1. Create the following tables with the mentioned constraints.(See the Excel file attached Tables.xls (Question 1 Sheet) )

CREATE TABLE items\_sub  
  (  
     items\_id           *INT* PRIMARY KEY,  
     items\_sub\_desc     *VARCHAR*(25) NOT NULL,  
     item\_expiry\_date   *DATETIME*,  
     item\_creation\_date *DATETIME*,  
     amount             *INT* CHECK(amount>0)  
  );  
  
CREATE TABLE dept  
  (  
     dept\_id   *INT* PRIMARY KEY,  
     dept\_desc *VARCHAR*(25) NOT NULL  
  );  
  
CREATE TABLE makers  
  (  
     maker\_id      *INT* PRIMARY KEY,  
     maker\_name    *VARCHAR*(25) NOT NULL,  
     maker\_city    *VARCHAR*(25),  
     maker\_address *VARCHAR*(25),  
     maker\_pincode *VARCHAR*(25),  
     maker\_phone   *VARCHAR*(25)  
  );  
  
CREATE TABLE items

(

item\_id INT PRIMARY KEY,

item\_desc VARCHAR(25) NOT NULL,

item\_sub\_id INT FOREIGN KEY REFERENCES items\_sub(items\_id),

quantity INT,

item\_code VARCHAR(25) UNIQUE,

item\_dept\_id INT FOREIGN KEY REFERENCES dept(dept\_id),

item\_maker\_id INT FOREIGN KEY REFERENCES makers(maker\_id),

booked\_date DATETIME

);

1. Have a look at the data below(See the Excel file attached Tables.xls (Question 2 Sheet) )

2.a. Insert the First Row of Item table into the table without the columns specified in the insert clause.

INSERT INTO items  
VALUES      (100,  
             ' Soap Liquids ',  
             50,  
             2,  
             ' PRO100 ',  
             1000,  
             1,  
             '01-01-12')

2.b Insert the Second Row of Item Table into the table with the columns specified in the insert clause explicitly and confirm the data inserted.

We can’t insert the values for the second row because it will not accept duplicate values,because it is a primary key

2.c Insert the third row, fourth row and Fifth of items table as it is in the table output

INSERT INTO items  
VALUES      (101,  
             ' Biscuits ',  
             60,  
             1,  
             ' PRO101 ',  
             1001,  
             2,  
             '01-12-11 ' ),  
            ( 102,  
              ' Drinks ',  
              70,  
              4,  
              ' 102DR ',  
              1001,  
              3,  
              '01-10-11' ),  
            ( 103,  
              ' Soap Liquids ',  
              51,  
              1,  
              ' PRO103 ',  
              1000,  
              4,  
              '01-04-12 ' );

2.d Update the Item Code for the Item Id 102 to PRO102.’

update items set Item\_Code =’PRO102’

WHERE Item\_id=102;

2.e Select the Id and desc and maker id of all the items

select Item\_id,Item\_desc,Item\_Maker\_id from Items;

2.f Select the Id , ItemDesc where the quantity is more than 2

select Item\_id,Item\_Desc

from items

where Quantity >2;

2.g Calculate a Quantity increase of 5 for all the items and display them in a new column and order them by maximum.

alter table items

ADD NEW\_QUANTITY int;

UPDATE items set NEW\_QUANTITY=Quantity+5;

Select \* from items

ORDER BY NEW\_QUANTITY DESC;

2.h Calculate the Monthly Gross Quantity along with 5 quantity added to all the item (for example for Item id 100, the qty is 2 for Jan 01, aft qty added is 7, the qty for Jan 02 is 3, aft qty added is 8, so monthly gross for Jan is 15) in a separate column with the tile Monthly Gross

select datepart(month,Booked\_Date) AS MONTH,(Quantity+NEW\_QUANTITY) as monthly\_gross from items

group by datepart(month,Booked\_Date);

2.i Display all the items that are brought in 2009

select \* from items

where

datediff(year,Booked\_Date,2009)=0; # we won’t get the output because there is no uear like 2009 we will get empty table

2.j Display the items that are brought between 1 Jun 2011 and 1 Dec 2012

select \* from items

where

Booked\_Date between '2011/01/01' and '2012/12/01';

2.k Display the Items that are made by maker Id 2 and the booked date between Jan 2010 and Dec 2011

select \*

from items

where Item\_Maker\_id=2 and (Booked\_Date between '2010/01/01' and '2011/12/01');

2.l Display all the item\_id,items\_desc based on the latest booked date order.

