SQL - Assignment 1

(ITEMS AND PRODCUTS DATABASE)

**Basic Questions**

1. **Create two Databases Name :- Brands, and Products**

Syntax:

CREATE DATABASE BRANDS;

CREATE DATABASE PRODUCTS;

1. **Create two tables in SQL Server name as ITEMS\_TABLE in Brands database and PRODUCT\_TABLE in Products database.**

Syntax:

ITEMS\_TABLE in Brands database

USE BRANDS;

CREATE TABLE ITEMS\_TABLE

(ID INT IDENTITY(1,1) PRIMARY KEY,

ITEM\_DESCRIPTION VARCHAR(255),

VENDOR\_NOS INT,

VENDOR\_NAME VARCHAR(255),

BOTTLE\_SIZE INT,

BOTTLE\_PRICE DECIMAL(2,2));

PRODUCT\_TABLE in Products database

USE PRODUCTS;

CREATE TABLE PRODUCT\_TABLE

(

PRODUCT\_ID INT IDENTITY(1,1) PRIMARY KEY,

COUNTRY VARCHAR(255),

PRODUCTS VARCHAR(255),

UNITS\_SOLD DECIMAL(6,2),

MANUFACTURING\_PRICE INT,

SALE\_PRICE INT,

GROSS\_SALES BIGINT,

SALES BIGINT,

COGS BIGINT,

PROFIT BIGINT,

DATES DATE,

MONTH\_NUMBER INT,

MONTH\_NAME VARCHAR(255),

YEARS INT);

1. Records Insertion in ITEMS\_TABLE

USE BRANDS;

--- to avoid Arithmetic overflow error converting numeric to data type numeric.---

ALTER TABLE ITEMS\_TABLE ALTER COLUMN BOTTLE\_PRICE DECIMAL(6,2);

INSERT INTO ITEMS\_TABLE

VALUES

('Travis Hasse Apple Pie',305,'Mhw Ltd',750,9.77),

('D aristi Xtabentun',391,'Anchor Distilling (preiss Imports)', 750,14.12),

('Hiram Walker Preach Brandy',370,'Pernod Ricard Usa/Austin Nichols',1000,6.5),

('Oak Cross Whisky', 305, 'Mhw Ltd',750,25.33),

('Uv Red(cherry) Vodka', 380,'Philips Beverage Company',200,1.97),

('Heaven Hill Old Style White Label', 259, 'Heaven Hill Distilleries Inc.',750,6.37),

('Hyde Herbal Liqueur', 194,'Fire Tail Brands Llc',750,9.06),

('Dupont Calvados Fine Resetve',403, 'Robert Kacher Selections', 750,23.61);

--- RECORDS INSERTION PRODUCT\_TABLE----

USE PRODUCTS;

INSERT INTO PRODUCT\_TABLE

Values

('Canada', 'Carretera', 1618.5,3,20,32370,32370,16185,16185,'1/1/2014',1,'January',2014),

('Germany','Carretera',1321,3,20,26420,26420,13210,13210,'1/1/2014',1,'January',2015),

('France','Carretera',2178,3,15,32670,32670,271780,10890,'6/1/2014',6,'June',2016),

('Germany','Carretera',888,3,15,13320,13320,8880,4440,'6/1/2014',6,'June',2017),

('Mexico','Carretera',2470,3,15,37050,37050,24700,12350,'6/1/2014',6,'June',2018),

('Germany','Carretera',1513,3,250,529550,529550,393380,136170,'12/1/2014',12,'December',2019),

('Germany','Montana',921,5,15,13815,13815,9210,4605,'3/1/2014',3,'March',2020),

('Germany','Montana',2518,5,12,30216,30216,7554,22662,'6/1/2014',6,'June',2020);

1. **Delete those product having the Units Sold 1618.5, 888 and 2470.**

Syntax:

DELETE FROM PRODUCT\_TABLE WHERE UNITS\_SOLD IN (1618.5,888,2470);

1. **Select all records from the bottle\_info table.**

Syntax:

SELECT \* FROM bottle\_info;

1. **Select the item\_description and bottle\_price for all items in the bottle\_info table**

Syntax:

SELECT item\_description, bottle\_price FROM bottle\_info;

1. **Find the item\_description and bottle\_price of items where bottle\_price is greater than 20**

Syntax:

SELECT item\_description, bottle\_price FROM bottle\_info WHERE bottle\_price>20;

1. **Select Unique Country from the product\_sales table**

Syntax:

SELECT DISTINCT(COUNTRY) FROM PRODUCT\_TABLE;

1. **Count the number of Countries in the product\_sales table**

Syntax:

SELECT COUNT(COUNTRY) AS NUM\_OF\_COUNTRIES FROM PRODUCT\_TABLE;

