**Database Model of a Truck Maintenance Company**  
  
Learning Team-A  
Bachmeier and Miller  
DBM 380: Database Concepts  
University of Phoenix  
October 14th, 2013  
Instructor: Sandra La Vallee

Table of Contents

[Database Model of a Truck Maintenance Company 3](#_Toc371001293)

[Vendor 3](#_Toc371001294)

[Constraints 3](#_Toc371001295)

[Parts 3](#_Toc371001296)

[Constraints 3](#_Toc371001297)

[ContactType 4](#_Toc371001298)

[Contact 4](#_Toc371001299)

[City 4](#_Toc371001300)

[ContractType 5](#_Toc371001301)

[PurchaseRecord 5](#_Toc371001302)

[InventoryActivityTransaction 5](#_Toc371001303)

[TransactionType 6](#_Toc371001304)

[Manufacturer 6](#_Toc371001305)

[VehicleType 6](#_Toc371001306)

[Vehicle 7](#_Toc371001307)

[VehnicleMaintenance 7](#_Toc371001308)

[TireMaintenance 8](#_Toc371001309)

[MaintenanceRecord 9](#_Toc371001310)

[WorkOrder 9](#_Toc371001311)

[OrderCatalogue 10](#_Toc371001312)

[MaintenanceJob 10](#_Toc371001313)

[Database Schema Not Normalized 11](#_Toc371001314)

[Database Schema Normalized 11](#_Toc371001315)

[Example Queries 12](#_Toc371001316)

[Cost of Labor 12](#_Toc371001317)

[Owners Cars 13](#_Toc371001318)

# Database Model of a Truck Maintenance Company

## Vendor

This entity outlines all of the Vendors that the Huffman Trucking Company utilizes. The Vendor entity relates to the Parts Catalogue and Contacts entities. A Vendor may have 1 or many parts they provide. A Vendor may have 1 or more contacts, billing or order.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| VendorId | Integer (PK) | Primary Identifier of Vendor |
| Name | Nvarchar(50) | Name of the vendor |

## Parts

This entity outlines all of the Parts that Huffman Trucking has on hand. A part has several types of basic information such as a name, type, etc. A part is related to a Vendor that Huffman orders from and also relates to a Manufacturer that creates the part.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| PartId | Integer (PK) | Primary Identifier of Parts |
| VendorId | Integer (FK) | Reference to the part vendor |
| PartName | Nvarchar(255) | Name of the part |
| PartDescription | Nvarchar(max) | Description of the part |
| ManufactureId | Integer (FK) | Reference to the Manufacturer |
| QuantityInStock | Integer | Count of the number |
| ReorderPoint | Integer | Minimum stock point |

## ContactType

This entity outlines the Contacts for the Vendors that Huffman Trucking does business with. A Vendor may have 1 or more contacts, such as a Billing Contact and an Order Contact.   
Contact types is a Text PK that helps identify a unique contact type. This attribute is an intelligent key vs. a non-intelligent key, which can be used on its own to perform basic filtering.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| ContractTypeId | Integer(PK) | Primary ID |
| ContractType | Integer(FK) | Type of the Contract |

## Contact

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| ContactId | Integer(PK) | Primary ID |
| ContractTypeId | Integer(FK) | Reference to ContractType |
| VendorId | Integer(FK) | Reference to Vendor |
| Address | NVARCHAR(200) | Shipping address |
| AddressUnitNumber | NVARCHAR(200) | Suite or unit number |
| CityId | Integer(FK) | Reference to City |
| PrimaryContactName | NVARCHAR(100) | Name of the primary contact |
| PhoneNumber | CHAR(11) | Phone number of the contact |
| FaxNumber | CHAR(11) NULLABLE | Fax number of the contact |

## City

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| CityId | Integer(PK) | Primary ID |
| CityName | NVARCHAR(100) | Name of the city |
| State | Integer(FK) | Contact’s state of residence |
| ZipCode | CHAR(10) | Contact’s zip code |

## **ContractType**

This entity is a reference table and simply helps identify the Vendor contract type. This entity will reduce maintenance and improve data consistency.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| ContractTypeId | Integer (PK) | Primary identifier |
| Description | NVARCHAR(255) | Description of the contract |

