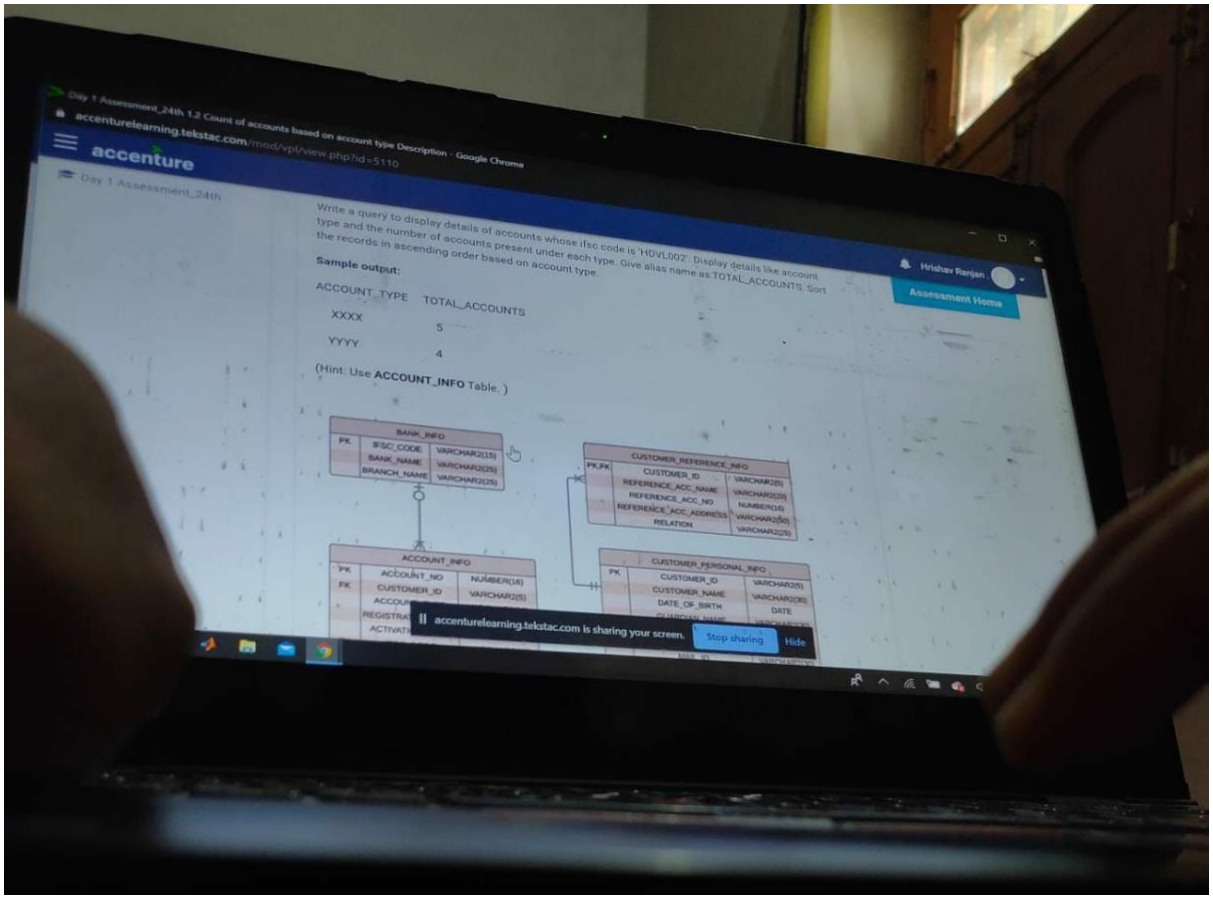
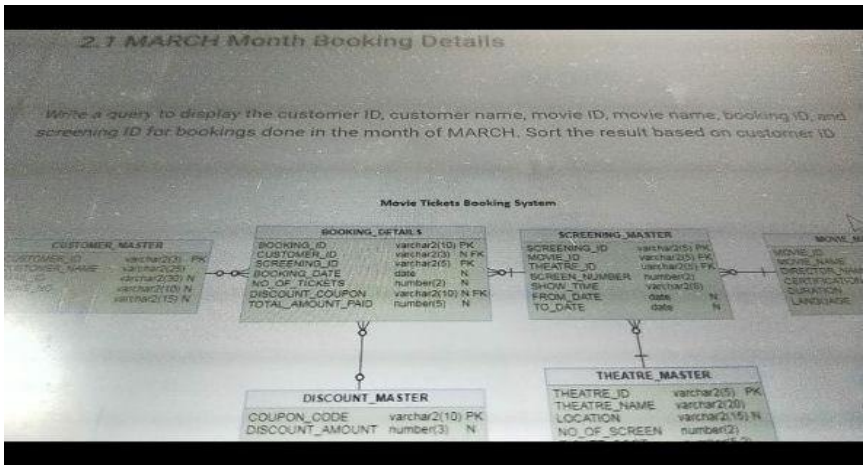
**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**




