#Discussion 1 Task 1

```sql
USE PUBS;
SELECT * FROM titles;
SELECT title,ytd_sales FROM titles WHERE ytd_sales > 8000;
```

#Discussion 1 Task 2

```sql
USE PUBS;
SELECT * FROM titles;
SELECT title,royalty FROM titles WHERE royalty>12 AND royalty<24;
```

#Discussion 1 Task 3

```sql
USE PUBS;
SELECT * FROM titles;
SELECT type,AVG(price) AS AveragePrice,SUM(ytd_sales) AS TotalYearlySales
FROM titles GROUP BY type;
```

#Discussion 2 Task 1(i)

```sql
USE PUBS;
SELECT  title,CONCAT(au_fname,' ',au_lname) AS AuthorName FROM titles JOIN
titleauthor ON titles.title_id = titleauthor.title_id JOIN authors ON
titleauthor.au_id = authors.au_id;
```

#Discussion 2 Task 1(ii)

```
USE PUBS;

SELECT  title,CONCAT(au_fname,' ',au_lname) AS AuthorName,pub_name FROM titles JOIN

titleauthor ON titles.title_id = titleauthor.title_id JOIN authors ON

titleauthor.au_id = authors.au_id JOIN publishers ON titles.pub_id = publishers.pub_id;
```

#Discussion 2 Task 2

```
USE PUBS;

SELECT CONCAT(au_fname,' ',au_lname) AS AuthorName,authors.city,pub_name FROM

authors,publishers WHERE authors.city = publishers.city;
```

Another method

```
USE PUBS;

SELECT CONCAT(au_fname,' ',au_lname) AS AuthorName,authors.city,pub_name FROM

authors JOIN publishers ON authors.city = publishers.city;
```

#Discussion 2 Task 3

```
USE PUBS;
SELECT title,CONCAT(au_fname,' ',au_lname)
AS AuthorName,royalty FROM authors  JOIN titleauthor
ON authors.au_id  = titleauthor.au_id JOIN titles
ON titleauthor.title_id =titles.title_id
WHERE royalty = (SELECT MAX(royalty) FROM titles);
```

#Create Table CustomerAndSuppliers

```
USE Assignment;
CREATE TABLE CustomerAndSuppliers(
cust_id CHAR(6) PRIMARY KEY CHECK (cust_id LIKE '[CS][0-9][0-9][0-9][0-9][0-9]'),
cust_fname CHAR(15) NOT NULL,
cust_lname VARCHAR(15),
cust_address TEXT,
cust_telno CHAR(12) CHECK (cust_telno LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
cust_city CHAR(12) DEFAULT 'Rajshahi',
sales_amnt MONEY CHECK (sales_amnt > = 0),
proc_amnt MONEY CHECK (proc_amnt > = 0)
);
```

```
INSERT INTO CustomerAndSuppliers(cust_id,cust_fname,cust_lname,cust_address,cust_telno,cust_city,sales_amnt,proc_amnt)

VALUES('C00001','Hasem','Ali','Lebubagan,Binodpur','017-95125465','Rajshahi',0,0);

INSERT INTO CustomerAndSuppliers(cust_id,cust_fname,cust_lname,cust_address,cust_telno,cust_city,sales_amnt,proc_amnt)

VALUES('C00002','Nurnobi','Mia','Lebubagan,Binodpur','017-60340575','Rajshahi',0,0);

INSERT INTO CustomerAndSuppliers(cust_id,cust_fname,cust_lname,cust_address,cust_telno,cust_city,sales_amnt,proc_amnt)

VALUES('C00003','Mehedi','Hasan','Lebubagan,Binodpur','017-74490826','Rajshahi',0,0);


SELECT * FROM CustomerAndSuppliers;
```