## PurchaseRecord

This entity outlines the Parts that Huffman Trucking has purchased from its Vendors. A transaction is considered a purchase of a specific part.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| PurchaseRecordId | Integer (PK) | Primary Identifier |
| CatalogId | Integer (FK) | Reference Catalog |
| QuantityPurchased | Integer | Total number of parts purchased |
| Price | Money | Price per item in U.S. Currency |

## InventoryActivityTransaction

This entity outlines inventory activity, such as adds or deletes. A delete would be damaged part returned to a Vendor.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| SalesRecordId | Integer (PK) | Primary Identifier |
| TransactionTypeId | Integer (FK) | Identifies TransactionType |
| TransactionDate | DateTime | Date transaction occurred |
| PricePerUnit | Money | Price per unit |
| TotalUnits | Integer | Number of units in the transaction |

## TransactionType

This entity is a reference table and simply helps identify the transaction type. This entity will reduce maintenance and improve data consistency and relates to the Inventory Activity entity.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer (PK) | Primary Identifier |
| Description | NVARCHAR(100) | Description of the transaction |

## Manufacturer

This entity relates to the Parts Catalogue along with the TireMaintenance entity. In short a part has one and only one Vendor, but a Vendor can have one or more parts.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| ManufactuerId | Integer (PK) | Identifier |
| Name | NVARCHAR(100) | Name of the manufacturer |

## VehicleType

This entity is a reference table and simply helps identify the vehicle description. This entity will ensure consistency.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer (PK) | Primary identifier |
| Description | NVARCHAR(100) | Description of the vehicle type |

## 

## Vehicle

This entity stores the primary characteristics of the vehicles Huffman Trucking owns. This entity relates to the VehicleMaintenace, TireMaintenance, and MaintenanceWorkOrder entities. The vehicle table relates to the VehicleMaintenane and TireMaintenance as a 1 to 1 and acts as more or less a summary table. Whereas the relationship between the vehicle and relationships between the vehicle and MaintenanceWorkOrder is 1 to many.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| Id | Integer (PK) | Identifies Vehicle |
| VIN | CHAR(17) | Key that identifies a unique vehicle |
| VehicleTypeId | Integer (FK) | Relates to VehicleType |
| ClassificationId | Integer (FK) | Classification of the vehicle |
| Weight | Integer | Weight of the vehicle |
| Mileage | Integer | Mileage of the vehicle |
| Capacity | Integer | Amount of freight the vehicle can haul |

## VehnicleMaintenance

This entity is primarily a summary table which will store information as to the current state of maintenance on a vehicle,   and a method to track last maintenance and next maintenance attributes for a vehicle. This entity has a 1 to 1 relationship with the vehicle entity.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| ID | Integer (PK) | Identifies VehicleMaintenance |
| VehicleId | Integer (FK) | Refers to a unique vehicle |
| MaintenanceTypeId | Integer (FK) | Refers to MaintenanceTypeID |
| MaintenanceDate | DateTime | The date maintenance was performed |
| NextScheduledDate | DateTime nullable | Date of next scheduled maintenance |
| Under Warranty | Bit | Whether vehicle is under warranty |

## TireMaintenance

This entity is used to track what types of tires are installed on each vehicle and what the state of said maintenance is for that vehicle / tire combination. It relates to the vehicle table having a 1 to 1 relationship.  
MaintenanceWorkOrderID is an AutoNumber PK which uniquely identifies a specific work order.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| MaintenanceWorkOrderId | AutoNumber(PK) | Identifies the maintenance work order |
| TireBarcode | CHAR(15) | Individual tire barcode |
| MaintainceTypeId | Integer (FK) | Refers to VehicleMaintenance |
| TireMaintenceDate | DateTime | Date of the tire maintenance |
| RotationMileage | Integer | Mileage since last rotation |
| NextSchedule | Datetime | Next scheduled maintenance |
| DisposalDate | Date | Date of a tire’s disposal |