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

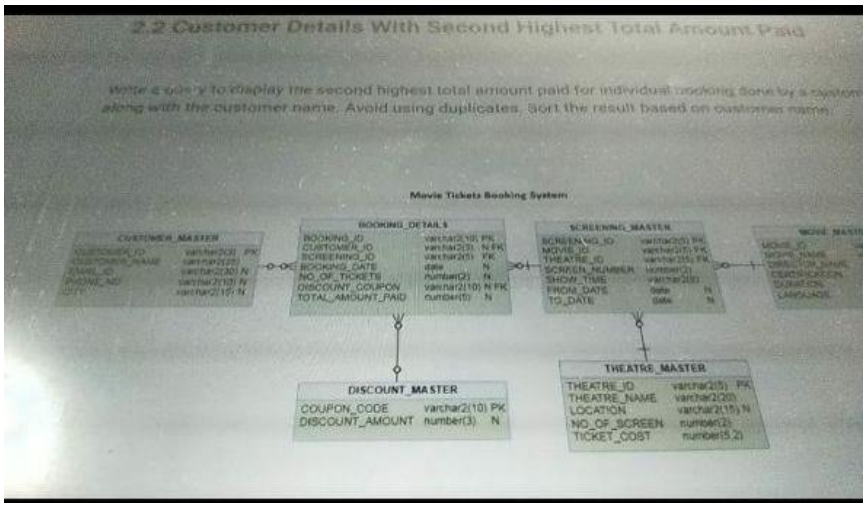



Select c.customer_id,c.customer_name,m.movie_id,m.movie_name,b.booking_id,b.screening_id from CUSTOMER_MASTER c inner join booking_detail b using(customer_id) Inner join screening_master s using(screening_id) Inner join Movie m using(movie_id) where BOOKING_DATE LIKE '%%-MAR-%%' order by customer_id;

OR

To_char(b.booking_date,'Mon')='MAR' order by c.customer_id;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**




