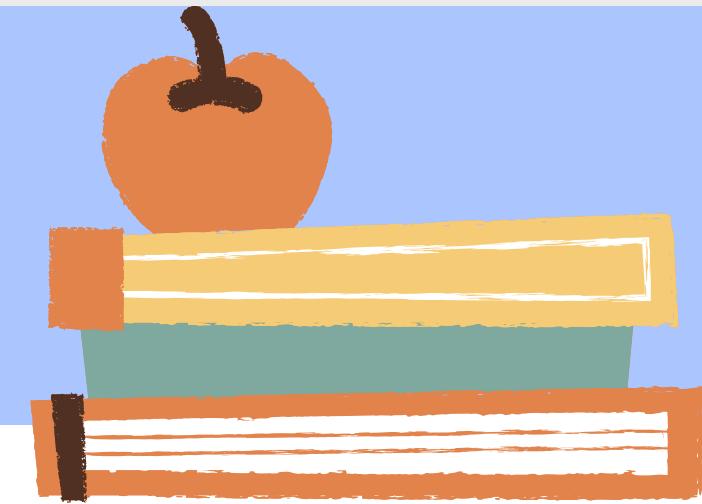


#Question 2

#Calculate the cumulative sum of quantities sold for each product over time.

#Solution 2

```
select product_id, order_date, quantity,  
sum(quantity) over (partition by product_id order by order_date)  
from  
(select order_items.product_id,  
orders.order_date, sum(order_items.quantity) quantity  
from orders join order_items  
on orders.order_id = order_items.order_id  
group by order_items.product_id, orders.order_date) a;
```



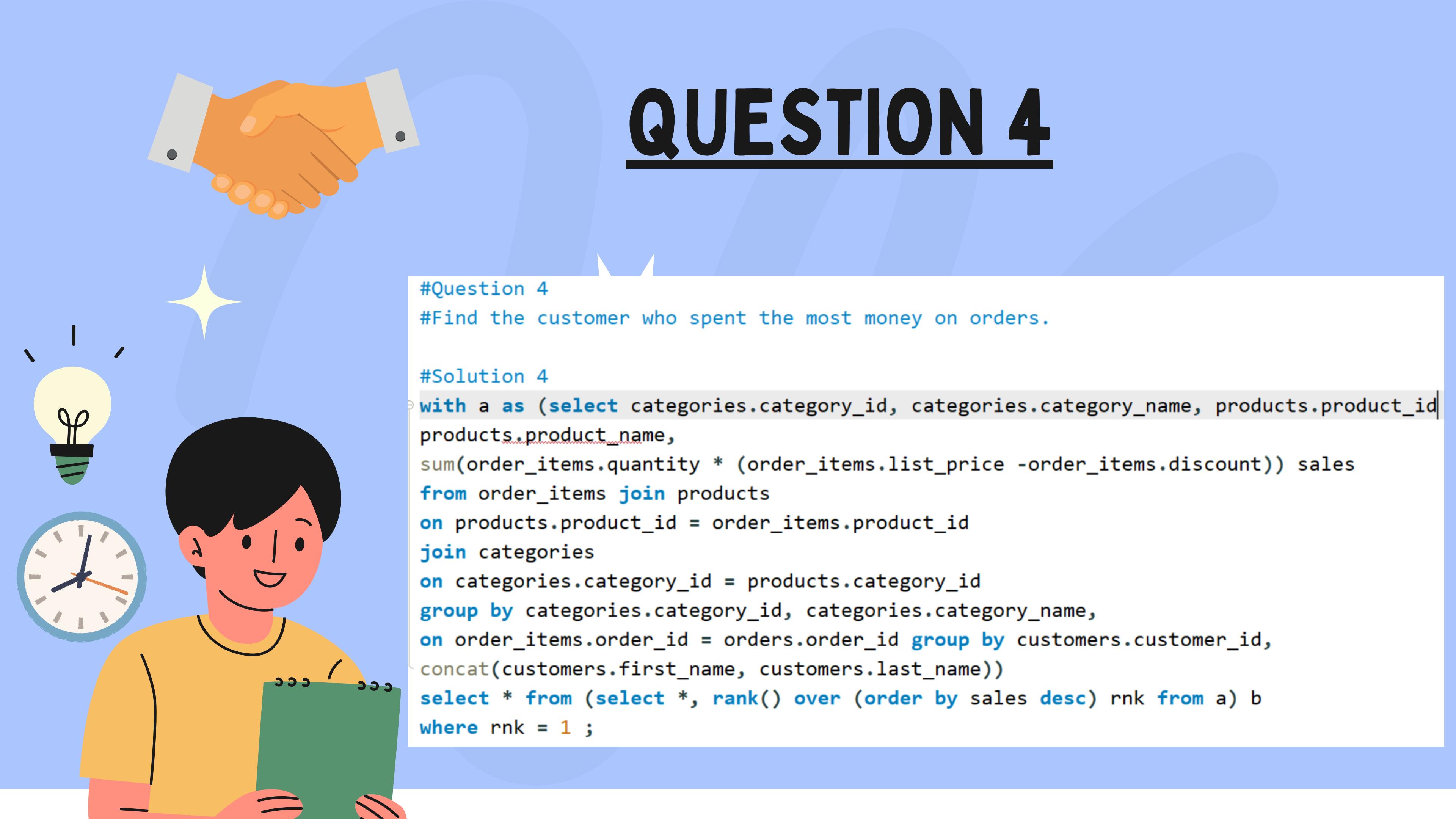
QUESTION 3

```
#Question 3  
#Find the product with the highest total sales  
#(quantity * price) for each category.
```

```
#Solution 3  
with a as (select categories.category_id, categories.category_name, products.product_id,  
products.product_name,  
sum(order_items.quantity * (order_items.list_price -order_items.discount)) sales  
from order_items join products  
on products.product_id = order_items.product_id  
join categories  
on categories.category_id = products.category_id  
group by categories.category_id, categories.category_name,  
products.product_id, products.product_name)  
select * from  
(select *, rank() over (partition by category_id order by sales desc) as rnk  
from a) b  
where rnk = 1;
```



