Sales Order Processing

1) APPLICATION.

M/s ABC and Co. Ltd. deals in machinery spare parts. They have branches all over the country. The branches employ marketing representatives for the territories under them. The representatives book orders from various customers.
The project involves building an application to book orders from the customers, and generate the reports.

2) TABLES.

The following tables should be created:-

Branch_mst

Branch_cd	Char(4)	Primary key
Branch_name	Varchar(25)	Not null

Mktg_territory_mst

Area_cd	Char(4)	Primary key
Area_name	Varchar(25)	Not null
Branch_cd	Char(4)	Not null

Mktg_rep_mst

Rep_cd	Char(4)	Primary key
Rep_name	Varchar(25)	Not null
Area_cd	Char(4)	Not null

Customer_mst

Cust_cd	Char(4)	Primary key
Cust_name	Varchar(25)	Not null

Product_mst

Prod_cd	Char(4)	Primary key
Prod_name	Varchar(25)	Not null
Qty_on_hand	Int(6)	Not null
Reorder_level	Int(6)	Not null
Maximum_level	Int(6)	Not null
Booked_qty	Int(6)	Not null

Order_mst

Branch_cd	Char(4)	
Order_no	Int(6)	
Order_dt	Date	Not null
Cust_cd	Char(4)	Not null
Rep_cd	Char(4)	Not null
Order_status	Char(1)	Not null
Valid_upto	Date	Not null
Remarks	Varchar(200)	

Order_dtl

Branch_cd	Char(4)	
Order_no	Int(6)	
Prod_cd	Char(4)	Not null
Qty	Int(3)	Not null
Rate	Float(6,2)	Not null

3) VALIDITY CHECKS.

The following checks should be maintained:-

Mktg_territory_mst

Branch_cd must be a valid branch.

Mktg_rep_mst

Area_cd must be a valid marketing territory.

Product_mst

Booked_qty must not be negative.

Order_mst

Branch_cd and Order_no together is the Primary key.
Branch_cd must be a valid branch.
Cust_cd must be a valid customer.
Rep_cd must be a valid marketing representative.
Order_status can be Cancelled, Delivered or Pending. Default value is P.
Valid_upto should be at least one week later than Order_dt.

Order_dtl

Branch_cd, Order_no and Prod_cd together is the Primary key.
Branch_cd and Order_no must represent a valid order.
Prod_cd must be a valid product.
Qty and Rate must not be negative.

General

Booked_qty in Product_mst must reflect the total Quantity for pending orders at all times.
Qty booked cannot be greater than Qty_on_hand.
Reorder Level cannot be greater than the Maximum Level.
No order can be modified once it has been Cancelled or Delivered.

Write **Database triggers** to update the Booked_qty and Qty_on_hand in the Product_mst.

4) USER INPUT.

The following Forms should be created:-

(a) Programs for maintaining (inserting/updating/deleting) each of the following masters:-

1) Branch_mst
2) Mktg_territory_mst
3) Mktg_rep_mst
4) Customer_mst
5) Product_mst

(a) Programs for accepting new orders and modifying pending orders.

The programs should display one order at a time,
It should display all items for the order,
It should display the customer's name for the order,
It should display the product names for each item,
It should display the amount for each item, It
should display the total amount for the order, It
should perform the necessary validity checks.

(b) A separate program for marking orders as Cancelled or Delivered.

This program is similar to the previous one. No changes other than updating the orders as Cancelled or Delivered are to be allowed.

5) REPORTS.

(a) Summary reports for listing order totals:-

1) Area-wise.
2) Rep-wise.
3) Customer-wise.
4) Product-wise.

(a) A detail report to list order totals:-

Branch-wise, Area-wise, Rep-wise, Product-wise With summary totals at each level.

(b) A matrix report to print area-wise product-wise sales summary.

The report should list Total Quantity and Average Price.

(c) A report to print Pending, Cancelled, Delivered or All orders during a given period. The type of order and the period to be delivered by depends on runtime parameters. The report should list all the details of an order including Customer Name, Product Name, Item-wise Totals and Total Order Value.

(d) A report to print all the items with the quantity below the reorder level.

GUIDELINES.

1. Create necessary indexes to make your queries work faster. Implement this for any 2 tables.
2. Some queries and DML operations could be performed on Views. Implement this for any 2 tables.
3. Use auto_increment where required (e.g. Branch_cd column of Branch_mst). If necessary, the sequences could be concatenated with character strings to produce alphanumeric codes.
4. Use Stored Procedures, and Stored Functions where necessary.
5. The Reports have to be created using SQL (and MySQL PL if required).