**//Basic Queries//**

1. Find Total number of customers in the database using the Function **totalCustomer** to make easier statistical view.

select distinct totalCustomer from customer;

----------------------------------------------------------------------

2. Find average rating of a specific book where book id is given. Use **avgRating(b\_id)** function to make smaller query. Take Id from user input

select DISTINCT b\_id,book\_title,avgRating(b\_id)

from reviews join books on(b\_id=book\_id)

where b\_id = '&b\_id';

----------------------------------------------------------------------

3. Show All purchase list/bills of ALL customers with purchase date, who have bought something.

select distinct purchase.c\_id, order\_count\*unit\_price, purchase.PURCHASE\_DATE

from purchase join books on(b\_id= book\_id)

ORDER BY c\_id;

----------------------------------------------------------------------

4. Show Total revenue of any specific customer as it is needed to update their discount level. Take user input for that.

select distinct customer.customer\_name, purchase.c\_id, sum(order\_count\*unit\_price)

from purchase join books on(b\_id= book\_id) join customer on(c\_id = customer\_id)

where customer\_name = '&customer\_name'

group by customer.customer\_name,purchase.c\_id;

----------------------------------------------------------------------

5. Show Total revenue of every customer as it is needed to update their discount level or have an overview on regular/most important customer. Sort them according to their revenue- high to low.

With temp as (select c\_id,sum(order\_count\*unit\_price) as total

from purchase join books on(b\_id= book\_id)

group by c\_id)

select distinct customer.customer\_id, customer.CUSTOMER\_NAME, total

from temp join customer on(c\_id = customer\_id)

order by total desc;

----------------------------------------------------------------------

6. Find the customer who has the most revenue , who has spent most to purchase books.

With temp as (select c\_id,sum(order\_count\*unit\_price) as total

from purchase join books on(b\_id= book\_id)

group by c\_id)

select distinct customer.customer\_id, customer.CUSTOMER\_NAME, total

from temp join customer on(c\_id = customer\_id)

where total = (select max(total) from temp);

----------------------------------------------------------------------

7. Show all the reviews along with author,book title and review date of a specific book by all customers who have given a feedback. take user input.

select c\_id,book\_title,summery,review\_date

from reviews join books on(b\_id = book\_id)

where b\_id = '&b\_id';

----------------------------------------------------------------------

8. Show all the Rent orders from all customers along with customer name and sort them by date of apply.

select customer\_name,c\_id, days, rent\_date

from rent join customer on(c\_id = customer\_id)

order by rent\_date;

----------------------------------------------------------------------

9. Show the wishlist for specific customer alongwith the info of the book.And customer should be able to put his/her customer id via hand input.

select book\_title, b\_id,c\_id

from wishlist join books on(b\_id=book\_id)

where c\_id = '&c\_id';

----------------------------------------------------------------------

10. Show start and end date of rental issue for every customer who have issued a book according to the returning date of the book.

select c\_id,b\_id, rent\_date, (rent\_date + days ) as returning\_date

from rent

order by returning\_date;

----------------------------------------------------------------------

11. Show the customers who has due date to return the book (those who have to return to the day instant). This query will find out which customers have to or haven’t return their books in the designated date.

select c\_id,b\_id,(rent\_date + days ) as returning\_date, sysdate as current\_date

from rent

where rent\_date >= sysdate

order by returning\_date;

----------------------------------------------------------------------