**Final Project**

James McColl

ASU ID: 1217045596

Arizona State University

IFT 300 #91025: Intermediate Database Management Systems

Professor Pramod Kumar

November 30, 2021

Table of Contents

[Database Project Domain: Aerospace Manufacturing 3](#_Toc89238200)

[Entities to be Modeled 4](#_Toc89238201)

[Entity Relationship Diagram 5](#_Toc89238202)

[Tables of Required Entities 6](#_Toc89238203)

[Database and Table Creation DDL Statements 7](#_Toc89238204)

[Data Insertion DML Statements 9](#_Toc89238205)

[DML Statements (Select Queries) 11](#_Toc89238206)

[Query 1: Which employee is managing each contract? 11](#_Toc89238207)

[Query 2: Which CNC machines are available for use? 12](#_Toc89238208)

[Query 3: Which parts are necessary for each assembly? 13](#_Toc89238209)

[Query 4: What is the balance due for each invoice? 14](#_Toc89238210)

[Query 5: What is the total amount owed to each supplier where the total is greater than $100? 15](#_Toc89238211)

[Query 6: Which supplier supplied the material for each order? 16](#_Toc89238212)

[Query 7: Which parts are NOT used in an assembly? 17](#_Toc89238213)

[Query 8: Which machines will be available next and who are the current machinists? 18](#_Toc89238214)

[Summary 19](#_Toc89238215)

[References 20](#_Toc89238216)

**Final Project**

# Database Project Domain: Aerospace Manufacturing

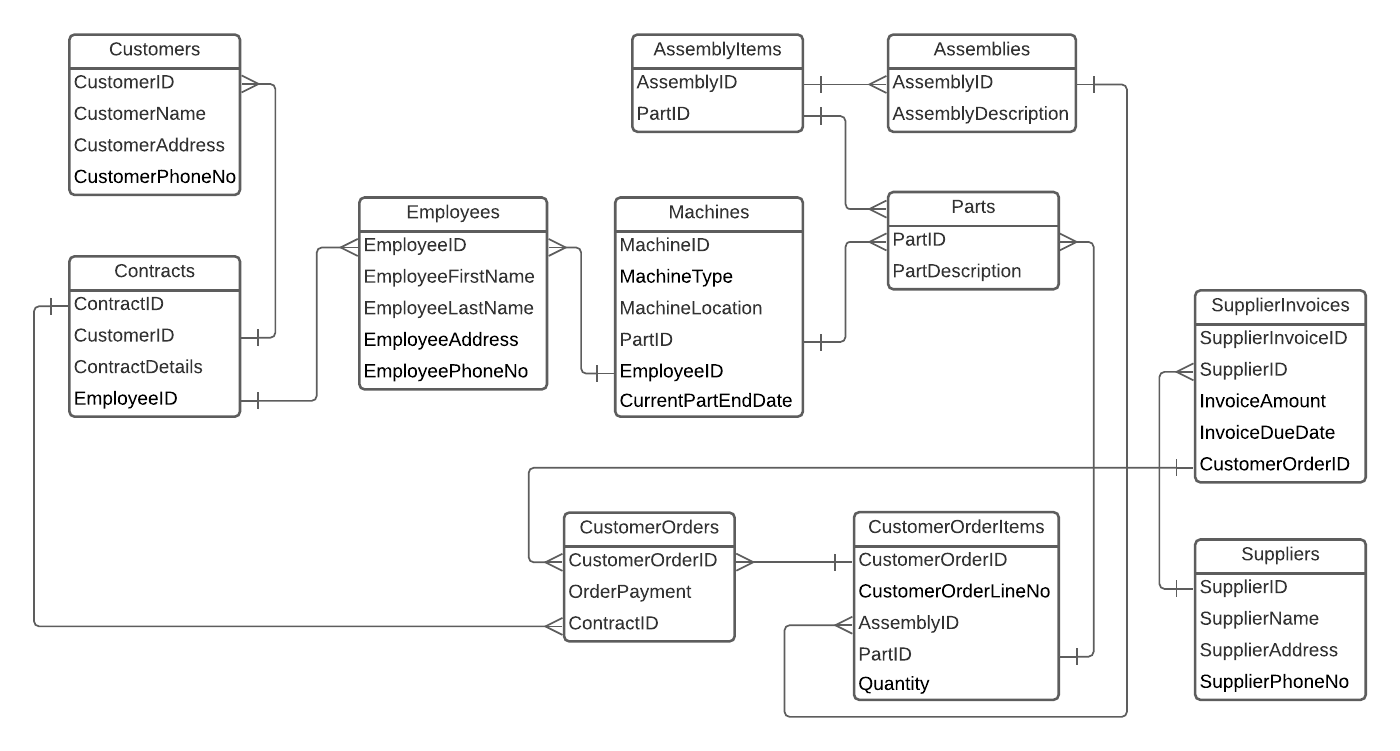
In CNC (Computer Numerical Control) machine shops, there are an incredible number of moving pieces that must be well-documented and properly tracked. There is also a need for streamlined processes that will increase efficiency through optimized use of machinery and labor hours. Once contracts are won to manufacture a specific part or assembly, a product manager is assigned to track the lifecycle of the manufacturing process and engineers must develop a specific process for each part. For each order within the contract, the source material must be purchased from an approved supplier and details of the purchase must be documented. The material is then given to the CNC programmers and machinists for production. The machines must be utilized to their greatest potential while keeping track of deadlines for each order. After the manufacturing process is complete, the orders must be packaged and delivered by the shipping team and invoices should be sent upon delivery. Because of these needs, a robust and functional database system is critical when fulfilling the daily tasks of an Aerospace Manufacturing shop. The manufacturing database will answer the following questions:

