SQL EXAM.

QN.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.

QUERY→

```
SELECT customer.cust_name AS "Customer",
customer.grade AS "Grade",orders.ord_no AS "Order No."
FROM orders, salesman, customer
WHERE orders.customer_id = customer.customer_id
AND orders.salesman_id = salesman.salesman_id
AND salesman.city IS NOT NULL
AND customer.grade IS NOT NULL;
```

QN.2:

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

QUERY-->

```
SELECT ord_no, purch_amt, ord_date, salesman_id FROM orders
WHERE salesman_id IN ( SELECT salesman_id FROM salesman
WHERE commission = ( SELECT MAX(commission) FROM salesman) );
```

QN.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.)

QUERY->

```
SELECT *FROM orders
WHERE salesman_id =(SELECT salesman_id
   FROM salesman
   WHERE City='Nagpur');
```

QN.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).

QUERY→
```
SELECT ord_date, SUM(purch_amt),
SUM(purch_amt)*.15 FROM orders

GROUP BY ord_date

ORDER BY ord_date;
```

QN.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.

QUERY→
```
SELECT *FROM orders

WHERE salesman_id =(SELECT DISTINCT salesman_id FROM orders

   WHERE  purch_amt > avg(purch_amt);
```

QN.6:

Write a query to determine the Nth (Say N=5) highest `purch_amt` from Orders table.

QUERY→
```
SELECT purch_amt  FROM  ORDER BY order DESC LIMIT N-1, 1;
```

QN.7:

What are Entities and Relationships?

ANS→

- **Entity-Relationship model or E- R model is used to create a relationship between different attributes or entities.**
- **It describes the structure of the database with the help of the ER Diagram or Entity Relationship Diagram.**
- **ER model creates a simple design view of the data.**

QN.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"

Query→Select customer_id , ba.account_number, Case when

ifnull(balance_amount,0) = 0 then Transaction_amount

else balance_amount end as balance_amount

from Bank_Account_Details ba inner join bank_account_transaction bat on ba.account_number = bat.account_number and account_type = "Credit Card";

QN.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.

Query→

Select ba.Account_Number, Balance_amount, Transaction_amount,

Transaction_Date from Bank_Account_Details ba inner join bank_account_transaction bat on ba.account_number = bat.account_number And

(Transaction_Date between "2020-03-01" and "2020-04-30");

QN.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 .

Query→Select ba.Customer_id, ba.Account_Number, Balance_amount, Transaction_amount, Transaction_Date from Bank_Account_Details ba Left join bank_account_transaction bat on ba.account_number = bat.account_number And NOT (date_format(Transaction_Date , '%Y-%m') = "2020-03");
