# MSIS 2603 - Database Management Systems

**T-Shirt Memories** ☺

An overview of the database

By: Lasya Pathuri

**TABLE OF CONTENTS**

## BUSINESS DESCRIPTION

***T-shirt Memories*** is a Textile business that turns your memorable old T-shirts in to new products. People have many memorable T-shirts piled up from school, college, Office logo shirts, few won as a gift and many more. We don't like to throw them; however, they occupy huge space in our wardrobe. ***T-shirt Memories*** will collect all the piled-up T-shirts and make it into a new useful product. It collects old T-shirts from its customers and turns them into a product of customers choice. The choices include a new T-shirt which is a collage of all, Pillow or a Shopping bag. The customer can choose the color and size for the new product. The sizes can be either small, medium or large. The customers will be charged according to the size and the product they choose.

The products are manufactured in-house. There are three different types of employees at the company – Manager, Tailors and Machine operators. The shirts collected from customer would be picked up by machine operator and will go into a cutting machine. It would take a perfect square from each of the T-shirts. Customer can put a mark on the shirt which part he wants to be cut for new product while sending. Tailor would sew all the squares cut together on the front of the product. On the back a high-quality fleece is used which will be in color of customers choice. The fleece is purchased from fabric stores based on yards. Managers will be assigned to supervise the order and machine operators. Also, the manager does a quality check of the product before dispatching it to the customer.

## DATABASE DESCRIPTION

The database for T-Shirt Memories maintains records of the customers, products ordered, Machines utilized to fulfill an order, Tailors and machine operators who worked on orders, Managers who supervise the order and approves the order, suppliers from which the company acquires raw materials. The database provides multiple user views to ease in data sharing and prevent redundancy of data. The users are the company management, managers and the delivery personnel/agency. Managers can see the order details and the rating given to the order by the customer. Delivery personnel/agency will be provided with details of customers to deliver the packages. Finance Management has a view to track the orders failed due to quality check and loss incurred due to re-work on that failed order. As T-Shirt Memories main business is to sell products online and has to deal with many customers on day to day, the database would be very useful for information sharing between Customers and T-Shirt Materials.

