**Technical Documentation – DSD2 A1**

Create Table SQL Statements

/\*

The Customer table will be used to store all informtion regarding customers

who have purchased items with BMC Apps'.

\*/

CREATE TABLE Customer(

CustomerNo BIGINT NOT NULL IDENTITY (1,1),/\*IDENTITY (1,1) Will be used to increment the CustomerNo key field each time a new record is created.\*/

Forename varchar(50) NOT NULL,/\*This field will be used to store the Customers forename\*/

Surname varchar(50) NOT NULL,/\*This field will be used to store the Customers Surename\*/

Email varchar(100) NOT NULL,/\*This field will store the customers email address\*/

ContactNumber varchar (64) NOT NULL,/\*This field will store the customers contact number\*/

CustomerAddress varchar(255) NOT NULL, /\*This field will store the customers address\*/

PRIMARY KEY (CustomerNo) /\*The primary key for the customers table will be CustomerNo as it will allow customer records to be uniquely identified\*/

);

/\*

The Paymet table will be used to store all informtion regarding Payments

which have been made to BMC Apps'.

\*/

CREATE TABLE Payment(

PaymentNo BIGINT NOT NULL IDENTITY (1,1),/\*IDENTITY (1,1) Will be used to increment the PaymentNo key field each time a new record is created.\*/

PaymentType varchar(50) NOT NULL,/\*Payment type will store the data which distinguishes if a payment made was cash or card\*/

PaymentAmount varchar(50) NOT NULL, /\*This field will store the amount the payment was for\*/

PaymentDate DATE NOT NULL,/\*This wll store te date the payment was made\*/

PRIMARY KEY (PaymentNo) /\*The primary key for this Payment table will be PaymentNo as it will allow each payment record to have its own identity\*/

);

/\*The Product table will be used to store all of the products which

BMC Apps' have to offer for their customers\*/

CREATE TABLE Product (

ProductNo BIGINT NOT NULL IDENTITY (1,1), /\*IDENTITY (1,1) Will be used to increment the ProductNo key field each time a new record is created.\*/

ProductName varchar(50) NOT NULL, /\*This field will store the name of the product\*/

ProductCost INT NOT NULL, /\*This field will store the cost of the product\*/

AppType varchar(50) NOT NULL, /\*This field will store the type of application which the product is \*/

ProductRelease DATE NOT NULL, /\*Product Release field will store the date which the product was released\*/

PRIMARY KEY (ProductNo)/\*The primary key for this Procuct table will be ProductNo as it will allow each product record to have its own identity\*/

);

CREATE TABLE MultiItem(

MultiItemNo BIGINT NOT NULL IDENTITY (1,1),

LicenceNumber BIGINT NOT NULL,

CustomerNo BIGINT NOT NULL,

ProductNo BIGINT NOT NULL,

OrderNo BIGINT NOT NULL,

PRIMARY KEY (MultiItemNo),

FOREIGN KEY (CustomerNo) REFERENCES Customer(CustomerNo),

FOREIGN KEY (ProductNo) REFERENCES Product(ProductNo),

FOREIGN KEY (OrderNo) REFERENCES ProductOrder(OrderNo)

);

/\*The ProductOrder table will be used to store all orders

which have been made to BMC Apps' \*/

CREATE TABLE ProductOrder(

OrderNo BIGINT NOT NULL IDENTITY (1,1), /\*IDENTITY (1,1) Will be used to increment the OrderNo key field each time a new record is created.\*/

OrderDate DATE NOT NULL, /\*The Order date field will store the date the order was made\*/

OrderCost varchar(20), /\*The order cost field will state the cost of the order\*/

CustomerNo BIGINT NOT NULL, /\*The customer No will identfy the customer who has made the order\*/

PaymentNo BIGINT NOT NULL, /\*The payment no will identify the payment record from this order\*/

PRIMARY KEY (OrderNo), /\*The primary key for this ProcuctOrder table will be OrderNo as it will allow each order record to have its own identity\*/

FOREIGN KEY (CustomerNo) REFERENCES Customer(CustomerNo), /\*Foreign key CustomerNo will create the link between this order table and the customer table\*/

FOREIGN KEY (PaymentNo) REFERENCES Payment(PaymentNo) /\*Foreign key PaymentNo will create link between this table and the payment table\*/

);

Insert Customer Stored Procedure

CREATE PROCEDURE InsertCustomer

AS

BEGIN

INSERT INTO Customer

VALUES ('Mary', 'Johnson', 'maryJohnson@gmail.com', '07939662306', '33 Oakhurst Avenue')

INSERT INTO Customer

VALUES ('Mike', 'Jones', 'mikeJones@icloud.com', '07923364906', '1 Bentwood Road')

INSERT INTO Customer

VALUES ('Janet', 'Loghran', 'janetLoghran@yahoo.co.uk', '02890765465', '33 Hazelwood Park')

INSERT INTO Customer

VALUES ('Bob', 'Smith', 'bobbySmith@belfastmet.ac.uk', '07878665453', '12 Malone Road')

