1) Create Database name ‘project1’

```
CREATE DATABASE PROJECT1;
```

2) Create table with all the column name as data type ‘Varchar’

```
create table SALES_DATASET_RFM_PRJ(
  ordernumber VARCHAR(200),
  quantityordered VARCHAR(200),
  priceeach        VARCHAR(200),
  orderlinenumber  VARCHAR(200),
  sales            VARCHAR(200),
  orderdate        VARCHAR(200),
  status           VARCHAR(200),
  productline      VARCHAR(200),
  msrp             VARCHAR(200),
  productcode      VARCHAR(200),
  customername     VARCHAR(200),
  phone            VARCHAR(200),
  addressline1     VARCHAR(200),
  addressline2     VARCHAR(200),
  city             VARCHAR(200),
  state            VARCHAR(200),
  postalcode       VARCHAR(200),
  country          VARCHAR(200),
  territory        VARCHAR(200),
  contactfullname  VARCHAR(200),
  dealsize         VARCHAR(200)
)
```

3) Convert all the datatype to it correct form(Before all of it is varchar)

```
/* 3) Now as all the table data_type is VARCHAR we need to ALTER the data_type to match
with all the column*/

/*a) Ordernumber column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN ordernumber numeric;

/*b) quantityordered column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN quantityordered numeric;
```

```
/*c) priceeach column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN priceeach numeric;

/*d) orderlinenumber column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN orderlinenumber numeric;

/*e) sales column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN sales decimal;

/*f) orderdate column*/
SET SQL_SAFE_UPDATES = 0;
UPDATE sales_dataset_rfm_prj
SET orderdate = STR_TO_DATE(orderdate, '%m/%d/%Y %H:%i')
WHERE orderdate LIKE '%/%/% %:%';
SET SQL_SAFE_UPDATES = 1;
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN orderdate datetime;

/*g)msrp column*/
ALTER TABLE sales_dataset_rfm_prj
MODIFY COLUMN msrp numeric;
```

4) Check Null/Blank in every column

```
SELECT COUNT(*)
FROM sales_dataset_rfm_prj
WHERE orderlinenumber IS NULL OR orderlinenumber = ' ';
```

=> I have checked for every column in the table and seen that there is no null and blank

5) Add in column CONTACTLASTNAME, CONTACTFIRSTNAME that been taken break out from CONTACTFULLNAME. We have to capitalize the first letter and all lower letter for the rest.

```
/* Add in column CONTACTLASTNAME, CONTACTFIRSTNAME and Capitalize the first letter*/

/*a) add contactlastname column*/
ALTER TABLE sales_dataset_rfm_prj
ADD column contactlastname varchar(100);
```

```
]/*b) add contactfirstname column*/
ALTER TABLE sales_dataset_rfm_prj
ADD column contactfirstname varchar(100);

/*c) update contactlastname column*/
UPDATE sales_dataset_rfm_prj
SET contactfirstname = SUBSTRING_INDEX(contactfullname, '-', 1);

/*d) update contactfirstname column*/
UPDATE sales_dataset_rfm_prj
SET contactlastname = SUBSTRING_INDEX(contactfullname, '-', -1);

/*e) Capitalize the first letter and lower all letter inside column*/
SET SQL_SAFE_UPDATES = 0;
UPDATE sales_dataset_rfm_prj
SET
    contactfirstname = CONCAT(
        UPPER(SUBSTRING(contactfirstname, 1, 1)),
        LOWER(SUBSTRING(contactfirstname, 2))
    ),
    contactlastname = CONCAT(
        UPPER(SUBSTRING(contactlastname, 1, 1)),
        LOWER(SUBSTRING(contactlastname, 2))
    )
WHERE
    contactfirstname IS NOT NULL AND contactfirstname != ''
    AND contactlastname IS NOT NULL AND contactlastname != '';

SET SQL_SAFE_UPDATES = 1;
```

=> To work on this problem I first create two new column name ‘contactlastname’ , ‘contactfirstname’. Then I use Substring Index function to take all the values from contactfirstname until its delimiter ‘-’ and assign that values for column contactfirstname. And vice versa for the contactlastname.

=> Then I use Upper and Lower function with Substring function to Capitalize the first letter and Lower the rest letter.

/*6) Extract Month, year, Quarter from orderdate column*/

```
SET SQL_SAFE_UPDATES = 0;
```

```
ALTER TABLE sales_dataset_rfm_prj
ADD COLUMN order_year INT;

ALTER TABLE sales_dataset_rfm_prj
ADD COLUMN order_month INT;

ALTER TABLE sales_dataset_rfm_prj
ADD COLUMN order_quarter INT;

UPDATE sales_dataset_rfm_prj
SET
   order_year = YEAR(orderdate),
   order_month = MONTH(orderdate),
   order_quarter = QUARTER(orderdate);
SET SQL_SAFE_UPDATES = 1;
```

/*7) Finding outlier for column Quantity Ordered*/
/*Using IQR/BOX Plot data to find out outliers*/

```
SET SQL_SAFE_UPDATES = 0;
SET @avg_quantity = (SELECT AVG(quantityordered) FROM sales_dataset_rfm_prj);
WITH IQR_Min_Max AS (
   SELECT
      MAX(CASE WHEN quartile = 1 THEN value END) AS Q1,
      MAX(CASE WHEN quartile = 3 THEN value END) AS Q3,
      (MAX(CASE WHEN quartile = 3 THEN value END) - MAX(CASE WHEN quartile = 1 THEN
value END)) AS IQR,
      MAX(CASE WHEN quartile = 1 THEN value END) - (1.5 * (MAX(CASE WHEN quartile = 3
THEN value END) - MAX(CASE WHEN quartile = 1 THEN value END))) AS Min_value,
      MAX(CASE WHEN quartile = 3 THEN value END) + (1.5 * (MAX(CASE WHEN quartile = 3
THEN value END) - MAX(CASE WHEN quartile = 1 THEN value END))) AS Max_value
   FROM (
      SELECT
         quantityordered AS value,
         NTILE(4) OVER (ORDER BY quantityordered) AS quartile
      FROM
         sales_dataset_rfm_prj
   ) AS quartile_data
)
UPDATE sales_dataset_rfm_prj
SET quantityordered = @avg_quantity
WHERE quantityordered < (SELECT Min_value FROM IQR_Min_Max)
  OR quantityordered > (SELECT Max_value FROM IQR_Min_Max);
```

```
SET SQL_SAFE_UPDATES = 1;
```

/*8) After the data clean then save it into new table named SALES_DATASET_RFM_PRJ_CLEAN */

```
CREATE TABLE SALES_DATASET_RFM_PRJ_CLEAN AS
SELECT *
FROM SALES_DATASET_RFM_PRJ
```

/*9) Check to see if all the outliers has been deleted*/

```
SELECT *
FROM SALES_DATASET_RFM_PRJ_CLEAN
Where quantityordered < 67;
```