1. Which employee is managing each contract?
2. Which CNC machines are available for use?
3. Which parts are necessary for each assembly?
4. What is the balance due for each invoice?
5. What is the total amount owed to each supplier where the total is greater than $100?
6. Which supplier supplied the material for each order?
7. Which parts are NOT used in an assembly?
8. Which machines will be available next and who are the current machinists?

# Entities to be Modeled

* Employees will have general information and title listed and may have their ID assigned to contracts, orders, or machines.
* Assemblies can be purchased through orders and will include each part within the assembly.
* Parts can be part of assemblies or ordered directly by the customer and will have part descriptions and specifications listed.
* Machines within the shop can be assigned a machinist and a part over specific time periods.
* Customers and their information will be linked to the contracts they have with the machine shop
* Contracts will include the terms of each contract and the customer.
* CustomerOrders – Each order should be related to a contract and will include all parts and assemblies ordered along with the price.
* Suppliers will be linked to the parts they supply raw material for and will send invoices.
* SupplierInvoices will be received by suppliers and contain payment information.

# Entity Relationship Diagram



# 

# Tables of Required Entities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table of Required DML Statements** | | | | | | |
|  | **Inner Join** | **Outer Join** | **Group By** | **Having** | **Aggregations** | **Alias & Computation** |
| **Tables** | CustomerOrders,  SupplierInvoices,  Suppliers,  Contracts,  Customers,  Employees | Parts,  AssemblyItems | Supplier-  Invoices,  Suppliers | Supplier-  Invoices,  Suppliers | Supplier-  Invoices,  Suppliers | Employees |
| **Query** | Query 6 | Query 7 | Query 5 | Query 5 | Query 5 | Query 8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table of Required Data Types** | | | | |
|  | **Date** | **Decimal** | **Integer** | **Varchar** |
| **Tables** | SupplierInvoices,  Machines | CustomerOrders,  SupplierInvoices | Employees,  Customers,  Suppliers | Employees,  Customers,  Suppliers |
| **Attributes** | InvoiceDueDate,  CurrentPartEndDate | OrderPayment,  InvoiceAmount | EmployeeID,  CustomerID,  SupplierID | EmployeeAddress,  CustomerAddress,  SupplierAddress |