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

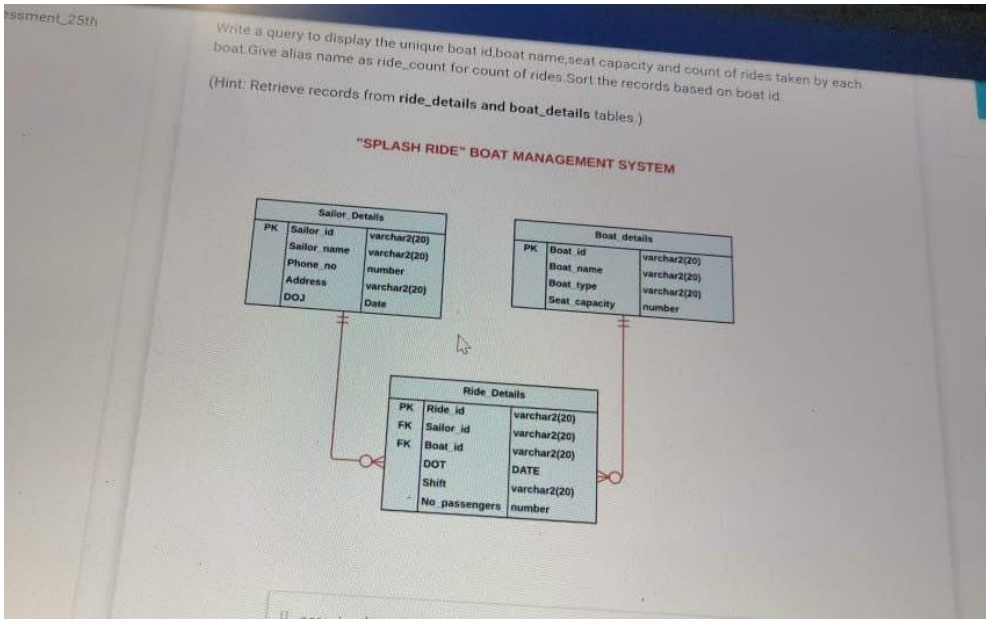
Group by

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;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**

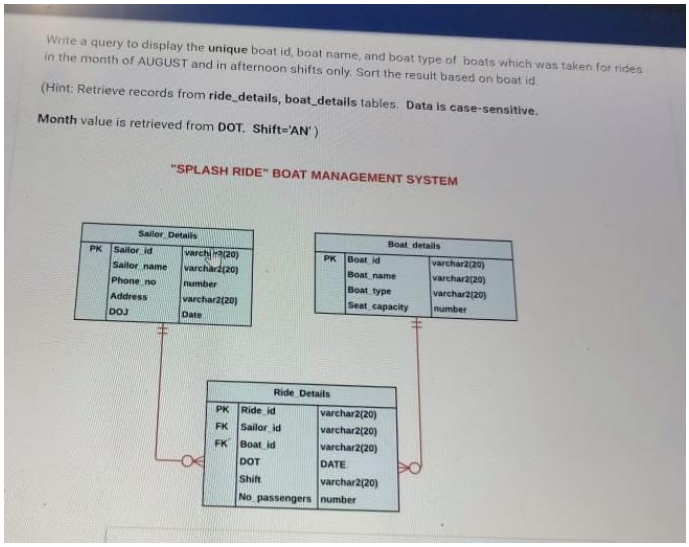



Select distinct b.boat_id,b.boat_name,b.seat_capacity,count(r.ride_id) as ride_count from

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

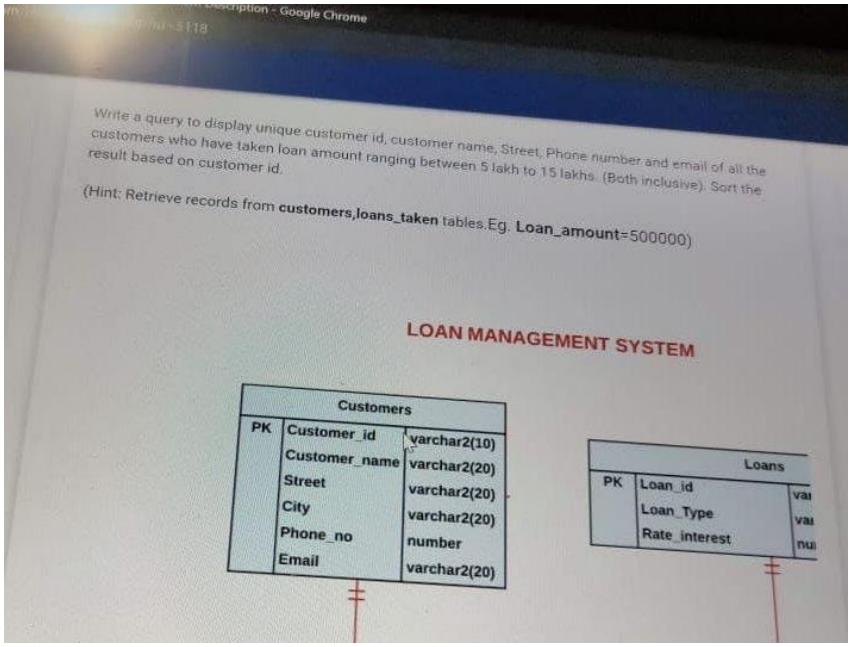
b.boat_id=r.boat_id where to_char(DOT,'Mon')='AUG' and r.Shift='AN'