1. **How Many Countries are there which contain the sales price between 10 to 20**

Syntax:

SELECT COUNT(COUNTRY) AS COUNTRY\_COUNT FROM PRODUCT\_TABLE WHERE SALE\_PRICE BETWEEN 10 AND 20;

**Intermediate Questions**

1. **Find the Total Sale Price, and Gross Sales**

Syntax:

SELECT PRODUCT\_ID, (UNITS\_SOLD\*MANUFACTURING\_PRICE\*SALE\_PRICE) AS TOTAL\_SALE\_PRICE, GROSS\_SALES FROM PRODUCT\_TABLE;

1. **In which year we have got the highest sales**

Syntax:

SELECT YEARS FROM PRODUCT\_TABLE WHERE SALES=(SELECT MAX(SALES) FROM PRODUCT\_TABLE);

1. **Which Product having the sales of $ 37,050.00**

**NOTE:** AS A PART OF A QUESTION A FEW ROWS WERE DELETED, I HAVE INSERTED THE DATA AGAIN AFTER TRUNCATING THE TABLE.

Syntax:

SELECT PRODUCTS FROM PRODUCT\_TABLE WHERE SALES=37050;

1. **Which Countries lies between profit of $ 4,605 to $ 22 ,662.00**

Syntax:

SELECT COUNTRY FROM PRODUCT\_TABLE WHERE PROFIT BETWEEN 4605 AND 22662;

1. **Which Product Id having the sales of $ 24 ,700.00**

Syntax:

SELECT PRODUCT\_ID FROM PRODUCT\_TABLE WHERE SALES=24700;

1. **Find the total Units Sold for each Country.**

Syntax:

SELECT COUNTRY, SUM(UNITS\_SOLD) AS UNITS\_SOLD FROM PRODUCT\_TABLE GROUP BY COUNTRY;

1. **Find the average sales for each country**

Syntax:

SELECT COUNTRY, AVG(SALE\_PRICE) AS AVG\_SALES FROM PRODUCT\_TABLE GROUP BY COUNTRY;

1. **Retrieve all products sold in 2014**

Syntax:

SELECT PRODUCTS FROM PRODUCT\_TABLE WHERE YEARS=2014;

1. **Find the maximum Profit in the product\_sales table.**

Syntax:

SELECT MAX(PROFIT) AS MAX\_PROFIT FROM PRODUCT\_TABLE;

1. **Retrieve the records from product\_sales where Profit is greater than the average Profit of all records.**

Syntax:

SELECT \* FROM PRODUCT\_TABLE WHERE PROFIT> (SELECT AVG(PROFIT) FROM PRODUCT\_TABLE);

1. **Find the item\_description having the bottle size of 750**

Syntax:

USE BRANDS;

SELECT ITEM\_DESCRIPTION FROM ITEMS\_TABLE WHERE BOTTLE\_SIZE=750;

1. **Find the vendor Name having the vendor\_nos 305, 380, 391**

Syntax:

SELECT VENDOR\_NAME FROM ITEMS\_TABLE WHERE VENDOR\_NOS IN (305,380,391);

1. **What is total Bottle\_price**

Syntax:

SELECT SUM(BOTTLE\_PRICE) AS TOTAL\_BOTTLE\_PRICE FROM ITEMS\_TABLE;

1. **Make Primary Key to Item\_id**

Syntax:

ALTER TABLE ITEMS\_TABLE ADD PRIMARY KEY (ITEM\_ID);

1. **Which item id having the bottle\_price of $ 5.06**

SYNTAX:

SELECT ID FROM ITEMS\_TABLE WHERE BOTTLE\_PRICE=5.06;

**Advance Questions**

1. **Apply INNER , FULL OUTER , LEFT JOIN types on both the table**

**INNER JOIN**

SYNTAX:

SELECT IT.\*,PT.\* FROM [BRANDS].[dbo].[ITEMS\_TABLE] AS IT JOIN [PRODUCTS].[dbo].[PRODUCT\_TABLE] AS PT ON IT.ID=PT.PRODUCT\_ID;

**FULL OUTER JOIN:**

SYNTAX:

SELECT IT.\*,PT.\* FROM [BRANDS].[dbo].[ITEMS\_TABLE] AS IT FULL OUTER JOIN [PRODUCTS].[dbo].[PRODUCT\_TABLE] AS PT ON IT.ID=PT.PRODUCT\_ID;

**LEFT JOIN**

SYNTAX:

SELECT IT.\*,PT.\* FROM [BRANDS].[dbo].[ITEMS\_TABLE] AS IT LEFT JOIN [PRODUCTS].[dbo].[PRODUCT\_TABLE] AS PT ON IT.ID=PT.PRODUCT\_ID;