#Create Table Item

```
USE Assignment;

CREATE TABLE Item(

item_id CHAR(6) PRIMARY KEY CHECK (item_id LIKE '[P][0-9][0-9][0-9][0-9][0-9]'),

item_name CHAR(12) NOT NULL,

item_category CHAR(10),

item_price FLOAT CHECK (item_price > = 0),

item_qoh INT CHECK (item_qoh > = 0),

item_last_sold DATE DEFAULT GETDATE()
```

```sql
);
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00001','Phone','Electrical','25000','50','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00002','Laptop','Electrical','50000','30','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00003','Gears','Mechanical','2000','100','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00004','Bearings','Mechanical','500','150','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00005','Antivirus','Software','700','20','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00006','Browser','Software','400','30','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00007','Comics','Books','1200','50','2025-01-17');
INSERT INTO
item(item_id,item_name,item_category,item_price,item_qoh,item_last_sold)
VALUES('P00008','Novels','Books','1500','50','2025-01-17');


SELECT * FROM Item;
```

#Create Table Transactions

```
USE Assignment;
CREATE TABLE Transactions(
tran_id CHAR(10) PRIMARY KEY CHECK (tran_id LIKE '[T][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
item_id CHAR(6) REFERENCES Item(item_id),
cust_id CHAR(6) REFERENCES CustomerAndSuppliers(cust_id),
tran_type CHAR(1) CHECK (tran_type IN ('S','O')),
tran_quantity INT CHECK (tran_quantity > = 0),
tran_date DATETIME DEFAULT GETDATE()
);

SELECT * FROM Transactions;
```

#Discussion 3 Task 1

```
USE Assignment;
GO
CREATE PROCEDURE printDetails
AS
BEGIN
SELECT item_category,SUM(item_qoh) AS TotalItem,AVG(item_price) AS newAvgPrice FROM Item GROUP BY item_category
END

EXEC printDetails;
```

#Discussion 3 Task 2

```
USE Assignment;
GO
CREATE PROCEDURE printDetails1 @itemCategory CHAR(10),@itemPrice INT
AS
BEGIN
SELECT * FROM Item WHERE item_category = @itemCategory AND item_price<@itemPrice
END

EXEC printDetails1 @itemCategory = 'Electrical',@itemPrice =  50000 ;
```

#Discussion 3 Task 3

```
USE Assignment;
GO
CREATE PROCEDURE printDetails2 @itemCategory CHAR(10),@desiredAvgValue FLOAT
AS
BEGIN
DECLARE @totalPrice DECIMAL(10,2),@totalItem INT,@currentAvgPrice FLOAT
SELECT @totalPrice = SUM(item_price),@totalItem = COUNT(*) FROM Item WHERE item_category = @itemCategory
SET @currentAvgPrice = @totalPrice/@totalItem
```

```
WHILE @currentAvgPrice<@desiredAvgValue
BEGIN
UPDATE Item
SET item_price = item_price + 0.1*(item_price) WHERE item_category = @itemCategory
SELECT @totalPrice = SUM(item_price) FROM Item WHERE item_category = @itemCategory
SET @currentAvgPrice = @totalPrice/@totalItem
END
SELECT item_category,AVG(item_price) newAvgPrice FROM Item WHERE item_category = @itemCategory
GROUP BY item_category
END

EXEC printDetails2 @itemCategory = 'Books',@desiredAvgValue = 1500;
```

#Discussion 4 Task 1

```
USE Assignment;
GO
CREATE TRIGGER updateAmounts ON Transactions AFTER INSERT
AS
BEGIN
DECLARE @tranType CHAR(1)
DECLARE @tranQuantity INT
DECLARE @itemPrice DECIMAL(10,2)
DECLARE @custID CHAR(6)
```

```sql
SELECT @tranType = tran_type,@tranQuantity = tran_quantity,@custID = cust_id FROM INSERTED
IF (@tranType = 'S')
BEGIN
SELECT @itemPrice = item_price FROM Item WHERE item_id IN (SELECT item_id FROM INSERTED)
UPDATE CustomerAndSuppliers
SET sales_amnt = sales_amnt + (@itemPrice*@tranQuantity) WHERE cust_id = @custID
END
IF (@tranType = 'O')
BEGIN
SELECT @itemPrice = item_price FROM Item WHERE item_id IN (SELECT item_id FROM INSERTED)
UPDATE CustomerAndSuppliers
SET proc_amnt = proc_amnt + (@itemPrice*@tranQuantity) WHERE cust_id = @custID
END
END
```