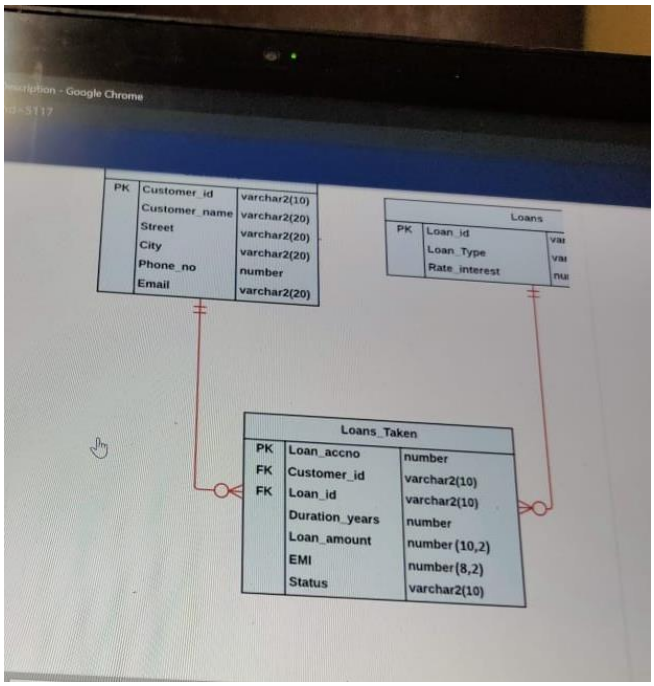
order by boat_id;

OR

b.boat_id=r.boat_id where DOT LIKE '%%-AUG-%%' and r.Shift='AN'

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**



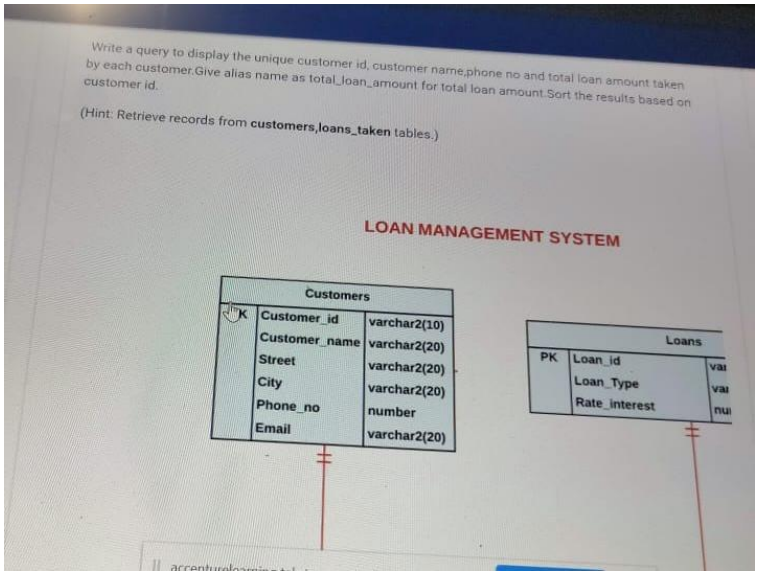

Select c.customer_id,c.customer_name,c.street,c.phone,c.email from

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;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**