1. **Find the item\_description and Product having the gross sales of 13,320.00**

SYNTAX

SELECT IT.\*,PT.\* FROM [BRANDS].[dbo].[ITEMS\_TABLE] AS IT JOIN [PRODUCTS].[dbo].[PRODUCT\_TABLE] AS PT ON IT.ID=PT.PRODUCT\_ID WHERE PT.GROSS\_SALES=13320;

1. **Split the Item\_description Column into Columns Item\_desc1 and Item\_desc2**

SYNTAX:

USE BRANDS;

SELECT LEFT(ITEM\_DESCRIPTION,LEN(ITEM\_DESCRIPTION)/2) AS ITEM\_DESC1, RIGHT(ITEM\_DESCRIPTION,LEN(ITEM\_DESCRIPTION)/2) AS ITEM\_DESC2 FROM ITEMS\_TABLE;

1. **Find the top 3 most expensive items in the bottle\_info table**

SYNTAX:

SELECT TOP 3 BOTTLE\_PRICE, ITEM\_DESCRIPTION FROM ITEMS\_TABLE ORDER BY BOTTLE\_PRICE DESC;

1. **Find the total Gross Sales and Profit for each Product in each Country.**

SYNTAX:

SELECT COUNTRY, SUM(Gross\_Sales) AS Total\_Gross\_Sales, SUM(PROFIT) AS TOTAL\_PROFIT FROM PRODUCT\_TABLE GROUP BY COUNTRY ORDER BY COUNTRY;

1. **Find the vendor\_name and item\_description of items with a bottle\_size of 750 and bottle\_price less than 10.**

SYNTAX:

SELECT VENDOR\_NAME,ITEM\_DESCRIPTION FROM ITEMS\_TABLE WHERE BOTTLE\_SIZE=750 AND BOTTLE\_PRICE<10;

1. **Find the Product with the highest Profit in 2019.**

SYNTAX:

SELECT MAX(PROFIT) AS MAX\_PROFIT FROM PRODUCT\_TABLE WHERE YEARS=2019;

1. **Retrieve the Product\_Id and Country of all records where the Profit is at least twice the COGS.**

SYNTAX:

SELECT PRODUCT\_ID, COUNTRY FROM PRODUCT\_TABLE WHERE PROFIT>=2\*COGS;

1. **Find the Country that had the highest total Gross Sales in 2018**

SYNTAX:

SELECT COUNTRY FROM PRODUCT\_TABLE WHERE GROSS\_SALES=(SELECT MAX(GROSS\_SALES) FROM PRODUCT\_TABLE WHERE GROSS\_SALES=(SELECT SUM(GROSS\_SALES) FROM PRODUCT\_TABLE WHERE YEARS=2018 GROUP BY YEARS ) );

1. **Calculate the total Sales for each Month Name across all years.**

SYNTAX:

SELECT MONTH\_NAME , SUM(SALES) AS SUM\_OF\_SALES FROM PRODUCT\_TABLE GROUP BY MONTH\_NAME;

1. **List the item\_description and vendor\_name for items whose vendor\_nos exists more than once in the bottle\_info table.**

SYNTAX:

SELECT VENDOR\_NAME, ITEM\_DESCRIPTION FROM ITEMS\_TABLE WHERE VENDOR\_NOS = (SELECT VENDOR\_NOS AS C\_VENDOR\_NOS FROM ITEMS\_TABLE GROUP BY VENDOR\_NOS HAVING COUNT(VENDOR\_NOS)>1 );

1. **Find the average Manufacturing Price for Product in each Country and only include those Country and Product combinations where the average is above 3**

SYNTAX:

SELECT PRODUCTS, COUNTRY, AVG(MANUFACTURING\_PRICE) AS AVG\_MANFACTURING\_PRICE FROM PRODUCT\_TABLE GROUP BY PRODUCTS, COUNTRY HAVING AVG(MANUFACTURING\_PRICE)>3;

**SUPER ADVANCE QUESTIONS**

1. **Find the item\_description and bottle\_price of items that have the same vendor\_name as items with Item\_Id 1**

SYNTAX:

SELECT ITEM\_DESCRIPTION, BOTTLE\_PRICE FROM ITEMS\_TABLE WHERE VENDOR\_NAME=(SELECT VENDOR\_NAME FROM ITEMS\_TABLE WHERE ID=1) AND ID!=1;

1. **Create a stored procedure to retrieve all records from the bottle\_info table where bottle\_price is greater than a given value**

SYNTAX:

CREATE PROCEDURE BOTTLE\_INFO @BOTTLEPRICE DECIMAL(6,2)