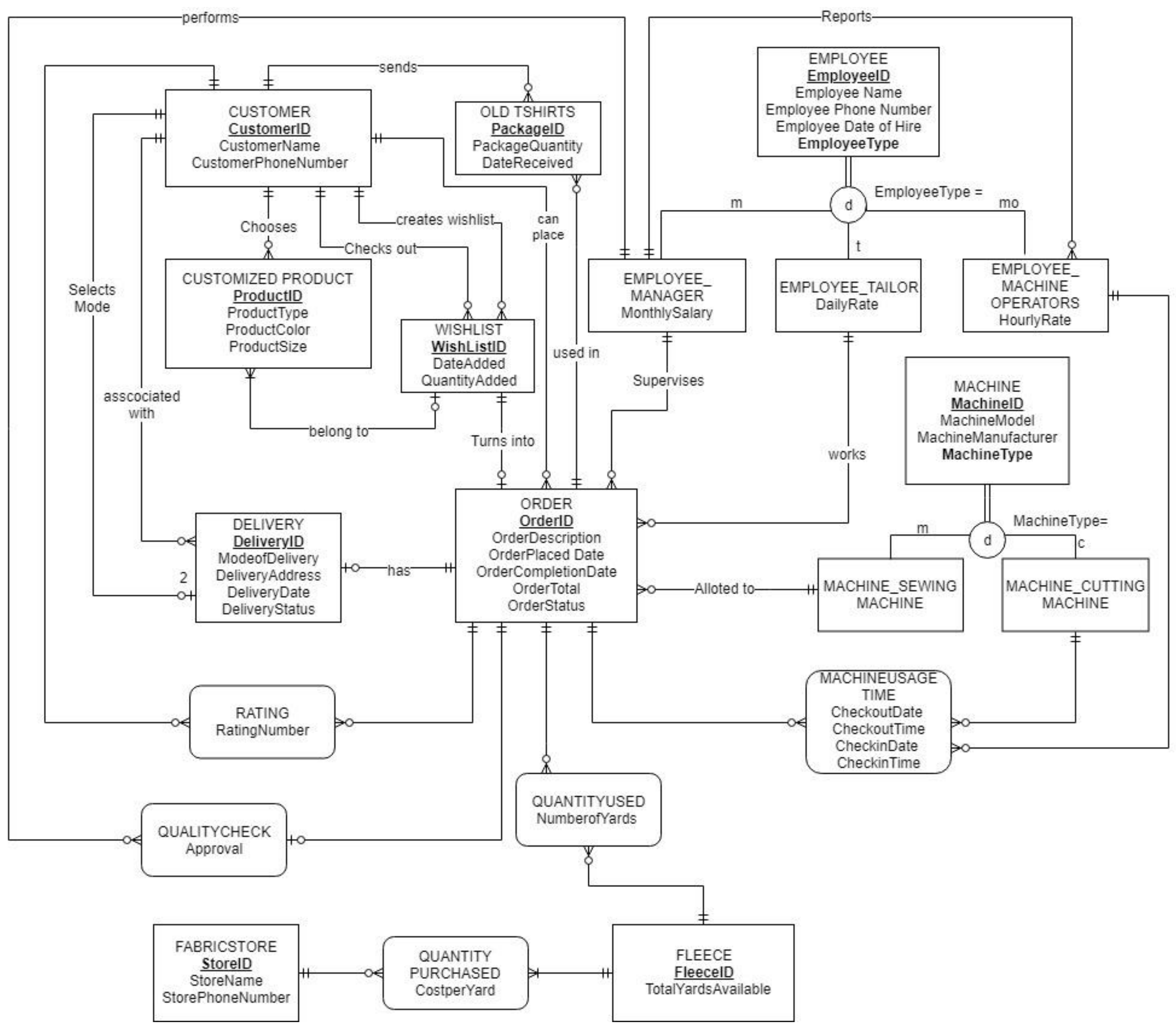
## VALUE OF DATABASE

T-Shirt memories has large amount of transactions, data inflow and outflow happening every day. There is no proper system to track huge amount of transactions happening. To maintain an effective flow of data and track all the transactions, company needs a database. It helps in preventing loss of data, avoid data redundancy, maintain consistency among all units of the company and also enforces proper standards to be followed. As T-Shirt memories sell the products to a number of customers and also purchases raw material from suppliers, it is important to have a database to track all the activities related to manufacturing products, supervising employee and Orders placed. The database also records the revenue generated by the company and loss incurred. The database allows to create user specific views to secure data from unauthorized users. It increases the productivity of the company. The database also helps the company to view the frequent customers and attract them by providing special offers. It helps the company to improve its customer base. It also helps the company to track performance of tailors and support in decision making on selecting efficient tailors and their compensation.

## BUSINESS RULES

1. Customer can create product of their own choice, depending on the options available, every product customer chooses will generate a unique productid.
2. Customer registered with T-Shirt memories can create many Wishlist by adding one or many products to one or many Wishlist. A customer can exist in the database even without creating a Wishlist.
3. Manager, Tailor and sewing Machine are assigned simultaneously when an order is placed and stored in the system.
4. Order can exist in the database without Old Tshirts- packageId and DeliveryID. Once the order is placed, customer send his shirts to T-Shirt memories and then the table is updated with package details.
5. Once the order is placed customer has two options to choose for mode of delivery(courier,pickup). After customer selects the mode and fills in delivery address deliveryid is generated and associated with order and customer.
6. A customer can exist in the data base without orderid, productid ,Old Tshirts – packageid ,wishlistid and deliveryid.
7. Employee Machine Operators check out Cutting Machines, the time and date of check-in and check-out are recorded.
8. Manager does quality check and approves to dispatch or not for an order after it is completed, if an order does not meet the requirements it needs to undergo re-work and again go through quality check.
9. Every Fleece purchased from fabric store will be stored with a unique Fleece ID and yards available on hand.
10. The quantity used in the order is recorded with unique orderID and FleeceID. Order can exist in the database before determining quantity used. Quantity used in the order can be updated later.

# CONCEPTUAL SCHEMA – ER MODEL



# LOGICAL SCHEMA – RELATIONAL TABLE DESIGN

| | | | | |
|---|---|---|---|---|
| **CUSTOMER** | **CustomerID** | CustomerName | CustomerPhoneNumber | |
| **CUSTOMIZED PRODUCT** | **ProductID** | ***CustomerID*** | ProductType | ProductColor |
| | ProductSize | | | |
| **WISHLIST** | **WishListID** | ***ProductID*** | ***CustomerWishListID*** | ***CustomerCheckoutID*** |
| | Quantity Added | DateAdded | | |
| **ORDER** | **OrderID** | ***CustomerID*** | ***WishListID*** | ***tEmployeeID*** |
| | ***sMachine ID*** | ***mEmployeeID*** | OrderDescription | OrderPlacedDate |
| | OrderCompletionDate | OrderStatus | | |
| **OLD TSHIRTS** | **PackageID** | ***CustomerID*** | ***OrderID*** | Package Quantity |
| | Date Rececived | | | |
| **EMPLOYEE** | **EmployeeID** | EmployeeName | EmployeePhoneNumber | EmployeeDateofHire |
| | **EmployeeType** | | | |
| **EMPLOYEE_MANAGER** | ***mEmployeeID*** | MonthlySalary | | |
| **EMPLOYEE_TAILOR** | ***tEmployeeID*** | DailyRate | | |
| **EMPLOYEE_MACHINEOPERATOR** | ***moEmployeeID*** | ***mEmployeeID*** | HourlyRate | |
| **MACHINE** | **MachineID** | MachineModel | MachineManufacturer | **MachineType** |
| **MACHINE_SEWING MACHINE** | ***sMachineID*** | | | |
| **MACHINE_CUTTING MACHINE** | ***cMachineID*** | | | |
| **FABRICSTORE** | **StoreID** | StoreName | StorePhone Number | |
| **QUANTITY PURCHASED** | ***StoreID*** | ***FleeceID*** | CostperYard | |
| **FLEECE** | **FleeceID** | TotalYardsAvailable | | |
| **MACHINEUSAGETIME** | **MachineUtilizedID** | ***moEmployeeID*** | ***cMachine ID*** | ***OrderID*** |
| | CheckoutDate | CheckoutDate and Time | CheckinDate | CheckinTime |
| **DELIVERY** | **DeliveryID** | ***Customerselection ID*** | ***CustomerDelivery ID*** | ***OrderID*** |
| | Mode of Delivery | DeliveryAddress | DeliveryDate | DeliveryStatus |
| **QUANTITY USED** | ***OrderID*** | ***FleeceID*** | NumberofYards | |
| **QUALITY CHECK** | ***mEmployeeID*** | ***OrderID*** | Approval | |
| **RATING** | ***CustomerID*** | ***OrderID*** | RatingNumber | |