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

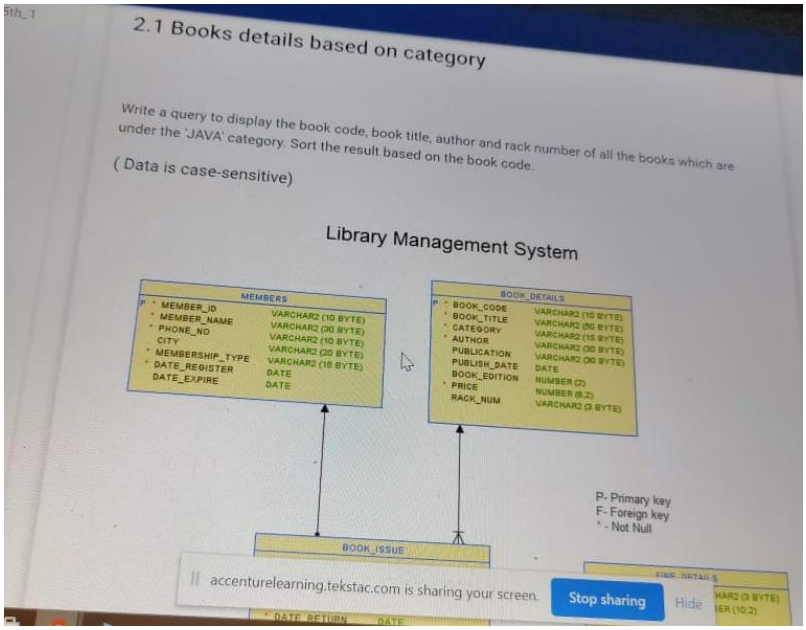
from customers c inner join loan_taken l