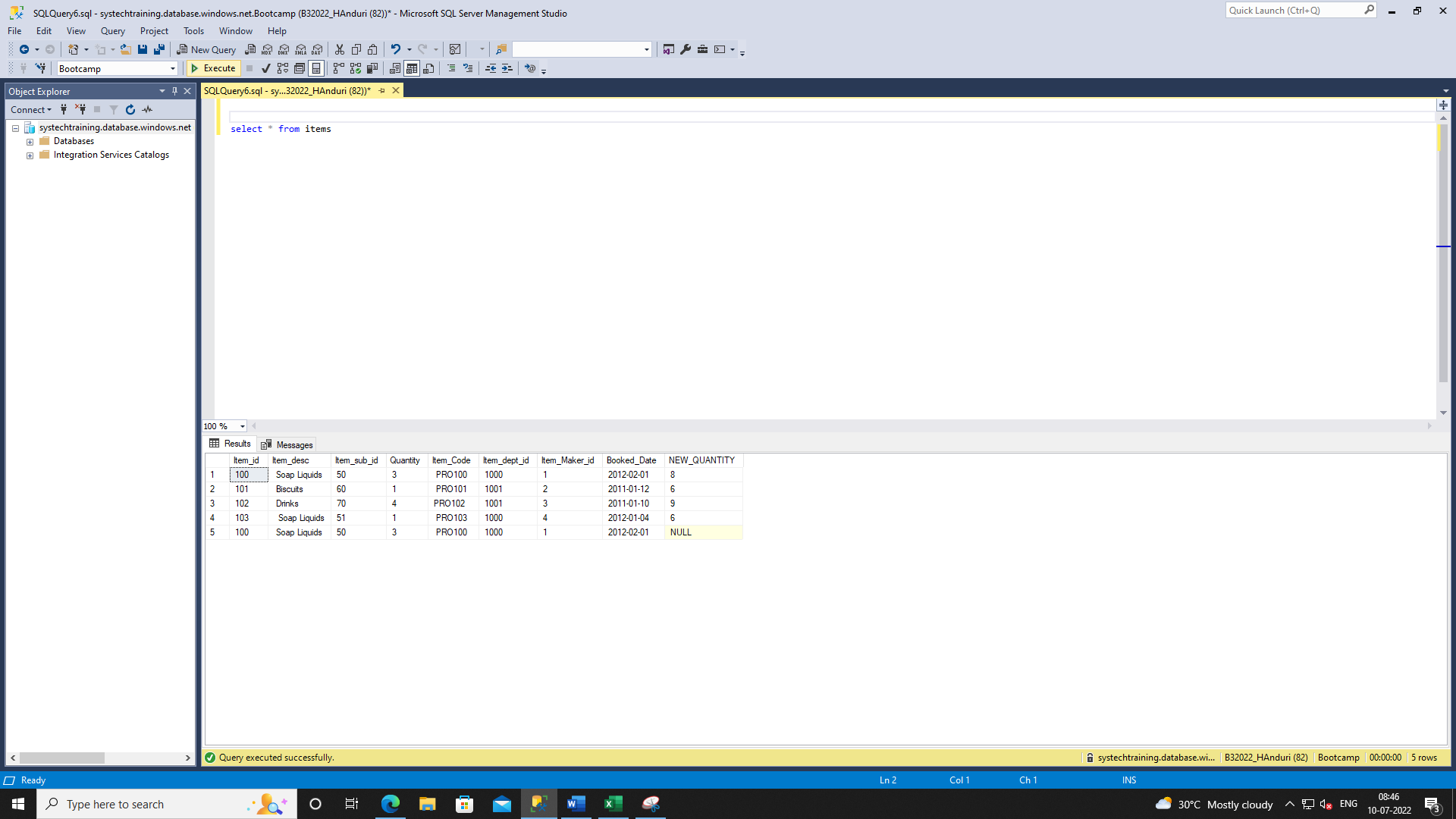
select \*

from items

order by Booked\_Date ;

2.m Check all the constraints are working (Include screenshots)

select \* from items



2.n Create a table 'items\_dup' from 'Items' and perform the following below

Select top 0 \* into items\_dup from items;

Delete the Items that are belonging to Maker Id 2 from the table items\_dup

delete from items\_dup where

Item\_Maker\_id=2;

Delete the Items that are belonging to the Year 2012

delete from items\_dup where

datediff(year,Booked\_Date,2012)=0;

Delete the records from items\_dup

delete from items\_dup;