# DATA DICTIONARY

**CUSTOMER**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| CustomerID | bigint | >0 | PK | Unique identifier of customer | 123450 |
| CustomerName | nvarchar(100) | | | Name of the customer | James Smith |
| CustomerPhoneNumber | char(10) | >0 | | Mobile phone number of customer | 6507097187 |

**CUSTOMIZEDPRODUCT**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| ProductID | bigint | >0 | PK | Unique indentifier for product | 551232 |
| ProductType | varchar(15) | ('Tshirt','pillow','shopping bag') | | Type of the product | Tshirt |
| ProductColor | varchar(10) | | | Color of the product | Purple |
| ProductSize | varchar(8) | ('Small','Medium','Large') | | Size of the product | standard |
| CustomerID | bigint | >0 | FK | Unique identifier of customer | 123450 |

**WISHLIST**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| WishListID | bigint | >0 | PK | Unique indentifier for wishlist | 12351 |
| DateAdded | date | | | Date when product is added to the wishlist | 8/16/2018 |
| QuantityAdded | int | >0 | | Quanitity of products contained | 3 |
| CustomerWishListID | bigint | >0 | FK | Unique identifier of customer | 123450 |
| CustomerCheckoutID | bigint | >0 | FK | Unique identifier of customer | 123450 |
| ProductID | bigint | >0 | PK | Unique indentifier for product | 551232 |

**OLD T-SHIRTS**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| PackageID | bigint | >0 | PK | Unique identifier for the T-shirt Package | 44123 |
| PackageQuantity | int | >0 | | Count of T-shirts in package | 4 |
| DateReceived | date | | | Date when package is received to T-shirt Memories | 8/16/2018 |
| CustomerID | bigint | >0 | FK | Unique identifier of customer | 123450 |
| OrderID | bigint | >0 | FK | Unique identifier for the order | 67823 |

**ORDER**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| OrderID | bigint | >0 | PK | Unique identifier for the order | 67823 |
| OrderDescription | nvarchar(500) | | | Decription of the order | Pillow of size standard and color purple |
| Orderplaced Date | date | | | Date when the order is placed | 8/16/2018 |
| OrderCompletion Date | date | | | Date when the order is ready for delivery | 12/21/2018 |
| OrderTotal | decimal(6.2) | >0.00 | | Total Amount of order to be paid by customer | 123.20 |
| OrderStatus | nvarchar(10) | | | Status of the order | Inprogress |
| CustomerID | bigint | >0 | FK | Unique identifier of customer | 123450 |
| WishListID | bigint | >0 | FK | Unique indentifier for the WishList | 12351 |
| tEmployeeID | bigint | >0 | FK | Unique identifier for Tailors | 987356 |
| sMachineID | bigint | >0 | FK | Unique identifier for Sewing Machines | 34530 |
| mEmployeeID | bigint | >0 | FK | Unique identifier for managers | 987414 |

| EMPLOYEE | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| EmployeeID | bigint | >0 | PK | Unique identifier of an employee | 987289 |
| EmployeeName | nvarchar(50) | | | Name of an Employee | Fransesca |
| EmployeePhoneNumber | char(10) | >0 | | Mobile phone number of Employee | 6507097187 |
| EmployeeDateofHire | date | | | Date when employee is hired by the company | 8/16/2018 |
| EmployeeType | varchar(2) | ('m','t','mo') | | Discriminator for an employee type, Manager(M), Tailor(T), Machine | mo |

| EMPLOYEE_MANAGER | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| mEmployeeID | bigint | >0 | PK,FK | Unique identifier for managers | 987414 |
| Monthly Salary | decimal(6,2) | >0.00 | | Monthly salary for Manager | 1200.00 |

| EMPLOEE_TAILOR | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| tEmployeeID | bigint | >0 | PK,FK | Unique identifier for Tailors | 987356 |
| Daily Rate | decimal(6,2) | >0.00 | | Daily rate for Tailor | 50.00 |

| EMPLOYEE_MACHINEOPERATOR | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| moEmployeeID | bigint | >0 | PK,FK | Unique identifier for Machine Operators | 12345 |
| Hourly Rate | decimal(6,2) | >0.00 | | Hourly rate for Machine Operator | 30.00 |
| mEmployeeID | bigint | >0 | FK | Unique identifier for managers | 12345 |

| MACHINE | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| MachineID | bigint | >0 | PK | Unique identifier for Machines | 34524 |
| MachineModel | int | >0 | | Model Number of the machine | 45678 |
| MachineManufacturer | nvarchar(10) | | | Manufacturer name of the Machine | Boss HP |
| MachineType | varchar(1) | ('c','s') | | Discriminator for machines, sewing machines(s), cutting machines(c) | c |

| MACHINE_SEWING MACHINE | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| sMachineID | bigint | >0 | PK,FK | Unique identifier for Sewing Machines | 34527 |

| MACHINE_CUTTING MACHINE | | | | | |
|---|---|---|---|---|---|
| **Name** | **Data Type** | **Constraints** | **Key** | **Description** | **Example Value** |
| cMachineID | bigint | >0 | PK,FK | Unique identifier for Cutting Machines | 34524 |