AS

SELECT \* FROM ITEMS\_TABLE WHERE BOTTLE\_PRICE>@BOTTLEPRICE

GO

EXEC BOTTLE\_INFO @BOTTLEPRICE=2.00

1. **Create a stored procedure to insert a new record into the product\_sales table.**

SYNTAX:

CREATE PROCEDURE PRODUCT\_NEW\_RCD

@COUNTRY NVARCHAR(255)=NULL,

@PRODUCTS NVARCHAR(255)=NULL,

@UNITSolds DECIMAL(6,2)=NULL,

@MFG\_PRICE INT=NULL,

@SALEPrice INT=NULL,

@GROSSSales BIGINT=NULL,

@SALE BIGINT=NULL,

@COGS BIGINT=NULL,

@PROFIT BIGINT=NULL

AS

SET NOCOUNT ON

INSERT INTO [dbo].[PRODUCT\_TABLE]

([COUNTRY],[PRODUCTS],[UNITS\_SOLD],[MANUFACTURING\_PRICE],[SALE\_PRICE],[GROSS\_SALES],[SALES],[COGS],[PROFIT])

VALUES

(

@COUNTRY, @PRODUCTS, @UNITSolds, @MFG\_PRICE, @SALEPrice, @GROSSSales, @SALE, @COGS, @PROFIT)

GO

EXEC PRODUCT\_NEW\_RCD @COUNTRY ='USA', @PRODUCTS='MONTANA', @UNITSolds=245.00, @MFG\_PRICE=2, @SALEPrice=10, @GROSSSales=54389, @SALE=54389, @COGS=1239, @PROFIT=7894

1. **Create a trigger to automatically update the Gross\_Sales field in the product\_sales table whenever Units\_Sold or Sale\_Price is updated.**

SYNTAX:

CREATE TRIGGER UPDATE\_GROSS\_SALES

ON [dbo].[PRODUCT\_TABLE]

AFTER UPDATE

AS BEGIN

SET NOCOUNT ON;

IF (SUBSTRING(COLUMNS\_UPDATED(),4,1) & CAST (0x20 AS int)=0\*20)

BEGIN

UPDATE [dbo].[PRODUCT\_TABLE]

SET [GROSS\_SALES]= (I.[UNITS\_SOLD] \* I.[SALE\_PRICE])

FROM [dbo].[PRODUCT\_TABLE] P INNER JOIN INSERTED I

ON P.PRODUCT\_ID=I.PRODUCT\_ID

WHERE CHECKSUM(P.UNITS\_SOLD,P.SALE\_PRICE)<> CHECKSUM(I.UNITS\_SOLD,I.SALE\_PRICE)

END

END

UPDATE [dbo].[PRODUCT\_TABLE]

SET [UNITS\_SOLD]=7, [SALE\_PRICE]=3

WHERE PRODUCT\_ID=9;

1. **Write a query to find all item\_description in the bottle\_info table that contain the word "Whisky" regardless of case.**

SYNTAX:

SELECT \* FROM [dbo].[ITEMS\_TABLE] WHERE ITEM\_DESCRIPTION LIKE '%Whisky%';

1. **Write a query to find the Country and Product where the Profit is greater than the average Profit of all products.**

SYNTAX:

SELECT COUNTRY,PRODUCTS FROM PRODUCT\_TABLE WHERE PROFIT> (SELECT AVG(PROFIT) FROM PRODUCT\_TABLE);

1. **Write a query to join the bottle\_info and product\_sales tables on a common field (e.g., vendor\_nos) and select relevant fields from both tables.**

SYNTAX:

SELECT \* FROM [PRODUCTS].[dbo].[PRODUCT\_TABLE] AS P JOIN [BRANDS].[dbo].[ITEMS\_TABLE] AS I ON P.PRODUCT\_ID=I.ID;

1. **Write a query to combine item\_description and vendor\_name into a single string for each record in the bottle\_info table, separated by a hyphen.**

INDEX:

SELECT [ITEM\_DESCRIPTION]+' - '+[VENDOR\_NAME] AS NAME FROM [dbo].[ITEMS\_TABLE];

1. **Write a query to display the bottle\_price rounded to one decimal place for each record in the bottle\_info table.**

SYNTAX:

SELECT CAST(ROUND(BOTTLE\_PRICE,1,1) AS decimal(6,1)) AS BOTTLE\_PRICE FROM [dbo].[ITEMS\_TABLE];

1. **Write a query to calculate the number of days between the current date and the Date field for each record in the product\_sales table.**

SYNTAX:

SELECT DATEDIFF(DAY,DATES,GETDATE()) as Days\_Diff FROM [dbo].[PRODUCT\_TABLE];