/\*Begum Zubeda A\*/

Create database ExamPGA02;

use ExamPGA02;

Select \* from bank\_account\_details;

Select \* from bank\_account\_transaction;

Select \* from bank\_customer;

Select \* from customer;

Select \* from salesman;

Select \* from orders;

/\*1. Write a SQL query which will sort out the customer and their grade who made an order. Every customer must have a grade and be served by at least one seller, who belongs to a region.\*/

Select c.\*, o.ord\_no, s.city as Sales\_City from customer c

Inner Join orders o On c.custemor\_id = o.customer\_id

Inner Join salesman s On c.salesman = s.salesman\_id

where c.grade IS NOT NULL and c.city = s.city

order by c.cust\_name, c.grade;

/\*2. Write a query for extracting the data from the order table for the salesman who earned the maximum commission.\*/

Select o.\*, s.name as salesman\_name, max(s.commision) as max\_commision from orders o

Inner Join salesman s ON o.salesman\_id = s.salesman\_id;

/\*3. From orders retrieve only ord\_no, purch\_amt, ord\_date, ord\_date, salesman\_id where salesman’s city is Nagpur(Note salesman\_id of orderstable must be other than the list within the IN operator.)\*/

Select o.ord\_no, o.purch\_amt, o.ord\_date, o.salesman\_id from orders o

Inner Join salesman s ON o.salesman\_id = s.salesman\_id

where s.city = 'Nagpur';

/\*4. Write a query to create a report with the order date in such a way that the latest order date will come last along with the total purchase amount and

the total commission for that date is (15 % for all sellers).\*/

Select o.ord\_no, str\_to\_date(o.ord\_date, '%d-%m-%Y') as order\_date, s.commision, sum(purch\_amt) as total\_purchase from orders o

Inner Join salesman s ON s.salesman\_id = o.salesman\_id

where s.commision = 0.15

group by order\_date

order by order\_date;

/\*5. Retrieve ord\_no, purch\_amt, ord\_date, ord\_date, salesman\_id from Orders table and display only those sellers whose purch\_amt is greater than average purch\_amt.\*/

Select ord\_no, purch\_amt, ord\_date, salesman\_id from orders

where purch\_amt > (

Select AVG(purch\_amt) from orders

);

/\*6. Write a query to determine the Nth (Say N=5) highest purch\_amt from Orders table. \*/

Select \* from(

Select \* from orders

order by purch\_amt desc limit 5

)as temp\_orders order by purch\_amt limit 1;

/\*7. What are Entities and Relationships? \*/

/\*Entities are objects that exist that can be anything a person, place and there can be a relationship between them like a person can be from a place say Mumbai\*/

/\*8. Print customer\_id, account\_number and balance\_amount, condition that if balance\_amount is nil then assign transaction\_amount for account\_type = "Credit Card" \*/

Select a.Customer\_id, a.Account\_Number, IF(a.Balance\_amount = 0 , t.Transaction\_amount, a.Balance\_amount) as BalanceAmount from bank\_account\_details a, bank\_account\_transaction t

where a.Account\_type = "Credit Card" and a.Account\_Number = t.Account\_Number;

/\*9. Print customer\_id, account\_number, balance\_amount, conPrint account\_number, balance\_amount, transaction\_amount from Bank\_Account\_Details and

bank\_account\_transaction for all the transactions occurred during march, 2020 and april, 2020. \*/

Alter table bank\_account\_transaction modify Transaction\_date date;

Select a.Customer\_id, a.Account\_Number, a.Balance\_amount, t.Transaction\_amount, t.Transaction\_Date from bank\_account\_details a

Inner Join bank\_account\_transaction t ON a.Account\_Number = t.Account\_Number

where t.Transaction\_Date between '2020-03-01' and '2020-04-01';

/\*10. Print all of the customer id, account number, balance\_amount, transaction\_amount from bank\_cutomer, bank\_account\_details and bank\_account\_transactions tables where excluding all of their transactions in march, 2020 month . \*/

Select a.Customer\_id, a.Account\_Number, a.Balance\_amount, t.Transaction\_amount, t.Transaction\_Date from bank\_account\_details a, bank\_account\_transaction t

where a.Account\_Number = t.Account\_Number and t.Transaction\_Date Between '2020-03-01' and '2020-03-31';