**DELIVERY**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| DeliveryID | bigint | >0 | PK | Unique identifier for each delivery | 11103 |
| ModeofDelivery | nvarchar(7) | ('pickup','dropoff') | | Delivery Mode | pickup |
| DeliveryAddress | nvarchar(100) | | | Address to be used for deliverery/pickup | 310, Maclane st, PaloaAlto,94396 |
| DeliveryDate | date | | | Date which product will be delivered | 8/16/2018 |
| DeliveryStatus | nvarchar(10) | | | Status of delivery | shipped |
| CustomerSelectionID | bigint | >0 | FK | Unique identifier for each Customer selection | 123450 |
| CustomerDeliveryID | bigint | >0 | FK | Unique identifier for each Customer Delivery | 123450 |
| OrderID | bigint | >0 | FK | Unique identifier for the order | 67823 |

**QUANTITY USED**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| OrderID | bigint | >0 | PK,FK | Unique identifier for the order | 67823 |
| FleeceID | bigint | >0 | PK,FK | Unique indetifier for each fleece purchased | 80945 |
| NumberofYards | int | >0 | | Number of Yards of Fleece used in order | 23 |

**MACHINEUSAGETIME**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| MachineUtilizedID | bigint | >0 | PF | Unique identifier for Machine Utilized | 22105 |
| CheckoutDate | date | | | Date when cutting machine is checked out | 8/16/2018 |
| CheckoutTime | time | | | Time when cutting machine is checked out | 3:15PM |
| CheckinDate | date | | | Date when cutting machine is checked in | 8/16/2018 |
| CheckinTime | time | | | Time when cutting machine is checked out | 4:15 PM |
| moEmployeeID | bigint | >0 | FK | Unique identifier for Machine Operators | 987419 |
| cMachineID | bigint | >0 | FK | Unique identifier for Cutting Machines | 34525 |
| OrderID | bigint | >0 | FK | Unique identifier for the order | 67823 |

**FABRICSTORE**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| StoreID | bigint | >0 | PF | Unique identifier for store | 50009 |
| StoreName | nvarchar(100) | | | Name of the store | HP |
| StorePhoneNumber | char(10) | >0 | | Mobile Phone Number of the store | 6507097187 |

**QUANTITY PURCHASED**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| CostperYard | decimal(4,2) | >0 | | Cost per each yard of fleece | 23.00 |
| StoreID | bigint | >0 | PK,FK | Unique identifier for store | 50009 |
| FleeceID | bigint | >0 | PK,FK | Unique indetifier for each fleece purchased | 80945 |

**FLEECE**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| FleeceID | bigint | >0 | PK | Unique indetifier for each fleece purchased | 80945 |
| TotalYardsAvailable | int | | | Number of Yards of Fleece Available | 80 |

**QUALITY CHECK**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| Approval | varchar(2) | ('ok','ko') | | Appoval to dispatch or not | ok |
| mEmployeeID | bigint | >0 | PK,FK | Unique identifier for managers, Checks the quality of items in each o | 123450 |
| OrderID | bigint | >0 | PK,FK | Unique identifier for the order | 67823 |

**RATING**

| Name | Data Type | Constraints | Key | Description | Example Value |
|---|---|---|---|---|---|
| RatingNumber | decimal(2,1) | >=0 and <=5 | | Raving given by customer to a particular order | 4.2 |
| CustomerID | bigint | >0 | PK,FK | Unique identifier of customer | 123450 |
| OrderID | bigint | >0 | PK,FK | Unique identifier for the order | 67823 |

## SQL STATEMENTS

### 1. Customer_T

```
CREATE TABLE Customer_T
(CustomerID int not null CHECK(CustomerID>0),
CustomerName nvarchar(100) not null,
CustomerPhoneNumber char (10),
CONSTRAINT Customer_PK PRIMARY KEY(CustomerID))
```

### 2. CustomizedProduct_T

```
CREATE TABLE CustomizedProduct_T
(ProductID int not null CHECK(ProductID>0),
ProductType varchar(15) CHECK(ProductType IN ('Tshirt','Pillow','Shopping bag')) not null,
ProductColor varchar(10) not null,
ProductSize varchar(8) CHECK( ProductSize IN('Small','Medium','Large')) not null,
CustomerID int not null CHECK(CustomerID>0)
CONSTRAINT CustomizedProduct_PK PRIMARY KEY(ProductID),
CONSTRAINT Customer_FK FOREIGN KEY(CustomerID)
REFERENCES Customer_T(CustomerID))
```

