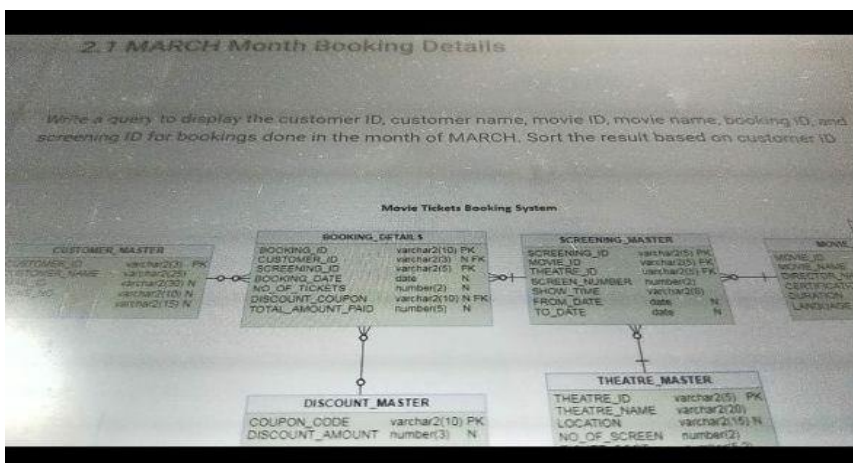


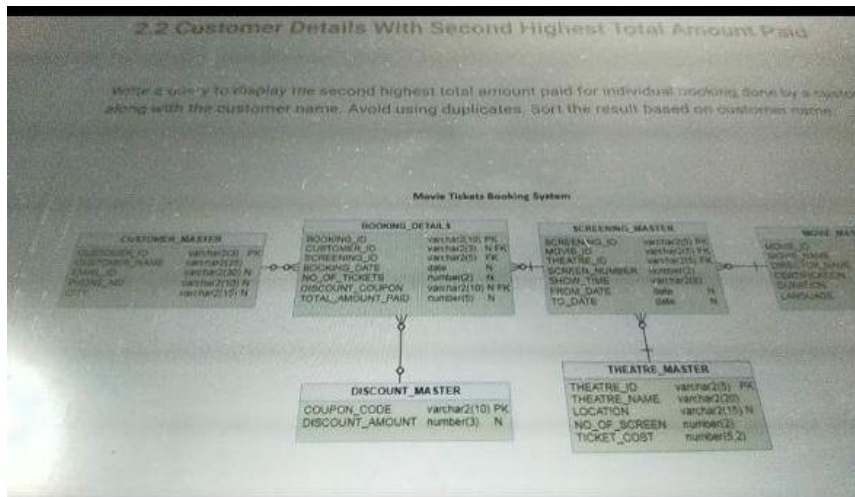
Select AccountType, count(AccountNo) AS TOTAL ACCOUNTS from Account\_Info where ifsc code="HDVL002" Group by AccountType order by AccountType;



Select c.customer\_id,c.customer\_name,m.movie\_id,m.movie\_name,b.booking\_id,b.screening\_id from CUSTOMER\_MASTER c inner join booking\_detail b using(customer\_id) Inner join screening\_master s using(screening\_id) Inner join Movie m using(movie\_id) where BOOKING\_DATE LIKE '%-MAR-%' order by customer\_id;

OR

To\_char(b.booking\_date,'Mon')='MAR' order by c.customer\_id;



Select distinct C.customer\_Name , max(B.Total\_Amount\_paid from Customer\_master C

inner join Booking\_details B on

C.cutomer\_id=B.customer\_id

Group by

C.customer\_Name

Having max( B.total\_amount\_paid) Not in (select max(total\_amount\_paid) from booking details  
Group by total\_amount\_paid)

Order by customer\_name;