## MaintenanceRecord

This entity will act as the primary table for tracking the maintenance of a single vehicle. That is, it will relate to the vehicle table via the VIN. Hence, a vehicle can have a single or many work orders where as a single work order can only relate to one vehicle.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| MaintenanceRecordID | AutoNumber | Identifies MaintenanceRecord |
| VIN | Text (FK) | Refers to a specific vehicle |
| MaintenaceTypeID | Text (FK) | Refers to MaintenanceType |
| DateStart | Date/Time | Start date of the maintenance |
| DateComplete | Date/Time | Completion date for the maintenance |

## WorkOrder

This is a child table to the parent table MaintenanceWorkOrder that is joined by the WorkOrderID. The purpose of this table is to track each line item/job that makes up a work order. Therefore, a work order can have one or more line items but a single line item can only belong to one work order. In addition, MaintenanceWorkOrderLine will also join to the orders table via the WorkOrderLineID. This will ultimately allow the tracking of each individual part to be associated to a specific maintenance line item by then joining to the parts catalogue.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| WorkOrderId | Integer (PK) | Identifies WorkOrder |
| WorkOrderTypeId | Integer (FK) | Identifies the type of work order |
| Description | Text | Description of work order |
| StartDate | DateTime | Start date of work order |
| CompletionDate | DateTime nullable | Completion date of work order |

## OrderCatalogue

This table will serve as an associative table that will bridge the gap between the MaintenanceWorkOrderLine table and the PartsCatalogue table. This will allow a work order line to have one or many parts and a single part or many parts can belong to a MaintenanceWorkOrderLine.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| CatalogueId | Integer (PK) | Identifies OrderCatalogue |
| PartNumber | Integer (PK) | Identifies individual parts |

## MaintenanceJob

This entity is a reference table and simply helps identify the various types of maintenance. This entity will reduce data maintenance and improve data consistency.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| MaintenanceJobId | Integer (PK) | Identifies MaintenanceJob |
| LevelCode | CHAR(15) | Number designating degree of scope |
| Description | NVARCHAR(255) | Description of a particular job |
| HoursRequired | Float | Hours required for a job |
| RecommendedSchedule | Timespan | Recommended time between services |
| MaximumSchedule | Timespan | Maximum time between services |

# Database Schema Not Normalized

The next step in creating a database is to establish the relationships between entities in a database. This is generally done through an Entity Relationship Diagram or ERD. In an ERD, the relationships between entities are drawn using lines between each other. An ERD can help to design a database in a much more efficient manner, allowing the constructors to see exactly how the database will function when completed. There are three different types of relationships for better organization: One-to-One, One-to-Many, and Many-to-Many. A one-to-one relationship is a relation between a single row in one table and a single row in another. A one-to-many relationship is a table where a single row of one table relates to multiple items on another table. A many-to-many relationship is where multiple rows from one table relate to multiple rows from another table.

In order to create the ERD relationships need to be mapped out. For example, one entity is the vehicles entity. This entity will have a one-to-many relationship with other entities. For example, the car may have multiple work orders done on it and therefore will need to connect to all of the associated maintenance entities. The entities have been laid out in the chart above and will be organized into tables with attributes displayed below the name of the table. In order to ensure that each table is unique, the top attribute in each table will be assigned as the primary key. This will also allow these keys to be inserted into other tables in the ERD plan. This will also allow for foreign keys to be inserted into other tables where necessary.



# Database Schema Normalized



All tables are joined on foreign keys that match the name of primary key.

For instance-

Select \*

From EmployeePayRate r inner join Employee e

On r.EmployeeId = e.EmployeeId

# Example Queries

## Cost of Labor

Select Description as LaborType, sum( p.Salary \* lc.Hours ) as LaborCost

From ServiceRequest r

inner join LaborCost lc

On r.ServiceRequestId =lc.ServiceRequestId

Inner join Employee e

Cross Apply CurrentPay(e.EmployeeId) p

Inner join LaborType lt

On lc.LaborTypeId = lt.LaborTypeId

Group by lt.Description

Order by LaborType asc

## Owners Cars

Select o.[Billing Name], s.Name as State, v.VIN, m.Description as Make, mo.Description as Model

From Owner o

inner join State s

On o.StateId = s.StateId

Inner join Vehicle v

On o.OwnerId = v.OwnerId

Inner join VehicleMake m

On v.MakeId = m.MakeId

Inner Join VehilceModel mo

On v.ModelId = mo.ModelId