INSERT INTO Customer

VALUES ('Tiger', 'Woods', 'TW@yahoo.co.uk', '07939352306', '39 Jupeter Road')

INSERT INTO Customer

VALUES ('Zach', 'Johnson', 'ZachJohnston@icloud.com', '07865443256', '33 Ivor Park')

INSERT INTO Customer

VALUES ('Brian', 'Mcgee', 'BrianM@BTinternet.com', '02890765848', '103 Brentwood Road')

INSERT INTO Customer

VALUES ('Bobby', 'Jones', 'BobbyJones@gmail.co.uk', '07878776564', '122 Malone Road');

END

GO

EXEC InsertCustomer;

Insert MultiItem Stored Procedure

CREATE PROCEDURE MultiItemInsert

AS

BEGIN

INSERT INTO MultiItem

VALUES ('1','1','23', '31')

INSERT INTO MultiItem

VALUES('3','2','24','32')

INSERT INTO MultiItem

VALUES ('4', '3', '25','33')

INSERT INTO MultiItem

VALUES('2','4','27','34')

INSERT INTO MultiItem

VALUES ('5', '5', '26','35')

INSERT INTO MultiItem

VALUES('6','6','28','36')

INSERT INTO MultiItem

VALUES ('7', '7', '29','37')

INSERT INTO MultiItem

VALUES('8','4','30','38');

END

GO

EXEC MultiItemInsert;

Insert payment stored procedure

CREATE PROCEDURE InsertPayment

AS

BEGIN

INSERT INTO Payment

VALUES('Card', '56', '2017-11-07')

INSERT INTO Payment

VALUES('Cash', '54', '2018-02-12')

INSERT INTO Payment

VALUES('Card', '130', '2016-04-07')

INSERT INTO Payment

VALUES('Card', '150', '2017-07-16')

INSERT INTO Payment

VALUES('Cash', '20', '2019-04-14')

INSERT INTO Payment

VALUES('Card', '75', '2017-12-07')

INSERT INTO Payment

VALUES('Card', '56', '2017-11-07');

INSERT INTO Payment

VALUES('Cash', '85', '2019-11-07');

INSERT INTO Payment

VALUES('Cash', '199', '2019-04-30');

INSERT INTO Payment

VALUES('Card', '130', '2019-04-07');

INSERT INTO Payment

VALUES('Card', '150', '2019-07-16');

INSERT INTO Payment

VALUES('Cash', '20', '2019-06-14');

INSERT INTO Payment

VALUES('Card', '75', '2018-12-07');

INSERT INTO Payment

VALUES('Card', '56', '2018-11-07');

END

GO

EXEC InsertPayment;

Insert product Stored Procedure

CREATE PROCEDURE InsertProduct

AS

BEGIN

INSERT INTO Product

VALUES ('Microsoft Word 18', '56', 'Word Processing', '2018-09-01')

INSERT INTO Product

VALUES ('Powerpoint 2016', '54', 'Presentation Software', '2017-02-22')

INSERT INTO Product

VALUES ('Microsoft Excel 18', '73', 'Spreadsheet Software', '2016-07-13')

INSERT INTO Product

VALUES ('Read/Write', '130', 'Assistive Software', '2016-11-07')

INSERT INTO Product

VALUES ('Mindjet Manager', '150', 'Assistive Software', '2017-06-01')

INSERT INTO Product

VALUES ('Netbeans', '20', 'Development', '2016-03-12')

INSERT INTO Product

VALUES ('SQL Server Management Studio', '55', 'Development', '2019-04-01');

INSERT INTO Product

VALUES ('Excel', '75', 'Spreadsheet Software', '2016-11-03');

INSERT INTO Product

VALUES ('PhotoShop', '199', 'Photo Editing', '2019-02-01');

INSERT INTO Product

VALUES ('Mac OS', '150', 'Operating System', '2018-03-01');

INSERT INTO Product

VALUES ('SQL Server Management Studio 2019', '85', 'Development', '2019-04-30');

END

GO

EXEC InsertProduct;

Insert product order stored procedure

CREATE PROCEDURE InsertItemOrder

AS

BEGIN

INSERT INTO ProductOrder

VALUES ('2017-11-07', '56','1','1')

INSERT INTO ProductOrder

VALUES ('2018-02-12', '54','2','2')

INSERT INTO ProductOrder

VALUES ('2016-04-07', '130','3','5')

INSERT INTO ProductOrder

VALUES ('2018-02-12', '54','2','2')

INSERT INTO ProductOrder

VALUES ('2018-02-12', '54','2','2');

END

GO

Delete Payment Stored Procedure

CREATE PROCEDURE DeletePayment

AS

DELETE FROM Payment WHERE PaymentNo = 1;

GO

Delete product stored procedure

CREATE PROCEDURE DeleteProduct

AS

DELETE FROM Product WHERE ProductName = 'Netbeans';

GO

Modify Item Stored Procedure

CREATE PROCEDURE UpdateProductPrice

AS

UPDATE Product

SET ProductCost = '65'

WHERE ProductName = 'Powerpoint 2016';