OR

Select distinct C.customer\_Name , B.Total\_Amount\_paid from Customer\_master C

inner join Booking\_details B on

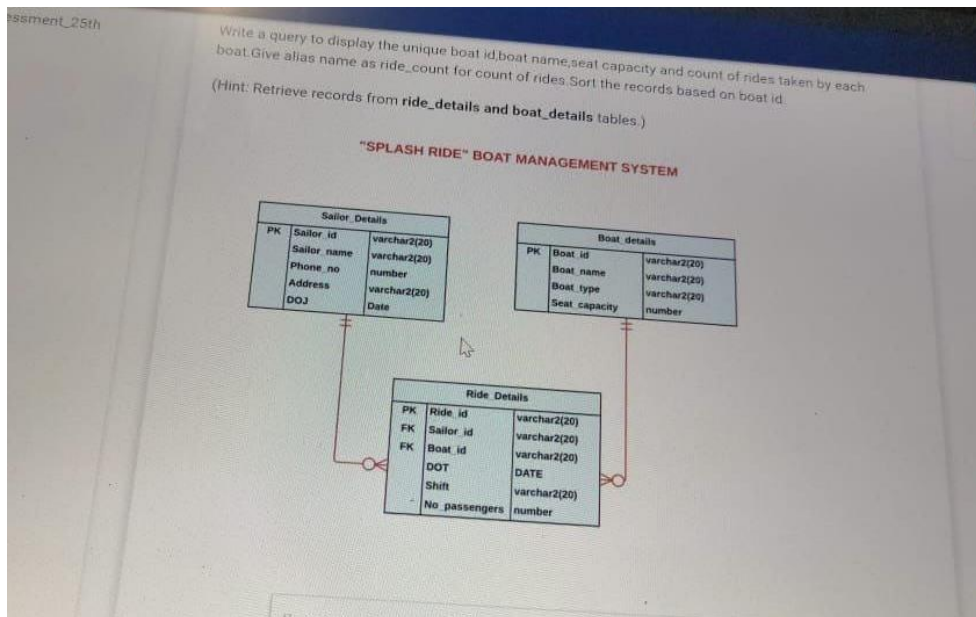
C.cutomer\_id=B.customer\_id

Group by

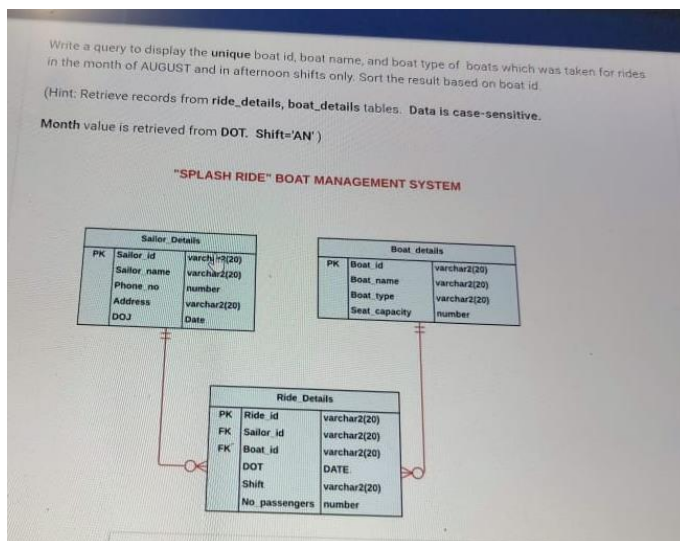
C.customer\_Name

Having max( B.total\_amount\_paid) =(select max(total\_amount\_paid) from booking details group by  
customer\_name having max(total\_amount\_paid) < (not in) (select max(total\_amount\_paid) from  
booking details )

Order by customer\_name;