#Table Customers

```
USE Shop;
CREATE TABLE Customers(
customer_id VARCHAR(4) PRIMARY KEY CHECK (customer_id LIKE '[C][0-9][0-9][0-9]'),
first_name VARCHAR(10) NOT NULL,
last_name VARCHAR(10) NOT NULL,
city VARCHAR(20) DEFAULT 'Rajshahi',
total_sale_quantity INT CHECK(total_sale_quantity > = 0)
);
INSERT INTO Customers(customer_id,first_name,last_name,city,total_sale_quantity)
VALUES('C001','Hasem','Ali','Rajshahi',0);
INSERT INTO Customers(customer_id,first_name,last_name,city,total_sale_quantity)
VALUES('C002','Nurnobi','Mia','Rajshahi',0);
INSERT INTO Customers(customer_id,first_name,last_name,city,total_sale_quantity)
VALUES('C003','Mehedi','Hasan','Rajshahi',0);
```

#Table Products

```
USE Shop;
CREATE TABLE Products(
product_id VARCHAR(4)  PRIMARY KEY CHECK (product_id LIKE '[P][0-9][0-9][0-9]'),
product_name VARCHAR(10) NOT NULL,
product_origin VARCHAR(10) CHECK (product_origin  IN ('Local','Foreign')),
price MONEY CHECK (price > = 0),
product_QOH INT
);
INSERT INTO Products(product_id,product_name,product_origin,price,product_QOH)
VALUES('P001','Mobile','Foreign',25000,50);
INSERT INTO Products(product_id,product_name,product_origin,price,product_QOH)
VALUES('P002','Fan','Local',500,100);
INSERT INTO Products(product_id,product_name,product_origin,price,product_QOH)
VALUES('P003','Laptop','Foreign',50000,70);
```

#Table Transactions

```
USE Shop;
CREATE TABLE Transactions(
tran_id VARCHAR(4) PRIMARY KEY CHECK(tran_id LIKE '[T][0-9][0-9][0-9]'),
tran_date DATE DEFAULT GETDATE(),
customer_id VARCHAR(4) REFERENCES Customers(customer_id),
product_id VARCHAR(4) REFERENCES Products(product_id),
quantity_sold INT CHECK (quantity_sold >  0)
);
```

#Stored Procedure

```
USE Shop;
SELECT * FROM Customers;
SELECT * FROM Products;
SELECT * FROM Transactions;
GO
CREATE PROCEDURE insertRecord @tran_id VARCHAR(4),@tran_date DATE,@cust_id VARCHAR(4),
@product_id VARCHAR(4),@quantity_sold INT
AS
BEGIN
DECLARE @product_qoh INT
DECLARE @product_price MONEY
DECLARE @paidAmount MONEY
DECLARE @amnt VARCHAR(10)
SELECT @product_qoh = product_QOH FROM Products
SELECT @product_price = price FROM Products WHERE product_id = @product_id

IF @quantity_sold <= @product_qoh
BEGIN
INSERT INTO Transactions(tran_id,tran_date,customer_id,product_id,quantity_sold)
VALUES(@tran_id,@tran_date,@cust_id,@product_id,@quantity_sold)
SET @paidAmount = (@quantity_sold * @product_price)
SET @amnt = @paidAmount;

PRINT('Paided Amount : ')+@amnt
END
ELSE
PRINT ('Item is not available')
END
EXEC insertRecord 'T001','2025-01-23','C001','P002',10;
```

#Trigger

```
USE Shop;
SELECT * FROM Customers;
SELECT * FROM Products;
GO
CREATE TRIGGER updateTables ON Transactions FOR INSERT
AS
BEGIN
DECLARE @quantity_sold INT
DECLARE @cust_id VARCHAR(4)
DECLARE @product_id VARCHAR(4)
SELECT @quantity_sold = quantity_sold,@cust_id = customer_id,@product_id = product_id FROM INSERTED
UPDATE Products
```

```
SET product_QOH = product_QOH - @quantity_sold WHERE product_id = @product_id

UPDATE Customers

SET total_sale_quantity = total_sale_quantity + @quantity_sold WHERE customer_id = @cust_id

END
```