### 3. WishList_T

```
CREATE TABLE WishList_T
(WishListID int not null CHECK(WishListID>0),
ProductID int not null CHECK(ProductID>0),
DateAdded date default GETDATE(),
QuantityAdded int not null CHECK(QuantityAdded>0),
CustomerWishListID int not null CHECK(CustomerWishListID>0),
CustomercheckoutID int not null CHECK(CustomercheckoutID>0)
CONSTRAINT Cart_PK PRIMARY KEY(WishListID),
CONSTRAINT Customer_FK1 FOREIGN KEY(CustomerWishListID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Customer_FK2 FOREIGN KEY(CustomercheckoutID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Product_FK FOREIGN KEY(ProductID) REFERENCES CustomizedProduct_T(ProductID))
```

### 4. OldTshirts_T

```
CREATE TABLE OldTshirts_T
(PackageID int not null CHECK(PackageID>0),
PackageQuantity int not null CHECK(PackageQuantity>0),
DateReceived date default GETDATE(),
CustomerID int not null CHECK(CustomerID>0),
OrderID int not null CHECK(OrderID>0),
CONSTRAINT OldTshirts_PK PRIMARY KEY(PackageID),
CONSTRAINT Customer_FK6 FOREIGN KEY(CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Order_FK FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID))
```

**5. Order_T**

```
CREATE TABLE Order_T
(OrderID int not null CHECK(OrderID>0),
OrderDescription nvarchar(500),
OrderPlacedDate date default GETDATE(),
OrderCompletiondate date,
OrderTotal decimal(6,2) not null check(OrderTotal>0),
OrderStatus nvarchar(10),
CustomerID int not null CHECK(CustomerID>0),
WishListID int not null CHECK(WishListID>0),
tEmployeeID int not null CHECK(tEmployeeID>0),
sMachineID int not null CHECK(sMachineID>0),
mEmployeeID int not null CHECK(mEmployeeID>0)
CONSTRAINT Order_PK PRIMARY KEY(OrderID),
CONSTRAINT Customer_FK3 FOREIGN KEY(CustomerID)
REFERENCES Customer_T(CustomerID),
CONSTRAINT WishList_FK FOREIGN KEY(WishListID)
REFERENCES WishList_T(WishListID),
CONSTRAINT Employee_Tailor_FK1 FOREIGN KEY(tEmployeeID)
REFERENCES Employee_Tailor_T(tEmployeeID),
CONSTRAINT Employee_SewingMachine_FK FOREIGN KEY(sMachineID)
REFERENCES Machine_SewingMachine_T(sMachineID),
CONSTRAINT Employee_Manager_FK2 FOREIGN KEY(mEmployeeID)
REFERENCES Employee_Manager_T(mEmployeeID))
```

**6. Employee_T**

```
CREATE TABLE Employee_T
(EmployeeID int not null CHECK(EmployeeID>0),
EmployeeName nvarchar(50) not null,
EmployeePhoneNumber char(10),
EmployeeDateofHire date default GETDATE(),
EmployeeType varchar(2) not null CHECK(EmployeeType IN('m','t','mo'))
CONSTRAINT Employee_PK PRIMARY KEY(EmployeeID))
```

**7. Employee_Manager_T**

```
(mEmployeeID int not null CHECK(mEmployeeID>0),
Monthlysalary decimal (6,2) not null CHECK(Monthlysalary>0.0),
CONSTRAINT Employee_Manager_PK PRIMARY KEY(mEmployeeID),
CONSTRAINT Employee_Manager_FK FOREIGN KEY(mEmployeeID)
REFERENCES Employee_T(EmployeeID))
```

**8. Employee_Tailor_T**

```
CREATE TABLE Employee_Tailor_T
```

```
(tEmployeeID int not null CHECK(tEmployeeID>0),
DailyRate decimal(6,2) not null CHECK(DailyRate>0),
CONSTRAINT Employee_Tailor_PK PRIMARY KEY(tEmployeeID),
CONSTRAINT Employee_Tailor_FK FOREIGN KEY(tEmployeeID)
REFERENCES Employee_T(EmployeeID))
```

**9. Employee_MachineOperator_T**

```
CREATE TABLE Employee_MachineOperator_T
(moEmployeeID int not null CHECK(moEmployeeID>0),
HourlyRate decimal(6,2) not null CHECK(HourlyRate>0),
mEmployeeID int not null CHECK(mEmployeeID>0)
CONSTRAINT Employee_MachineOperator_PK PRIMARY KEY(moEmployeeID)
CONSTRAINT Employee_MachineOperator_FK FOREIGN KEY(moEmployeeID)
REFERENCES Employee_T(EmployeeID),
CONSTRAINT Employee_Manager_FK1 FOREIGN KEY(mEmployeeID)
REFERENCES Employee_Manager_T(mEmployeeID))
```

**10. Machine_T**

```
CREATE TABLE Machine_T
(MachineID int not null CHECK(MachineID>0),
MachineModel int not null,
MachineManufacturer nvarchar(10),
MachineType varchar(1)  not null CHECK(MachineType IN('c','s'))
CONSTRAINT Machine_PK PRIMARY KEY(MachineID))
```