GO

EXEC UpdateProductPrice;

Modify Payment Stored Procedure

CREATE PROCEDURE ModifyPayment

AS

UPDATE Payment

SET PaymentType = 'Cash'

WHERE PaymentNo = 1;

GO

EXEC ModifyPayment;

Update Customer Record Stored Procedure

CREATE PROCEDURE UpdateCustomerRecord

AS

UPDATE Customer

SET Email = 'maryJohnson@yahoo.co.uk'

WHERE Forename = 'Mary'

AND Surname = 'Johnson';

GO

EXEC UpdateCustomerRecord;

Update Sales Total Trigger

CREATE TRIGGER CalcOrderTotal

ON ProductOrder

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SELECT SUM(ProductCost) AS [Total Sales Generated]

FROM Product

END;

Create Audit Table Statement

CREATE TABLE OrdersAudit

(

OrderAuditID integer Identity (1,1) primary key,

OrderNo integer,

OrderCost varchar(20),

CustomerNo BIGINT,

UpdatedBy nvarchar(128),

UpdatedOn datetime

);

Audit Table Trigger

CREATE TRIGGER TriggerAuditRecord on ProductOrder

after update, insert

as

begin

insert into OrdersAudit

(OrderNo, OrderCost, CustomerNo, UpdatedBy, UpdatedOn)

select i.OrderNo, i.OrderCost, i.CustomerNo, SUSER\_NAME(), GETDATE()

FROM ProductOrder t

INNER JOIN inserted i on t.OrderNo=i.OrderNo;

END

GO

Test Audit Table

use BMCApps

UPDATE ProductOrder

SET OrderCost = '60'

where OrderNo=31

Go

View Product Sales View

/\*

This command creates a view which shows the total sales generated by each individual product. The View uses an

inner join to join the tables Product and Multi Item

\*/

CREATE VIEW ViewProductSales AS

SELECT Product.ProductNo, Product.ProductName, ((Product.ProductCost)\*(COUNT(MultiItem.ProductNo))) AS [Total Sales]

FROM MultiItem

INNER JOIN Product ON MultiItem.ProductNo = Product.ProductNo

GROUP BY Product.ProductNo, Product.ProductName, Product.ProductCost;

GO

SELECT \* FROM ViewProductSales;

GO

Card Sales View

/\*

This command creates a view which shows the total sales by each individual product whereby the payment used was card.

The command uses numerous inner joins to join the tables needed.

\*/

CREATE VIEW TotalCardSales AS

SELECT Product.ProductNo, Product.ProductName, ((Product.ProductCost)\*(COUNT(MultiItem.ProductNo))) AS [Total Sales]

FROM MultiItem

INNER JOIN Product ON MultiItem.ProductNo = Product.ProductNo

INNER JOIN ProductOrder ON MultiItem.OrderNo = ProductOrder.OrderNo

INNER JOIN Payment ON ProductOrder.PaymentNo = Payment.PaymentNo

GROUP BY Product.ProductNo, Product.ProductName, Product.ProductCost, Payment.PaymentType

HAVING Payment.PaymentType = 'Card';

GO

SELECT \* FROM TotalCardSales;

GO

Cash Sales Create View

/\*

This command creates a view which shows the total sales by each individual product whereby the payment used was cash.

The command uses numerous inner joins to join the tables needed.

\*/

CREATE VIEW TotalCashSales AS

SELECT Product.ProductNo, Product.ProductName, ((Product.ProductCost)\*(COUNT(MultiItem.ProductNo))) AS [Total Sales]

FROM MultiItem

INNER JOIN Product ON MultiItem.ProductNo = Product.ProductNo

INNER JOIN ProductOrder ON MultiItem.OrderNo = ProductOrder.OrderNo

INNER JOIN Payment ON ProductOrder.PaymentNo = Payment.PaymentNo

GROUP BY Product.ProductNo, Product.ProductName, Product.ProductCost, Payment.PaymentType

HAVING Payment.PaymentType = 'Cash';

GO

SELECT \* FROM TotalCardSales;

GO

Sales By City Create View

CREATE VIEW SalesCountEachCountryBefore2019 AS

SELECT Payment.BillingCity, COUNT(MultiItem.ProductNo) AS [Number of Sales]

FROM MultiItem

INNER JOIN Product ON MultiItem.ProductNo = Product.ProductNo

INNER JOIN ProductOrder ON MultiItem.OrderNo = ProductOrder.OrderNo

INNER JOIN Payment ON ProductOrder.PaymentNo = Payment.PaymentNo

WHERE ProductOrder.OrderDate <'2019-01-01'

GROUP BY Payment.BillingCity

GO

SELECT \* FROM SalesCountEachCountryBefore2019

GO

View Customer Orders Create View

CREATE VIEW ViewCustomerOrders AS

SELECT Customer.Forename, Customer.Surname, MultiItem.ProductNo, Product.ProductName

FROM Customer

INNER JOIN MultiItem ON Customer.CustomerNo = MultiItem.CustomerNo

INNER JOIN Product ON MultiItem.ProductNo = Product.ProductNo;