QUESTION 4



```
#Question 4  
#Find the customer who spent the most money on orders.
```

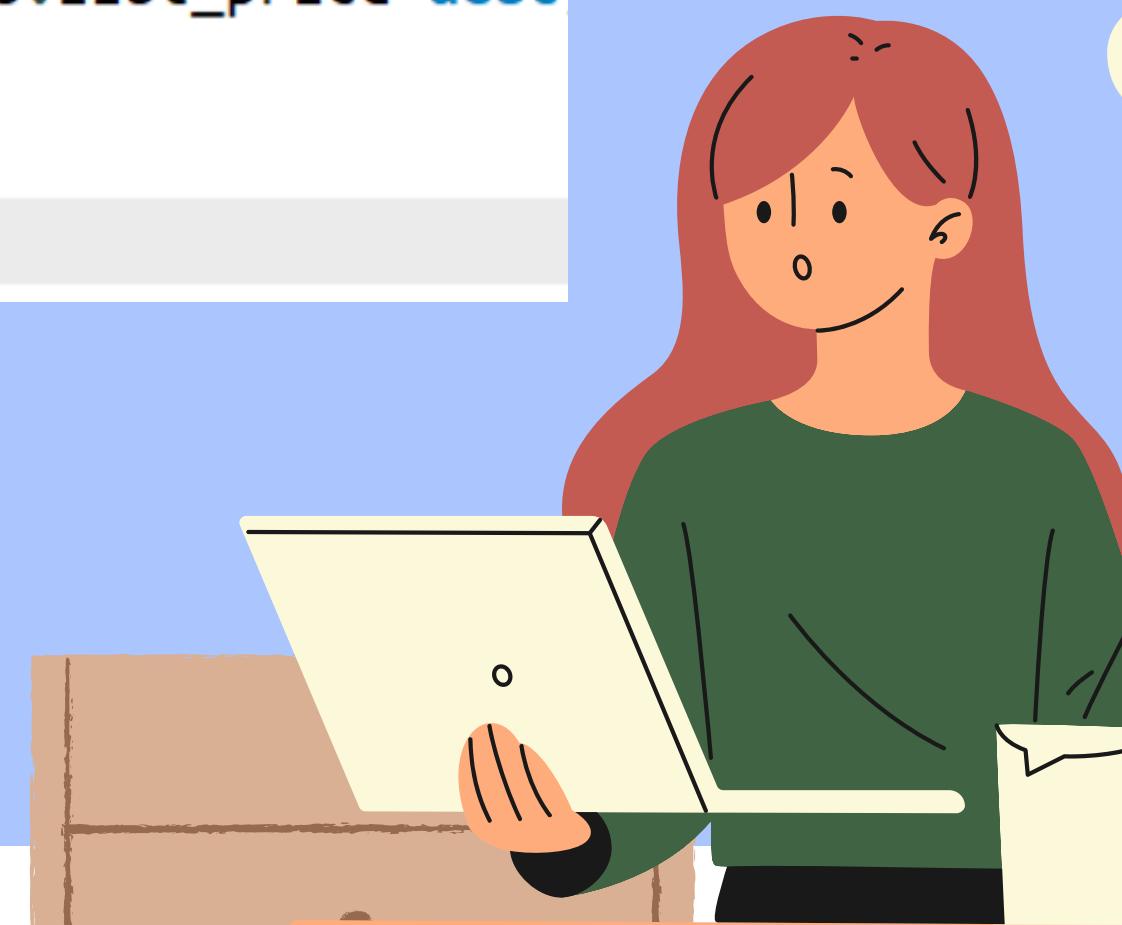
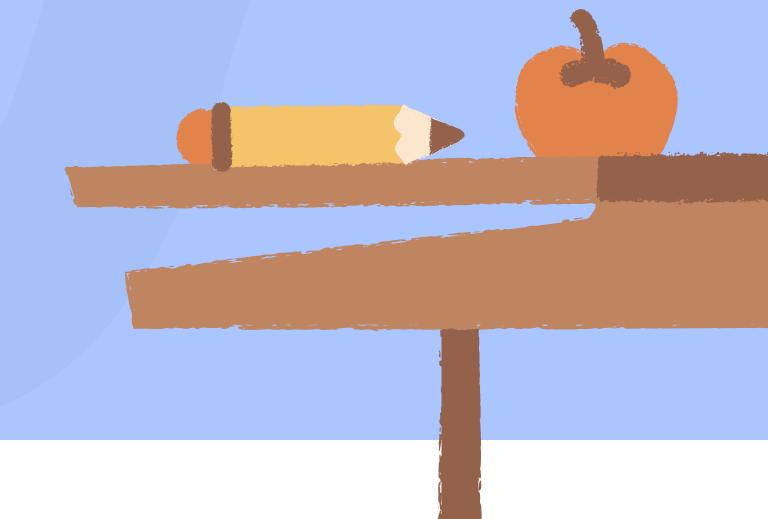
```
#Solution 4  
with a as (select categories.category_id, categories.category_name, products.product_id,  
products.product_name,  
sum(order_items.quantity * (order_items.list_price -order_items.discount)) sales  
from order_items join products  
on products.product_id = order_items.product_id  
join categories  
on categories.category_id = products.category_id  
group by categories.category_id, categories.category_name,  
on order_items.order_id = orders.order_id group by customers.customer_id,  
concat(customers.first_name, customers.last_name))  
select * from (select *, rank() over (order by sales desc) rnk from a) b  
where rnk = 1 ;
```

QUESTION 5



```
#Question 5
#Find the highest-priced product for each category name.
```

```
#Solution 5
select * from
(select categories.category_id, categories.category_name,
products.product_name,
products.list_price,
rank() over(partition by categories.category_id order by products.list_price desc
from products join categories
on products.category_id = categories.category_id) a
where rnk = 1;
```





QUESTION 6

#Question 6

#Find the total number of orders placed by each customer per store.

#Solution 6

SELECT

 StoreID,

 CustomerID,

 COUNT(OrderID) **AS** TotalOrders

FROM

 Orders

GROUP BY

 StoreID,

 CustomerID

ORDER BY

 StoreID,

 CustomerID;





QUESTION 7

#Question 7

#Find the names of staff members who have not made any sales.



