



OMS manages the orders placed by the buyers from initial acceptance to shipment. So that purpose I have created data model for order management system. I have catagoriesed the tables into 7 schemas. The major advantage is to catagories the tables into different schemas are as 1) Schemas provide the opportunity to simplify administration of security, backup and restore, and database management by allowing database objects, or entities, to be logically grouped together. List of schemas that are being used in this database are as given below. 1) common - Specifies the commonly (MASTER tables) used tables the OMS. 2) byr - Indicates buyer related entities. 3) splr - Indicates suplier related tables. 4) ord - Grouped order related tables in the database. 5) prdt - Specifies product related entities in the database. 6) trn - Indicates transacations related tables. 7) applog - Indicates applog related tables. (To capture errors raised by the stored procedures.) List of tables are as given below. 1) Country - Stores countries info in the table. 2) State - Persist country related states info. 3) District - Stores state related districts info. 4) City - Stores district related cities info. 5) Currency - Stores country specific currencies in the table. 6) AddressType - To store the different type of addresses. (Permenent AND Temparory) 7) OrderStatus - To persist different type of order status in the system. 8) Buyer - To store buyer specific info 9) BuyerAddresses - Buyer can have multiple addresses (Permenent AND Temparory). So that purpose I have created separate table to persist multiple addresses of the buyer. 10) Supplier - To store supplier/vendor specific info. (Supplier provides products) 11) ProductCategory - Different types of the products will be present in the system. So I have created product category table in the database. 12) ProductImage - To store multiple images of the product. 13) Order - Stores order details of the specific buyer. 14) OrderProduct - Order can have multiple products. So that we can keep track of individual product with its price. 15) Payment - Payment made against specific order info. 16) CRUDErrors - To capture errors raised by the stored procedures.

**Key Points of The Database**: 1) Defiened proper naming conventions for all the database objects to easy to understand. 2) Categorized the tables into different schemas to maintain schema based roles and backups. 3) Mantained referntial intigrities across all the database. 4) Used proper datatypes as per the specs. 5) Applied proper indexes on the table. 5) Defined various types of constraints on the table. 6) Maintained normalization in the database.

**Stored Procedure**

**Name :** ord.crud\_GenerateOrders **Parameters : 1)** @tblOrderTableType **2)** @tblOrderProductTableType (Table valued input parameters)

**Sample syntax for SP Execution :**

-- Declaring the variable of user defined table type

DECLARE @tblTypeOrder ord.tblTypeOrder

DECLARE @tblTypeOrderProduct prdt.tblTypeOrderProduct

--Inserting bulk orders

INSERT INTO @tblTypeOrder

(OrderID,BuyerID,OrderDateTime,TotalAmount,TotalDiscountAmount,TotalNetAmount,CurrencyID,OrderStatusId,ModifiedOn)

VALUES

(0,1,'2020-11-05 10:33:26.793',100,10,90,1,1,'2020-11-05 10:33:26.793')

,(3,2,'2020-11-06 10:33:26.793',500,10,490,1,1,'2020-11-06 10:33:26.793')

,(0,1,'2020-11-07 10:33:26.793',11100,10,11090,1,1,'2020-11-07 10:33:26.793')

,(0,1,'2020-11-07 10:33:26.793',1500,10,1490,1,1,'2020-11-07 10:33:26.793')

,(5,3,'2020-11-08 10:33:26.793',1100,10,1090,1,1,'2020-11-08 10:33:26.793')

,(0,1,'2020-11-08 10:33:26.793',200,10,110,1,1,'2020-11-08 10:33:26.793')

,(6,4,'2020-11-09 10:33:26.793',10,10,0,1,1,'2020-11-09 10:33:26.793')

,(7,5,'2020-11-09 10:33:26.793',50,10,40,1,1,'2020-11-09 10:33:26.793')

,(0,7,'2020-11-09 10:33:26.793',1020,10,1010,1,1,'2020-11-09 10:33:26.793')

,(9,9,'2020-11-09 10:33:26.793',1000,10,900,1,1,'2020-11-09 10:33:26.793')

--Inserting order items

INSERT INTO @tblTypeOrderProduct

(OrderProductID,OrderID,ProductID,Qty,Amount,DiscountAmount,NetAmount,ModifiedOn)

VALUES

(1,1,1,100,10,2,8,NULL),(0,2,3,200,10,190,8,NULL),(3,3,4,400,10,390,8,NULL)

,(4,4,4,500,10,490,8,NULL),(5,5,5,600,10,590,8,NULL),(2,6,6,100,10,90,8,NULL)

-- Executing procedure

EXEC ord.crud\_GenerateOrders @tblTypeOrder,@tblTypeOrderProduct

**Trigger**

**Name :** ord.trg\_OrderAuditRecords **On Table Name** : ord.[Order]

**Name :** prdt.trg\_OrderProductAuditRecords **On Table Name** : prdt.[OrderProduct]