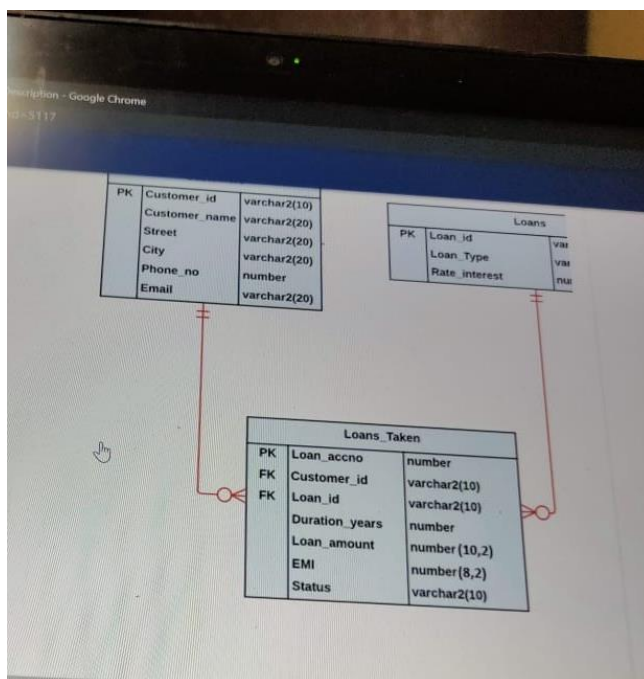
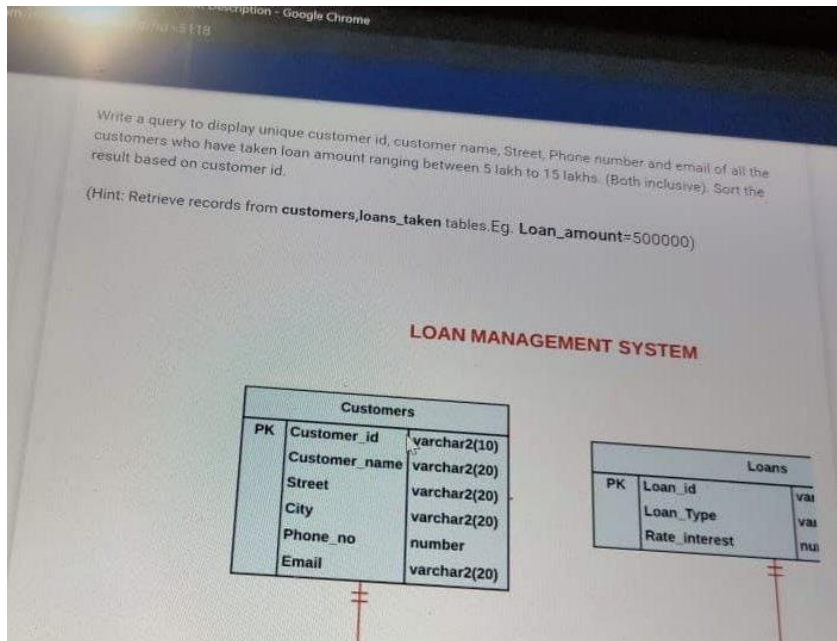
Select distinct b.boat\_id,b.boat\_name,b.seat\_capacity,count(r.ride\_id) as ride\_count from  
From boat\_details b inner join ride\_details r on b.boat\_id=r.boat\_id  
Group by boat\_id  
order by boat\_id;



Select distinct b.boat\_id,b.boat\_name,b.boat\_type  
from boat\_details b inner join Ride\_details r on  
b.boat\_id=r.boat\_id where to\_char(DOT,'Mon')='AUG' and r.Shift='AN'  
order by boat\_id;

OR

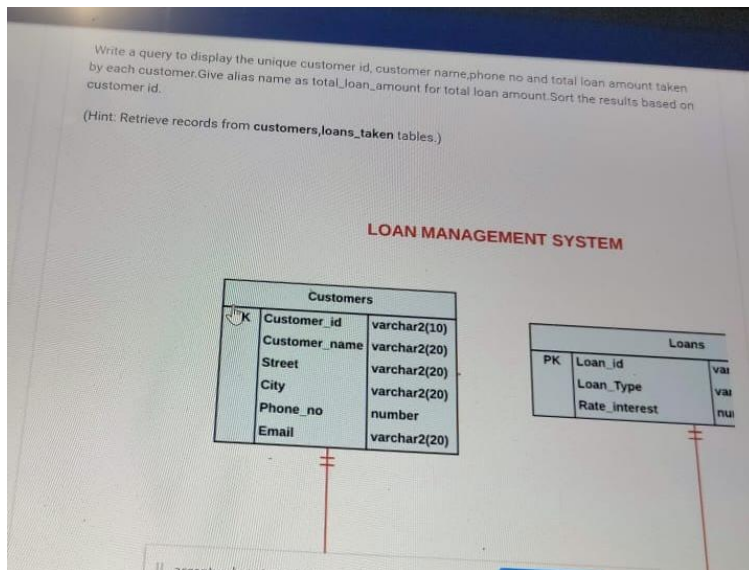
b.boat\_id=r.boat\_id where DOT LIKE '%-AUG-%' and r.Shift='AN'



```

Select c.customer_id,c.customer_name,c.street,c.phone,c.email from
Customers c inner join Loan_taken l on c.customer_id=l.customer_id
Where l.Loan_amount between 500000 and 1500000
Order by customer_id;