#Solution 7

```
select staffs.staff_id,  
concat(staffs.first_name, " ", staffs.last_name) full_name  
from staffs where not exists  
(select staff_id from orders  
where orders.staff_id = staffs.staff_id);
```

QUESTION 8

#Question 8

#Find the top 3 most sold products in terms of quantity.

#Solution 8

```
select product_name from
(select products.product_id,
products.product_name, sum(order_items.quantity) quantity,
rank() over(order by sum(order_items.quantity) desc)
rnk from products join order_items
on products.product_id = order_items.product_id
group by products.product_id,
products.product_name) a
where rnk <= 3;
```





QUESTION 9

#Question 9

#Find the median value of the price list.

#Solution 9

```
with m as
(select list_price, row_number() over (order by list_price) rn,
count(list_price) over() cn from order_items)
select case
when cn % 2 = 0 then (select ag(list_price)
from m where rn in (cn/2, (cn/2) + 1))
else (select list_price from m where rn = (cn+1)/2)
end as median from m limit 1;
```





QUESTION 10

#Question 10

#List all products that have never been ordered.(use Exists)

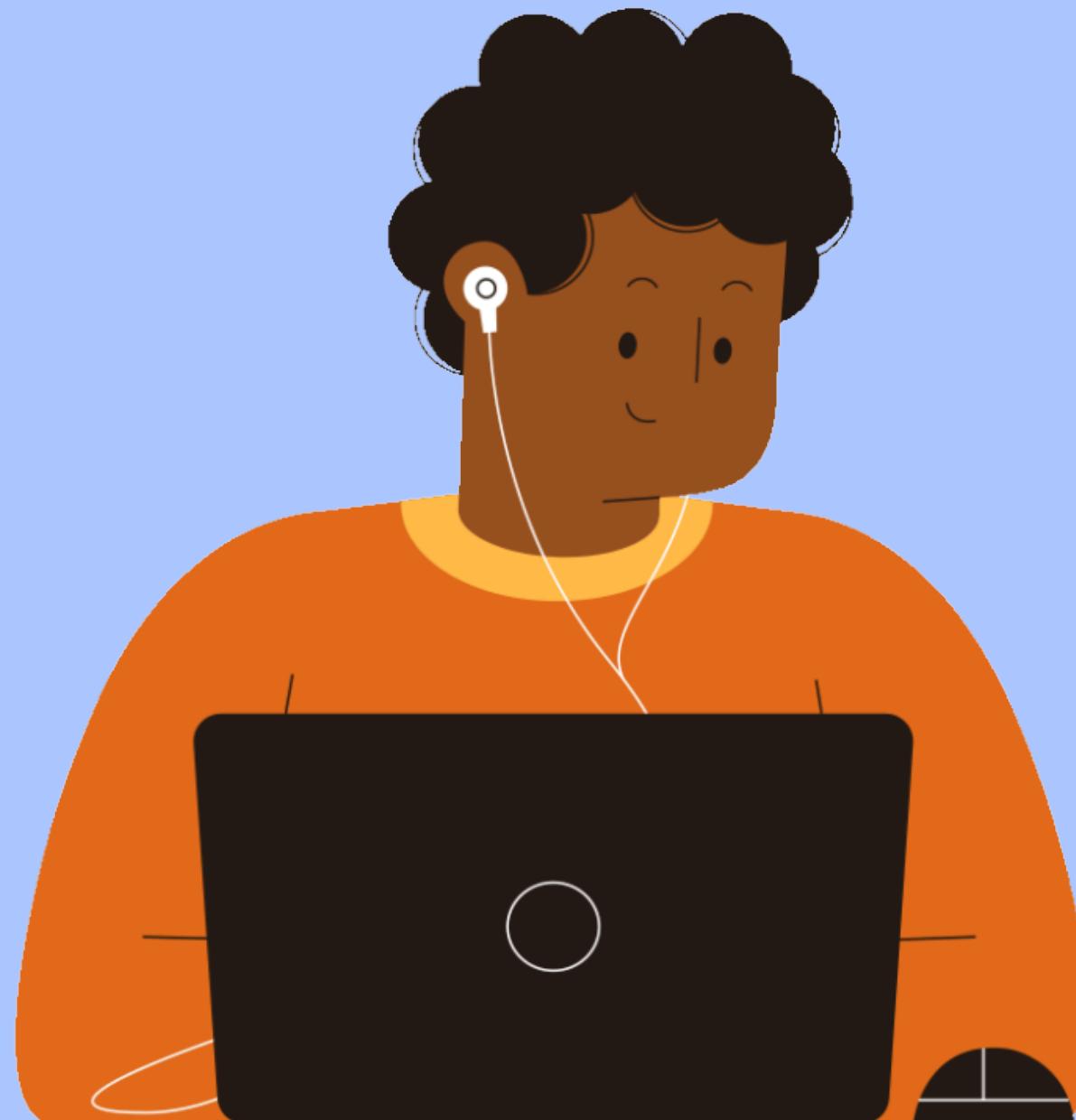


#Solution 10

```
select products.product_id,  
products.product_name from products  
where not exists (select product_id from order_items  
where order_items.product_id = products.product_id);
```

QUESTION 11

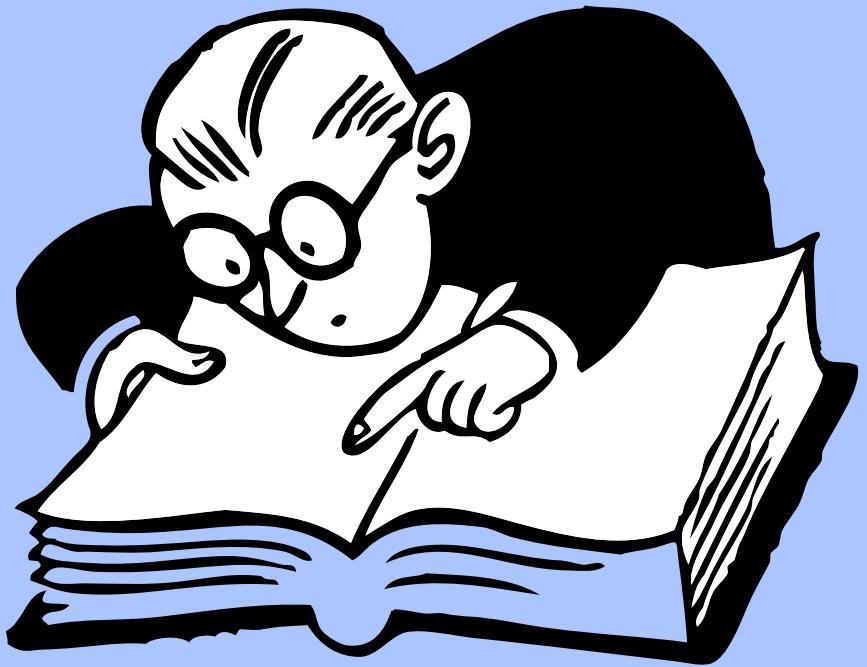