# Database and Table Creation DDL Statements

/\* Created by James McColl on 11/30/21 \*/

USE master

GO

CREATE DATABASE MachineShop

GO

USE MachineShop

GO

CREATE TABLE Employees(

EmployeeID INT PRIMARY KEY NOT NULL,

EmployeeFirstName VARCHAR(20) NOT NULL,

EmployeeLastName VARCHAR(20) NOT NULL,

EmployeeAddress VARCHAR(50) NOT NULL,

EmployeePhoneNo VARCHAR (20) NOT NULL

)

GO

CREATE TABLE Customers(

CustomerID INT PRIMARY KEY NOT NULL,

CustomerName VARCHAR(50) NOT NULL,

CustomerAddress VARCHAR(50) NOT NULL,

CustomerPhoneNo VARCHAR(20) NOT NULL

)

GO

CREATE TABLE Contracts(

ContractID INT PRIMARY KEY NOT NULL,

CustomerID INT REFERENCES Customers(CustomerID) NOT NULL,

ContractDetails VARCHAR(100) NOT NULL,

EmployeeID INT REFERENCES Employees(EmployeeID) NOT NULL

)

GO

CREATE TABLE CustomerOrders(

CustomerOrderID INT PRIMARY KEY NOT NULL,

OrderPayment DECIMAL(19,2) NOT NULL,

ContractID INT REFERENCES Contracts(ContractID)

)

GO

CREATE TABLE Suppliers(

SupplierID INT PRIMARY KEY NOT NULL,

SupplierName VARCHAR(50) NOT NULL,

SupplierAddress VARCHAR(50) NOT NULL,

SupplierPhoneNo VARCHAR(20) NOT NULL

)

GO

CREATE TABLE SupplierInvoices(

SupplierInvoiceID INT PRIMARY KEY NOT NULL,

SupplierID INT REFERENCES Suppliers(SupplierID) NOT NULL,

InvoiceAmount DECIMAL(19,2) NOT NULL,

InvoiceDueDate DATE NOT NULL,

CustomerOrderID INT REFERENCES CustomerOrders(CustomerOrderID) NOT NULL

)

GO

CREATE TABLE Parts(

PartID VARCHAR(50) PRIMARY KEY NOT NULL,

PartDescription VARCHAR(50) NOT NULL

)

GO

CREATE TABLE [Assemblies](

AssemblyID VARCHAR(50) PRIMARY KEY NOT NULL,

AssemblyDescription VARCHAR(50) NOT NULL

)

GO

CREATE TABLE AssemblyItems(

AssemblyID VARCHAR(50) REFERENCES [Assemblies](AssemblyID) NOT NULL,

PartID VARCHAR(50) REFERENCES Parts(PartID) NOT NULL

)

GO

CREATE TABLE CustomerOrderItems(

CustomerOrderID INT REFERENCES CustomerOrders(CustomerOrderID) NOT NULL,

CustomerOrderLineNo INT NOT NULL,

AssemblyID VARCHAR(50) REFERENCES [Assemblies](AssemblyID) NULL,

PartID VARCHAR(50) REFERENCES Parts(PartID) NULL,

Quantity INT NOT NULL,

PRIMARY KEY (CustomerOrderID, CustomerOrderLineNo),

CHECK (

(AssemblyID IS NULL AND PartID IS NOT NULL)

OR (PartID IS NULL AND AssemblyID IS NOT NULL)

)

)

GO

CREATE TABLE Machines(

MachineID INT PRIMARY KEY NOT NULL,

MachineType VARCHAR(20) NOT NULL,

MachineLocation VARCHAR(20) NOT NULL,

PartID VARCHAR(50) REFERENCES Parts(PartID) NULL,

EmployeeID INT REFERENCES Employees(EmployeeID) NULL,

CurrentPartEndDate DATE NULL,

CHECK (

MachineType = 'Lathe' OR

MachineType = 'Vertical Mill' OR

MachineType = 'Horizontal Mill' OR

MachineType = '5-Axis Mill'

)

)

GO

**Output:**

**Graphical user interface, application

Description automatically generated**

# Data Insertion DML Statements

USE MachineShop

GO

INSERT Employees

VALUES

(4567,'Tim','Lee','123 Oak Street','123-555-1831'),

(3920,'Bob','Scotch','234 Pine Street','123-555-0313'),

(1734,'Sam','Hill','345 Meadow Road','123-555-5123'),

(9472,'Melissa','Shore','456 Red Street','234-555-6234'),

(4352,'Jody','Hughes','567 Elm Street','123-555-9403')