```



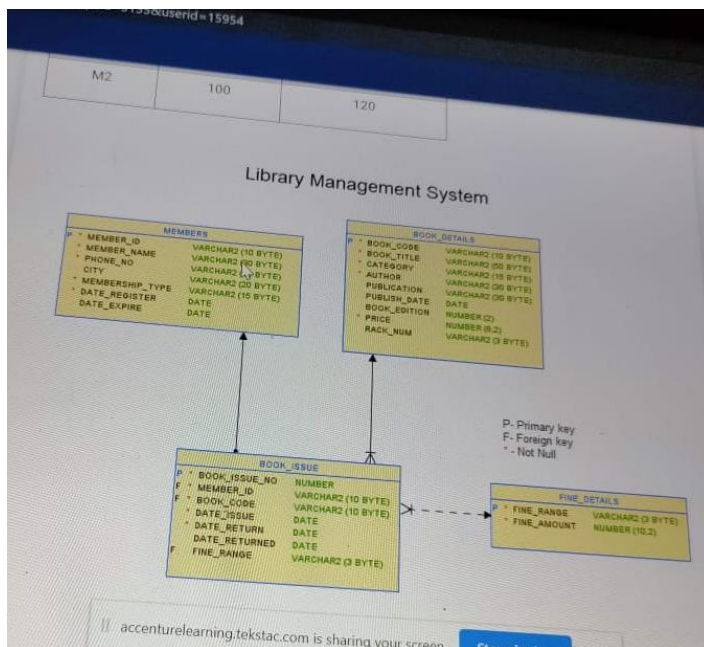
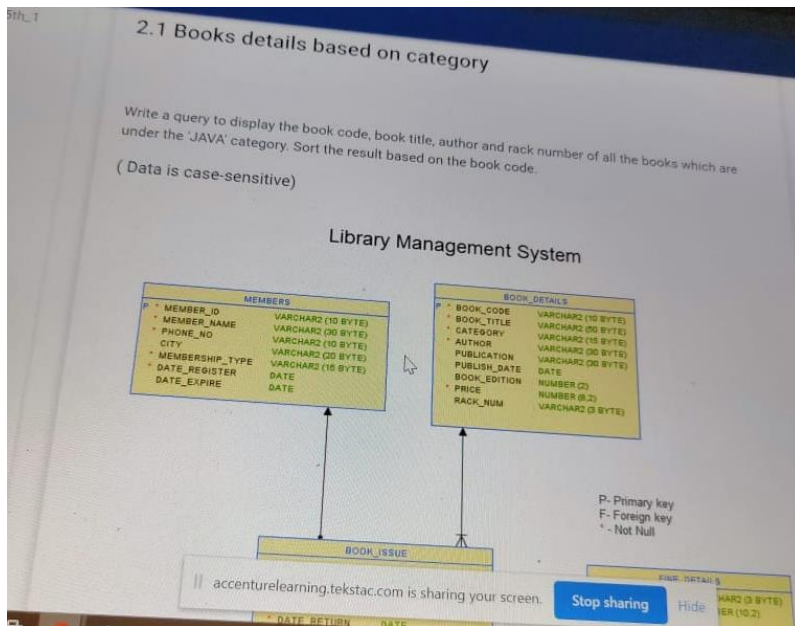
```
Select distinct c.customer_id,c.customer_name,c.phone_no,sum(l.loan_amount) as  
total_loan_amount
```

```
from customers c inner join loan_taken l
```

```
On c.customer_id=l.customer_id
```

```
Group by customer_id
```

```
Order by customer_id;
```



Select book\_code,book\_title,author,rack\_num from book\_details

Where category='JAVA'

Order by book\_code;

2.2 Fine details

The ABC college management decided to increase the fine amount for the books not returned, based on the fine range.

Write a query to display details like fine range, fine amount and new fine amount. Give an alias name as NEW\_FINE\_AMOUNT.

Refer to the given table for the new fine amount calculation. (Hint: Retrieve records from **Fine\_details** table. Use CASE or DECODE)

Fine_Range	Increased_Amount(in rs)
L1	5
L2	10
M1	15
M2	20
H1	

accenturelearning.tekstac.com is sharing your screen. [Stop sharing](#) [Hide](#)

L1	5
L2	10
M1	15
M2	20
H1	25
H2	30

Sample Output:

