# #Discussion 1 Task 1

USE PUBS;

SELECT \* FROM titles;

SELECT title,ytd\_sales FROM titles WHERE ytd\_sales > 8000;

# #Discussion 1 Task 2

USE PUBS;

SELECT \* FROM titles;

SELECT title,royalty FROM titles WHERE royalty>12 AND royalty<24;

# #Discussion 1 Task 3

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)

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,cu st\_city,sales\_amnt,proc\_amnt)

VALUES('C00001','Hasem','Ali','Lebubagan,Binodpur','01795125465','Rajshahi',0,0);

INSERT INTO

CustomerAndSuppliers(cust\_id,cust\_fname,cust\_lname,cust\_address,cust\_telno,cu st\_city,sales\_amnt,proc\_amnt)

VALUES('C00002','Nurnobi','Mia','Lebubagan,Binodpur','01760340575','Rajshahi',0,0);

INSERT INTO

CustomerAndSuppliers(cust\_id,cust\_fname,cust\_lname,cust\_address,cust\_telno,cu st\_city,sales\_amnt,proc\_amnt)

VALUES('C00003','Mehedi','Hasan','Lebubagan,Binodpur','01774490826','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() );

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)

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