GO

INSERT Customers

VALUES

(27,'High Flyers Inc.','753 Flyer Road','234-555-8342'),

(33,'Fighters','324 Tough Street','234-555-2634'),

(34,'Bob''s Assembly','843 Make Street','234-555-1254')

GO

INSERT Contracts

VALUES

(142,27,'12-Month Contract with 1 order per month',4567),

(172,27,'12-Month Contract with 1 order per month',4567),

(189,27,'12-Month Contract with 1 order per month',4567),

(99,33,'6-Month Contract with 2 orders per month',9472),

(151,34,'6-Month Contract with 3 orders per month',9472)

GO

INSERT Parts

VALUES

('172-A73','Wing Plate'),

('834-B21','Gear Shaft'),

('192-C89','Rotor Blade'),

('273-FF1','Front Connector'),

('833-DD1','Loose End'),

('413-A9M','Main Housing'),

('625-D1D','Motor Stabilizer'),

('236-9JF','Big Unit'),

('163-8DF','Tight Gyro'),

('755-ABC','Opposite Dongle'),

('645-JWI','Shock Enhancer'),

('942-1JJ','Tiny Lever')

GO

INSERT [Assemblies]

VALUES

('ROTO-MAN24','Main Rotor Assembly'),

('WING-THING7','Left Wing Assembly')

GO

INSERT AssemblyItems

VALUES

('ROTO-MAN24','192-C89'),

('ROTO-MAN24','625-D1D'),

('ROTO-MAN24','163-8DF'),

('ROTO-MAN24','273-FF1'),

('WING-THING7','172-A73'),

('WING-THING7','833-DD1'),

('WING-THING7','755-ABC')

GO

INSERT CustomerOrders

VALUES

(428,1111.32,142),

(491,842.12,172),

(510,937.11,189),

(397,2801.98,99),

(448,1587.09,151)

GO

INSERT CustomerOrderItems

VALUES

(428,1,NULL,'942-1JJ',8),

(428,2,NULL,'834-B21',8),

(491,1,NULL,'236-9JF',16),

(510,1,NULL,'413-A9M',4),

(510,2,NULL,'645-JWI',12),

(397,1,'ROTO-MAN24',NULL,1),

(448,1,'WING-THING7',NULL,3)

GO

INSERT Suppliers

VALUES

(13,'Raw Metal','513 Flyer Road','234-555-5111'),

(22,'Purity Promised','875 Tough Street','234-555-5132'),

(17,'Good Stuff Here','432 Make Street','123-555-5943')

GO

INSERT SupplierInvoices

VALUES

(174,13,272.22,'12-21-2021',428),

(189,22,132.01,'12-27-2021',491),

(201,22,227.10,'12-11-2021',510),

(155,13,141.89,'01-22-2022',397),

(133,17,89.37,'01-05-2022',448)

GO

INSERT Machines

VALUES

(1,'Lathe','A6','625-D1D',3920,'12-05-2021'),

(2,'Vertical Mill','B3','172-A73',1734,'12-06-2021'),

(3,'Horizontal Mill','D1','645-JWI',4352,'12-04-2021'),

(4,'5-Axis Mill','A3',NULL,NULL,NULL)

GO

**Graphical user interface, application

Description automatically generated**

# DML Statements (Select Queries)

## Query 1: Which employee is managing each contract?

SELECT

CustomerName,

ContractID,

EmployeeFirstName + ' ' + EmployeeLastName AS 'Program Manager',

ContractDetails

FROM Customers

JOIN Contracts

ON Customers.CustomerID = Contracts.CustomerID

JOIN Employees

ON Contracts.EmployeeID = Employees.EmployeeID

ORDER BY CustomerName, ContractID

**Graphical user interface, text, email

Description automatically generated**

## Query 2: Which CNC machines are available for use?

SELECT

MachineID,

MachineType,

MachineLocation

FROM Machines

WHERE PartID IS NULL AND EmployeeID IS NULL

**Graphical user interface, text, application, email

Description automatically generated**

## Query 3: Which parts are necessary for each assembly?

SELECT

AssemblyItems.\*,

PartDescription

FROM AssemblyItems JOIN Parts

ON AssemblyItems.PartID = Parts.PartID

WHERE AssemblyID = 'ROTO-MAN24'