FINE_RANGE	FINE_AMOUNT	NEW_FINE_AMOUNT
L1	50	55
M2	100	120

Select fine\_range, fine\_amount

Case

When fine\_range='L1' then 5+fine\_amount

When fine\_range='L2' then 10+fine\_amount

When fine\_range='M1' then 15+fine\_amount

When fine\_range='M2' then 20+fine\_amount

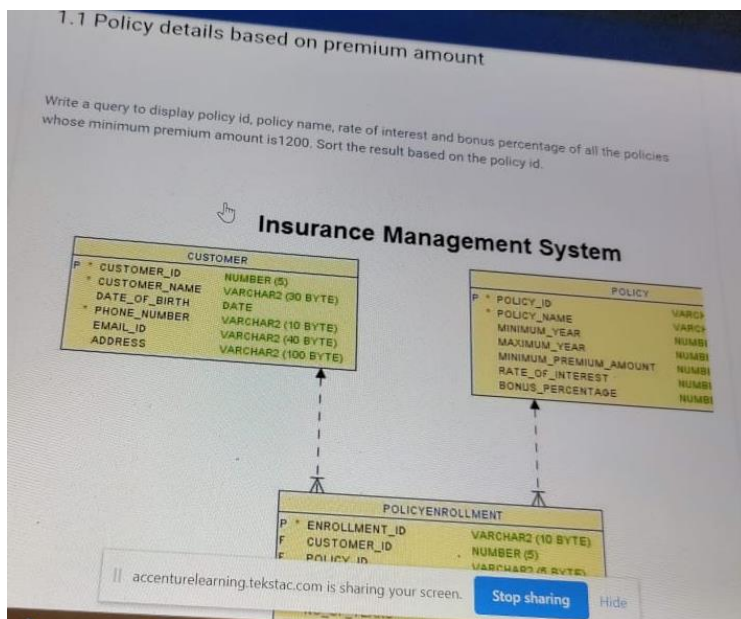
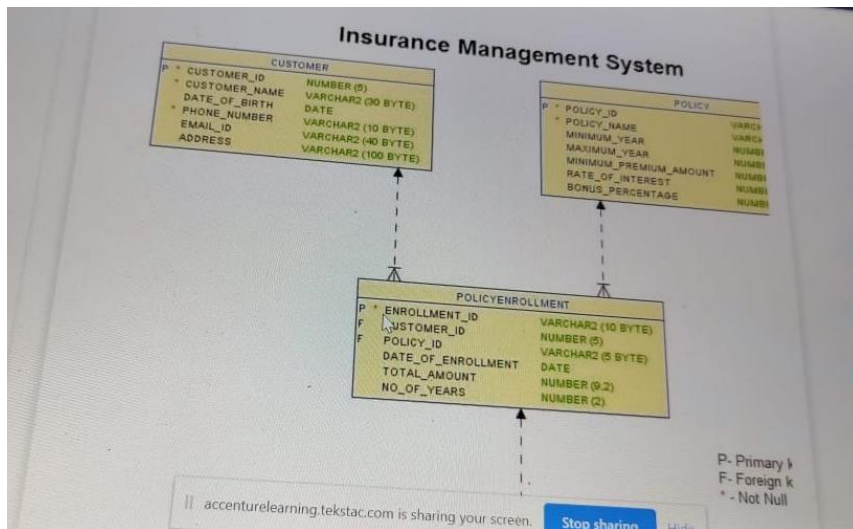
When fine\_range='H1' then 25+fine\_amount

When fine\_range='H2' then 30+fine\_amount

End as NEW\_FINE\_AMOUNT

From fine\_details f join book\_issue b

On f.fine\_range=b.fine\_range where b.date\_returned is null ;



Select policy\_id,policy\_name,rate\_of\_interest,bonus\_percentage from policy

Where minimum\_premium\_amount=1200

Order by policy\_id;



1.2 Enrollment Details

Write a query to display the customer name along with the policy enrollment date. Give an alias name as 'ENROLLMENT\_DETAILS'. Sort the result by customer\_name,date\_of\_enrollment in ascending order.

(HINT: Use **customer** and **policy enrollment** table. Use `concat()`)

Refer to the sample output.

ENROLLMENT_DETAILS
Bob has taken policy on 18-Nov-17
Bella has taken policy on 01-Dec-17

**Insurance Management System**

CUSTOMER	
P * CUSTOMER_ID	NUMBER (5)
* CUSTOMER_NAME	VARCHAR2 (30 BYTE)
DATE	
* PHONE	
EMAIL	
ADDRESS	VARCHAR2 (100 BYTE)

POLICY	
P * POLICY_ID	NUMBER (5)
* POLICY_NAME	VARCHAR2 (30 BYTE)
DATE	
AMOUNT	NUMBER (10)
RATE OF INTEREST	NUMBER (5)

accenturelearning.tekstac.com is sharing your screen. Stop sharing Hide

```
Select concat(concat(c.customer_name,' has taken policy on '),p.date_of_enrollment)
as Enrollment_details from customer c inner join policy_enrollment p
on c.customer_id=p.customer_id
order by c.customer_name,p.date_of_enrollment;
```