```
#Question 11
#List the names of staff members who have made more
#sales than the average number of sales by all staff members.
#Solution 11
select staffs.staff_id, coalesce(sum(order_items.quantity *
(order_items.list_price-order_items.discount)),0) sales
from orders right join staffs
on staffs.staff_id = orders.staff_id
left join order_items
on orders.order_id = order_items.order_id
group by staffs.staff_id
having sum(order_items.quantity *
(order_items.list_price-order_items.discount)) >
(select avg(sales) from (select staffs.staff_id,
| coalesce(sum(order_items.quantity * (order_items.list_price-order_items.discount)),0) sales
from orders right join staffs
on staffs.staff_id = orders.staff_id
left join order_items
I on orders.order_id = order_items.order_id
| ) )
```



QUESTION 12

```
#Question 12  
#Identify the customers who have ordered all  
#types of products (i.e., from every category).
```

```
#Solution 12  
select customers.customer_id  
from customers join orders  
on customers.customer_id = orders.customer_id  
join order_items  
on order_items.order_id = orders.order_id  
join products p  
on p.product_id = order_items.product_id  
group by customers.customer_id  
having count(distinct p.category_id) = (select count(category_id) from categories)  
select count(category_id) from categories;
```





HOPE YOU GUYS LIKED THE PROJECT

DO FOLLOW ME ON IINKED

IN

CLICK HERE