**11. Machine_SewingMachine_T**

```
CREATE TABLE Machine_SewingMachine_T
(sMachineID int not null CHECK(sMachineID>0),
CONSTRAINT Machine_SewingMachine_PK PRIMARY KEY(sMachineID),
CONSTRAINT Machine_SewingMachine_FK FOREIGN KEY(sMachineID)
REFERENCES Machine_T(MachineID))
```

**12. Machine_CuttingMachine_T**

```
CREATE TABLE Machine_CuttingMachine_T
(cMachineID int not null CHECK(cMachineID>0),
CONSTRAINT Machine_CuttingMachine_PK PRIMARY KEY(cMachineID),
CONSTRAINT Machine_CuttingMachine_FK FOREIGN KEY(cMachineID)
REFERENCES Machine_T(MachineID))
```

**13. Delivery_T**

```
CREATE TABLE Delivery_T
(DeliveryID int CHECK(DeliveryID>0),
ModeofDelivery nvarchar(7) not null CHECK(ModeofDelivery IN('pickup','courier')),
DeliveryAddress nvarchar(100),
Deliverydate date,
DeliveryStatus nvarchar(10),
CustomerSelectionID int not null CHECK(CustomerSelectionID>0),
CustomerDeliveryID int not null CHECK(CustomerDeliveryID>0),
OrderID int not null CHECK(OrderID>0),
CONSTRAINT Delivery_PK PRIMARY KEY(DeliveryID),
CONSTRAINT Customer_FK4 FOREIGN KEY(CustomerSelectionID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Customer_FK5 FOREIGN KEY(CustomerDeliveryID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Order_FK FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID))
```

**14. QuantityUsed_T**

```
CREATE TABLE QuantityUsed_T
(NumberofYards int not null CHECK(NumberofYards > 0),
OrderID int not null CHECK (OrderID>0),
FleeceID int not null CHECK(FleeceID>0)
CONSTRAINT QuantityUsed_PK PRIMARY KEY(OrderID,FleeceID),
CONSTRAINT QuantityUsed _FK FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT QuantityUsed _FK1 FOREIGN KEY(FleeceID) REFERENCES Fleece_T(FleeceID))
```

**15. MachineUsageTime_T**

```
CREATE TABLE MachineUsageTime_T
(MachineUtilizedID int not null CHECK(MachineUtilizedID>0),
CheckoutDate date,
CheckoutTime time,
CheckinDate date,
CheckinTime time,
moEmployeeID int not null CHECK(moEmployeeID>0),
cMachineID int not null CHECK(cMachineID>0),
OrderID int not null CHECK (OrderID>0)
CONSTRAINT UsageTime_PK PRIMARY KEY(MachineUtilizedID),
CONSTRAINT Employee_MachineOperator_FK1 FOREIGN KEY(moEmployeeID) REFERENCES
Employee_MachineOperator_T(moEmployeeID),
CONSTRAINT Machine_CuttingMachine_FK1 FOREIGN KEY(cMachineID) REFERENCES
Machine_CuttingMachine_T(cMachineID),
CONSTRAINT Order_FK1 FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID))
```

**16. FabricStore_T**

```
CREATE TABLE FabricStore_T
(StoreID int not null CHECK(StoreID>0),
StoreName nvarchar(100) not null,
StorePhoneNumber char(10),
CONSTRAINT FabricStore_PK PRIMARY KEY(StoreID))
```

**17. QuantityPurchased_T**

```
CREATE TABLE QuantityPurchased_T
(CostperYard decimal(4,2) not null CHECK(CostperYard>0.0),
StoreID int not null CHECK(StoreID>0),
FleeceID int not null CHECK(FleeceID>0),
CONSTRAINT QuantityPurchased_PK PRIMARY KEY(StoreID,FleeceID),
CONSTRAINT QuantityPurchased _FK FOREIGN KEY(StoreID) REFERENCES FabricStore_T(StoreID),
CONSTRAINT QuantityPurchased _FK1 FOREIGN KEY(FleeceID) REFERENCES Fleece_T(FleeceID))
```

**18. Fleece_T**

```
CREATE TABLE Fleece_T
(FleeceID int not null CHECK(FleeceID>0),
TotalYardsAvailable int not null CHECK(TotalYardsAvailable>0)
CONSTRAINT Fleece_PK PRIMARY KEY(FleeceID))
```

**19. QualityCheck_T**

```
CREATE TABLE QualityCheck_T
(Approval varchar(2) not null CHECK(Approval IN('ok','ko')),
OrderID int not null CHECK (OrderID>0),
mEmployeeID int not null CHECK(mEmployeeID>0)
CONSTRAINT QualityCheck_PK PRIMARY KEY(OrderID,mEmployeeID),
CONSTRAINT QualityCheck_FK FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT QualityCheck_FK1 FOREIGN KEY(mEmployeeID) REFERENCES
Employee_Manager_T(mEmployeeID))
```

**20. Rating_T**

```
CREATE TABLE Rating_T
(RatingNumber decimal(2,1) not null CHECK(RatingNumber>=0 and RatingNumber <=5),
CustomerID int not null CHECK(CustomerID>0),
OrderID int not null CHECK(OrderID>0),
CONSTRAINT Rating_PK PRIMARY KEY(CustomerID,OrderID),
CONSTRAINT Rating_FK FOREIGN KEY(CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Rating_FK1 FOREIGN KEY(OrderID) REFERENCES Order_T(OrderID))
```

## MATERIALIZED VIEWS AND PROCEDURES

**1. Manager:**

Managers at T-shirt Memories have customized views for order details and Rating details.