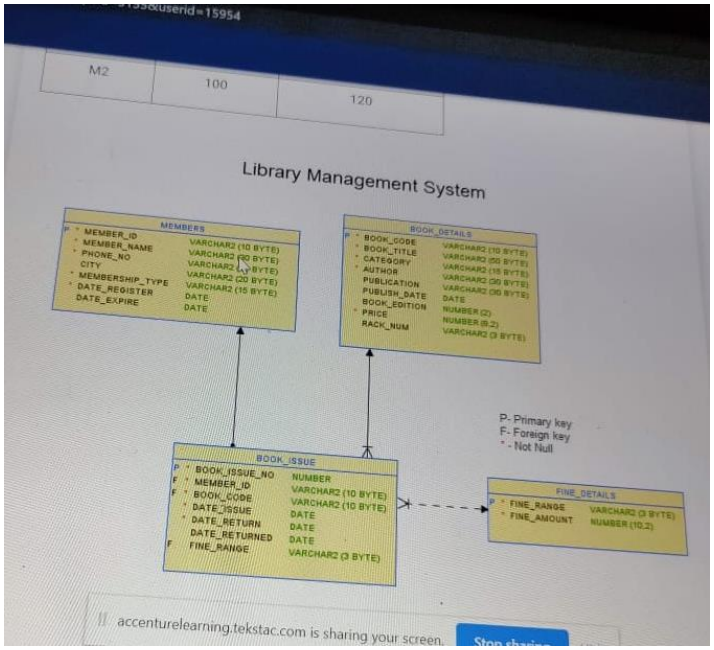
On c.customer_id=l.customer_id

Group by customer_id

Order by customer_id;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**



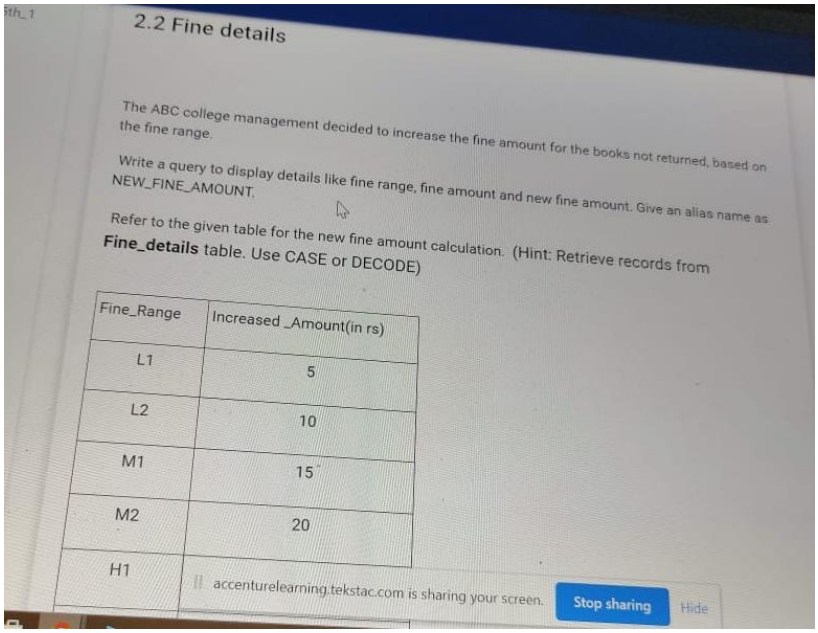




Select book_code,book_title,author,rack_num from book_details

Where category='JAVA'

Order by book_code;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**



## 2.2 Fine details

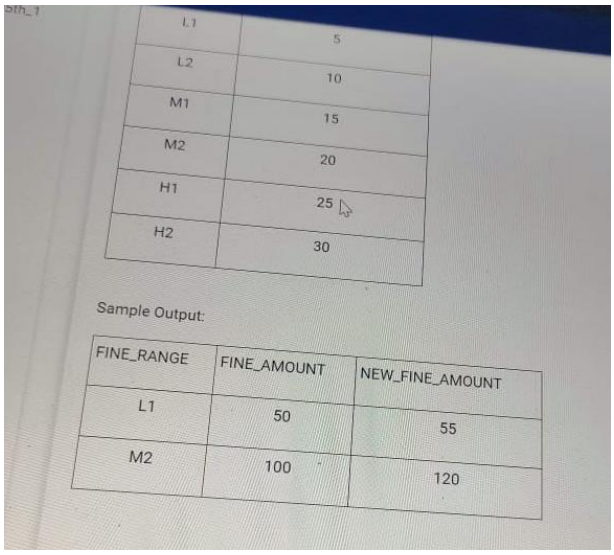
The ABC college management decided to increase the fine amount for the books not returned, based on the fine range.

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



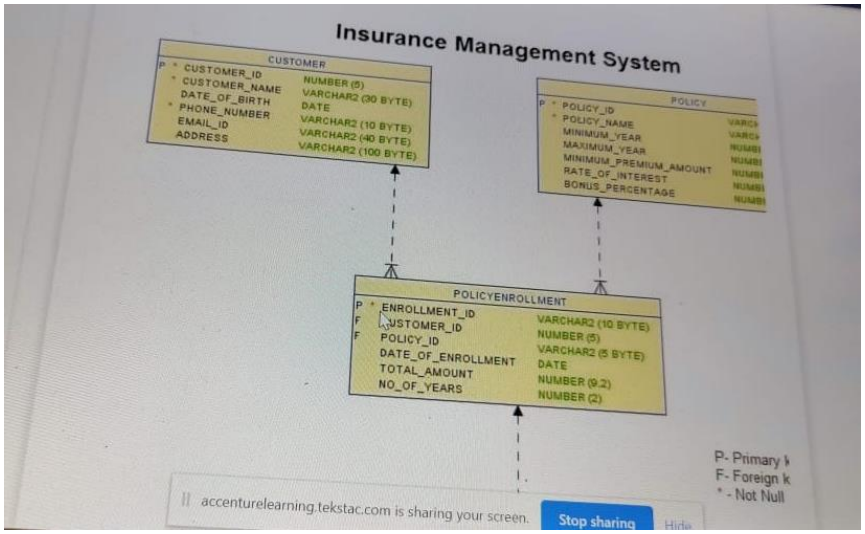
| | |
|---|---|
| L1 | 5 |
| L2 | 10 |
| M1 | 15 |
| M2 | 20 |
| H1 | 25 |
| H2 | 30 |