**Graphical user interface, text, application, email

Description automatically generated**

SELECT

AssemblyItems.\*,

PartDescription

FROM AssemblyItems JOIN Parts

ON AssemblyItems.PartID = Parts.PartID

WHERE AssemblyID = 'WING-THING7'

**Graphical user interface, text, application, email

Description automatically generated**

## Query 4: What is the balance due for each invoice?

SELECT

SupplierName,

Suppliers.SupplierID,

SupplierInvoiceID,

FORMAT (InvoiceAmount, 'C') AS 'Amount Owed',

InvoiceDueDate

FROM SupplierInvoices JOIN Suppliers

ON SupplierInvoices.SupplierID = Suppliers.SupplierID

ORDER BY SupplierID, [Amount Owed] DESC

**Graphical user interface, text, application, email

Description automatically generated**

## Query 5: What is the total amount owed to each supplier where the total is greater than $100?

SELECT

SupplierName,

Suppliers.SupplierID,

FORMAT (SUM(InvoiceAmount), 'C') AS 'Total Amount Owed',

FORMAT (MAX(InvoiceAmount), 'C') AS 'Largest Invoice',

MIN(SupplierInvoices.InvoiceDueDate) AS 'Earliest Due Date'

FROM SupplierInvoices JOIN Suppliers

ON SupplierInvoices.SupplierID = Suppliers.SupplierID

GROUP BY Suppliers.SupplierID, SupplierName

HAVING SUM(InvoiceAmount) > 100

ORDER BY SUM(InvoiceAmount) DESC

**Graphical user interface, text, application

Description automatically generated**

## Query 6: Which supplier supplied the material for each order?

SELECT

EmployeeFirstName + ' ' + EmployeeLastName AS 'Program Manager',

CustomerName AS Customer,

CustomerOrders.CustomerOrderID,

SupplierName AS 'Raw Material Supplier'

FROM CustomerOrders

JOIN SupplierInvoices

ON CustomerOrders.CustomerOrderID = SupplierInvoices.CustomerOrderID

JOIN Suppliers

ON SupplierInvoices.SupplierID = Suppliers.SupplierID

JOIN Contracts

ON CustomerOrders.ContractID = Contracts.ContractID

JOIN Customers

ON Contracts.CustomerID = Customers.CustomerID

JOIN Employees

ON Contracts.EmployeeID = Employees.EmployeeID

ORDER BY Customer

**Graphical user interface, text, application, email

Description automatically generated**

## Query 7: Which parts are NOT used in an assembly?

SELECT Parts.PartID, PartDescription

FROM Parts LEFT JOIN AssemblyItems

ON Parts.PartID = AssemblyItems.PartID

WHERE AssemblyID IS NULL

**Graphical user interface, text, application, email

Description automatically generated**

## Query 8: Which machines will be available next and who are the current machinists?

SELECT

MachineID,

MachineType,

MachineLocation,

EmployeeFirstName + ' ' + EmployeeLastName AS 'Machinist',

CurrentPartEndDate AS 'Date Available'

FROM Machines JOIN Employees

ON Machines.EmployeeID = Employees.EmployeeID

WHERE CurrentPartEndDate =

(SELECT MIN(CurrentPartEndDate)

FROM Machines)

**Graphical user interface, text, application

Description automatically generated**

# Summary

This was a great exercise that displayed the value that a relational database can have on an industry that heavily relies on tracking and accountability of many different parts, people, and equipment. This design made it incredibly easy to pull information relating all the way from the source material to the end product and customer delivery, resulting in an efficient and relatively painless audit process. This model could be enhanced by building out more details of the company infrastructure and setting up a table to hold data for every individual serialized part.

# References

Murach, J., & Syverson, B. (2020). Murach's SQL Server 2019 for Developers. Mike Murach & Associates.