**VIEW 1: Order Details**

The Order details view would help T-shirt Memories in the following ways:

- View Cost of Production for each order.
- Record the Profits of T-Shirt Memories for each order which are completed and got 'ok' for quality check.
- Analyze the number of T-shirts sent by customer for each order.
- Can know the customers who placed more than one order, generate insights about their orders and offer discounts in future.

```
CREATE TABLE OrderDetails_view
(CustomerID int not null,
OrderID int not null,
OrderTotal decimal(5,2),
CostOfProduction decimal(5,2),
Profit decimal(5,2),
PackageQuantity int not null)


CREATE PROCEDURE RefreshOrderDetails_view as
delete from OrderDetails_view
insert into OrderDetails_view
select customer_t.CustomerID,Order_T.OrderID,order_t.ordertotal,
cost_of_production_table.costofproduction,
(order_t.ordertotal - cost_of_production_table.costofproduction) as profit,
OldTshirts_T.PackageQuantity
from order_t, OldTshirts_T,customer_t,qualitycheck_t,
(select order_t.OrderID,
(QuantityUsed_T.NumberofYards * QuantityPurchased_T.CostperYard) as costofproduction
from QuantityUsed_T,QuantityPurchased_T,Fleece_T,Order_T
where
QuantityPurchased_T.FleeceID=Fleece_T.FleeceID and
Order_T.OrderID = QuantityUsed_T.OrderID and
QuantityUsed_T.FleeceID=Fleece_T.FleeceID ) as cost_of_production_table,

(select QualityCheck_T.Approval,Order_T.OrderID from
QualityCheck_T,Order_T
where Order_T.OrderID = QualityCheck_T.OrderID and
QualityCheck_T.Approval = 'ok') as orderquality_table

where Order_T.OrderID=cost_of_production_table.OrderID and
Order_T.OrderID=OldTshirts_T.OrderID and
Customer_T.CustomerID=Order_T.CustomerID and
Order_T.OrderID = orderquality_table.OrderID
group by customer_t.CustomerID,Order_T.OrderID,
order_t.ordertotal,cost_of_production_table.costofproduction,OldTshirts_T.PackageQuantity
```

**VIEW 2: Rating Details**

The Rating details view would help T-Shirt Memories in the following ways:

- To calculate the average, maximum and minimum rating received for an order worked by a Tailor. As one order can be completed by only one tailor, rating given to the Order will be considered as rating given to the work done by a particular tailor.
- T-Shirt Memories could use this rating to categorize tailors as beginners or experts and determine their hourly wage accordingly. T-Shirt Memories can also remove a tailor if he/she gets very low rating.

```
create table RatingDetails_View
(EmployeeTailorID int not null,
EmployeeTailorName nvarchar(50),
AverageRating decimal(2,1),
MaximumRating decimal(2,1),
MinimumRating decimal(2,1))

CREATE PROCEDURE RefreshRatingDetails_view as
delete from RatingDetails_view
insert into RatingDetails_view

select Employee_Tailor_T.tEmployeeID,Employee_T.EmployeeName,
cast(avg(Rating_T.RatingNumber) as decimal(2,1)) as AverageRating,
max(Rating_T.RatingNumber) as MaximumRating, min(Rating_T.RatingNumber) as MinimumRating
from Rating_T,Order_T,Employee_Tailor_T,Employee_T
where Rating_T.OrderID = Order_T.OrderID and
Employee_Tailor_T.tEmployeeID = Order_T.tEmployeeID and
Employee_Tailor_T.tEmployeeID = Employee_T.EmployeeID
group by Employee_Tailor_T.tEmployeeID, Employee_T.EmployeeName
```

**VIEW 3: Delivery Details - Delivery Personnel/Agency**

T-shirt Memories has customized view to provide delivery personnel/delivery agency details of delivery. The delivery details view gives an overview of customer name, customer address, deliveryID,Delivery date, orderId, Total amount of order, Quantity ordered. The main reason why T-Shirt memories have separate view for delivery details is because the mode of delivery can be two types, courier and pickup. The customer has an option to choose the mode of delivery after placing an order. T-Shirt Memories will forward the delivery details of customer who will choose mode of delivery as courier, to a third-party personnel/ delivery agency to deliver the package.