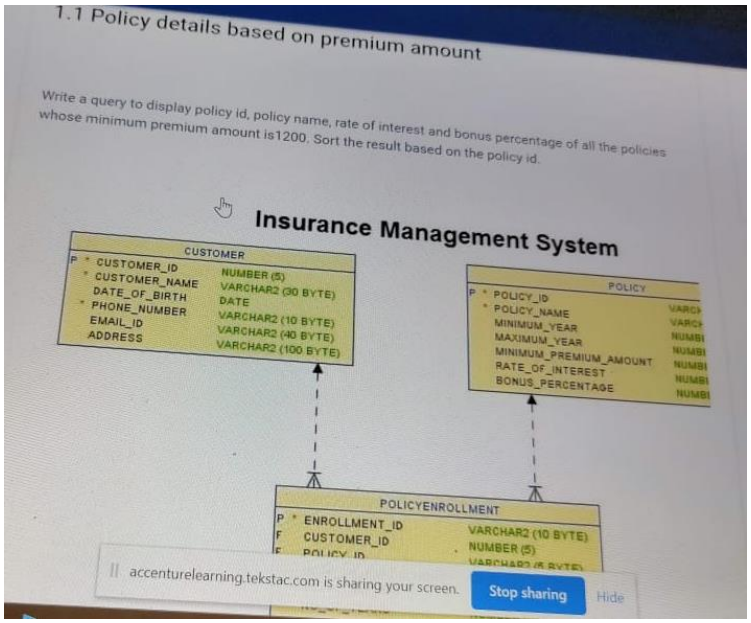
Sample Output:

| 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 ;
```

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**



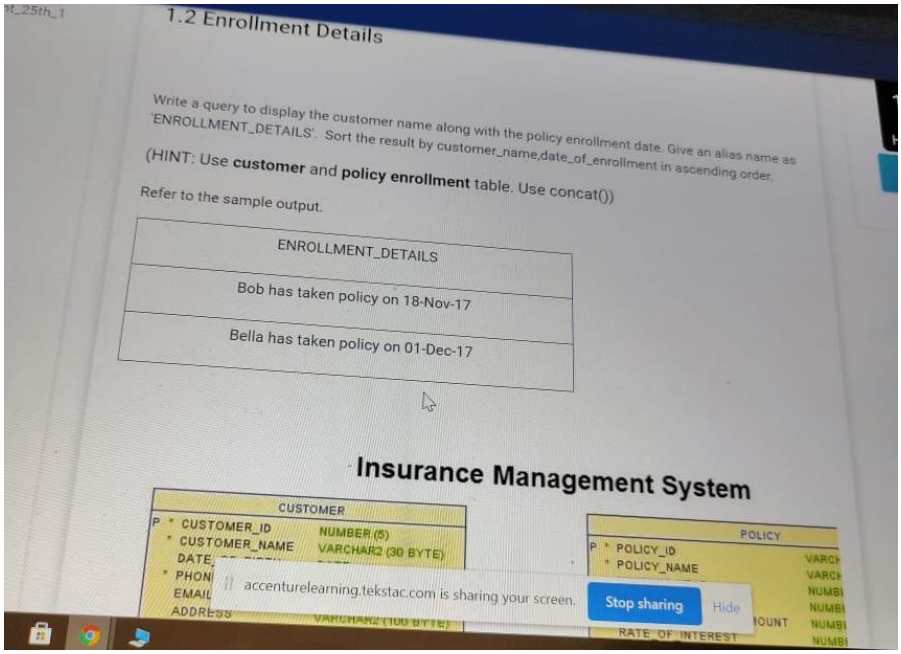




Select policy_id,policy_name,rate_of_interest,bonus_percentage from policy

Where minimum_premium_amount=1200

Order by policy_id;

**Date:** 24th and 25th August **Subject:** RDBMS **Final Assessment**




```
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;
```