The delivery details view would help T-Shirt Memories in the following ways:

- Ensure order is delivered by the delivery date with-out missing.
- Ensure successful delivery of an order.
- Prevent errors in delivery of packages to customers.

```
CREATE TABLE DeliveryDetails_view
(
CustomerName nvarchar(100) not null,
CustomerPhoneNumber char(10),
DeliveryID int not null,
DeliveryDate date,
DeliveryAddress nvarchar(50),
OrderID int not null,
OrderTotal decimal(5,2),
OrderQuantity int not null)

CREATE PROCEDURE RefreshDeliveryDetails_view as
delete from DeliveryDetails_view
insert into DeliveryDetails_view
```

```
select Customer_T.CustomerName,Customer_T.CustomerPhoneNumber,
Delivery_T.DeliveryID,Delivery_T.Deliverydate,Delivery_T.DeliveryAddress,
Order_T.OrderID,Order_T.ordertotal, WishList_T.QuantityAdded from
Customer_T,Order_T,Delivery_T,WishList_T where
Customer_T.CustomerID=Order_T.CustomerID and
WishList_T.WishListID=Order_T.WishListID and
Delivery_T.OrderID = Order_T.OrderID and
Delivery_T.ModeofDelivery = 'courier'
```

**VIEW 4: Quality Check - Finance Management**

Finance Management Team of T-Shirt memories have specialized views to monitor the quality of orders, loss incurred due to re-work and to view rating details.

The Quality Check details view would help T-shirt Memories in the following ways:

- Give an overview of how many orders failed to meet requirements and needed re-work.
- Track the Tailor and Machine Operator who worked on that order
- To Know the Manager who has done the quality check
- Amount of loss for the company due to re-work on failed order.

```
create table QualityCheck_view
(
OrderID int not null,
Approval varchar(2),
ManagerEmployeeID int not null,
TailorEmployeeID int not null,
MachineOperatorEmployeeID int not null,
TotalLoss decimal(5,2))
```

```
CREATE PROCEDURE RefreshQualitycheck_view as
delete from QualityCheck_view
insert into QualityCheck_view
select Order_T.OrderID, QualityCheck_T.Approval, QualityCheck_T.mEmployeeID,
Employee_Tailor_T.tEmployeeID , Employee_MachineOperator_T.moEmployeeID,
(- Expectedprofit- costofproduction) as TotalLoss
from
Order_T,QualityCheck_T,Employee_Tailor_T,Employee_Manager_T, Employee_MachineOperator_T,
MachineUsageTime_T,
(select (cost_of_production_table.costofproduction-order_t.ordertotal) as Expectedprofit, costofproduction,
order_t.OrderID
 from order_t,
(select order_t.OrderID,
(QuantityUsed_T.NumberofYards * QuantityPurchased_T.CostperYard) as costofproduction
from QuantityUsed_T,QuantityPurchased_T,Fleece_T,Order_T
where
QuantityPurchased_T.FleeceID=Fleece_T.FleeceID and
Order_T.OrderID = QuantityUsed_T.OrderID and
QuantityUsed_T.FleeceID=Fleece_T.FleeceID ) as cost_of_production_table
where Order_T.OrderID=cost_of_production_table.OrderID) as ExpectedProfit_table

where Order_T.OrderID= QualityCheck_T.OrderID and
Employee_Manager_T.mEmployeeID=QualityCheck_T.mEmployeeID and
Order_T.tEmployeeID = Employee_Tailor_T.tEmployeeID and
QualityCheck_T.Approval = 'ko' and
MachineUsageTime_T.OrderID = Order_T.OrderID and
MachineUsageTime_T.moEmployeeID = Employee_MachineOperator_T.moEmployeeId and
Order_T.OrderID=ExpectedProfit_table.OrderID
group by Order_T.OrderID, QualityCheck_T.Approval, QualityCheck_T.mEmployeeID,
Employee_Tailor_T.tEmployeeID, Employee_MachineOperator_T.moEmployeeID,
ExpectedProfit_table.Expectedprofit, ExpectedProfit_table.costofproduction
```

## DATABASE TRIGGERS

### Trigger: Cost Update Log

- The cost update log trigger would help T-Shirt Memories to keep a record of all the changes done to the cost per yard of the quantity purchased from fabric store.
- This Trigger will keep the record of old cost per yard, new cost per yard and date changed.
- This Trigger helps in management to identify any inappropriate attempts made to change the cost. If the cost per yard is inappropriately increased or decreased, it will have huge affect on cost of production and order total. It helps to maintain a clear record of audits.

```
create table Cost_Updates_Log
(FleeceID int not null,
UnitCostold decimal(4,2),
UnitCostnew decimal(4,2),
EmployeeIDold int,EmployeeIDnew int,
Updated char(1),Deleted char(1),UpdateDate datetime)

create trigger CostUpdate on QuantityPurchased_T
for update, delete
as declare @FleeceID int, @UnitCostold decimal(4,2), @UnitCostnew decimal(4,2),
@EmployeeIDold int , @EmployeeIDnew int

if (exists (select * from deleted) and exists (select * from inserted))
begin
insert into Cost_Updates_Log(FleeceID,UnitCostold,UnitCostnew, updated, deleted,
updatedate)
select inserted.FleeceID, inserted.CostperYard,deleted.CostperYard, 'Y','N', getdate()
```

```
from inserted, deleted
where inserted.Fleeceid = deleted.FleeceID
end

if (exists (select * from deleted) and not (exists (select * from inserted)))
begin
insert into Cost_Updates_Log (FleeceID,UnitCostold,UnitCostnew,updated, deleted,
updatedate)
select deleted.FleeceID, deleted.CostperYard, null, 'N', 'Y', getdate() from deleted
end
```