

Homework 4: Database Design/Modeling

Muntaha Pasha

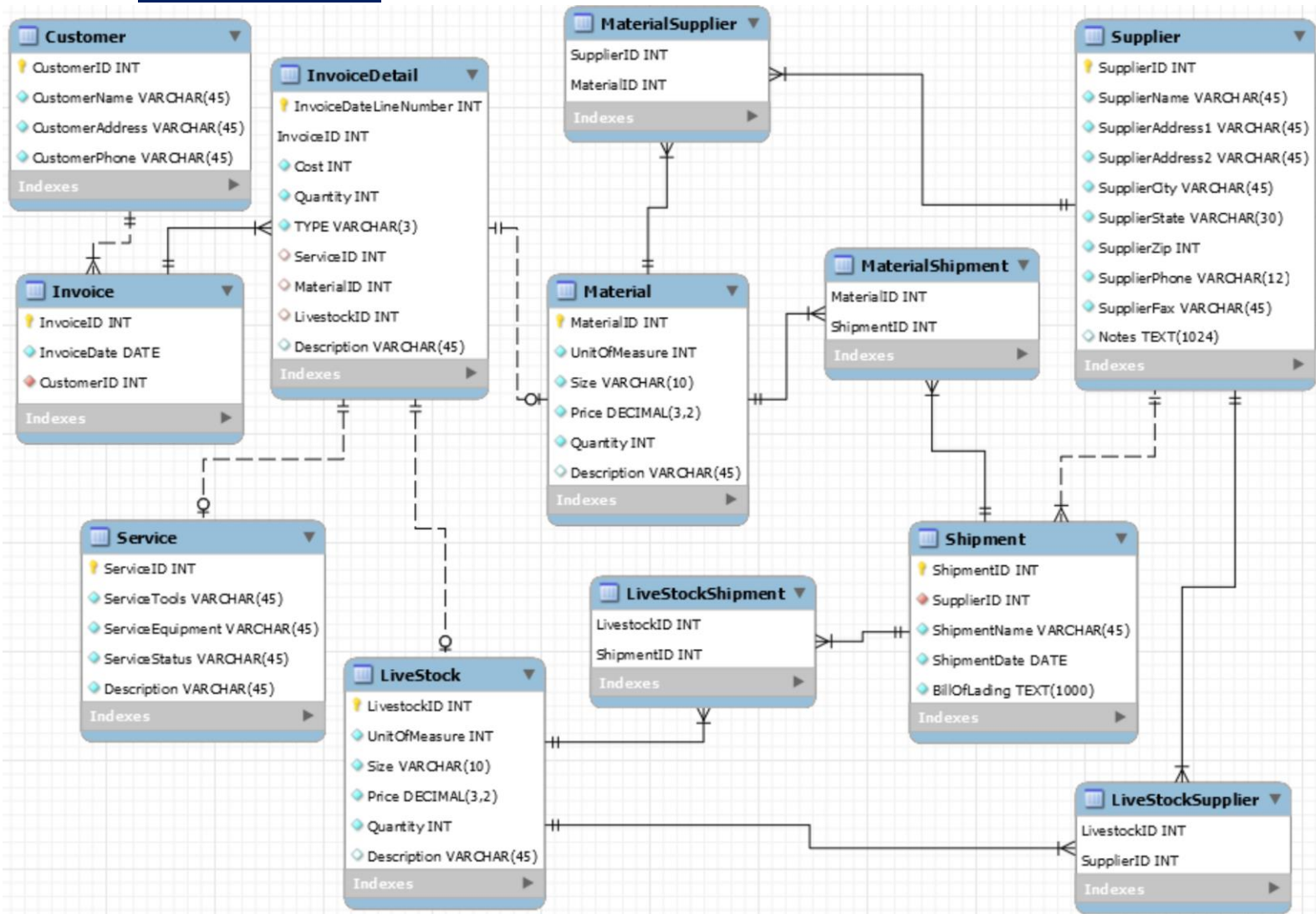
Small Guide:

Data Model – Pg 1

DDL Scripts – Pg 2-10

Assumptions – Pg 11

PART 1: Data Model



PART 2: DDL Scripts

```
-- MySQL Script generated by MySQL Workbench
-- Wed Apr  3 23:26:26 2019
-- Model: New Model      Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema mydb
-- -----

-- -----
-- Schema mydb
-- -----

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----
-- Table `mydb`.`Customer`
-- -----

DROP TABLE IF EXISTS `mydb`.`Customer` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Customer` (
  `CustomerID` INT NOT NULL,
  `CustomerName` VARCHAR(45) NOT NULL,
  `CustomerAddress` VARCHAR(45) NOT NULL,
  `CustomerPhone` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`CustomerID`))
ENGINE = InnoDB;
```

```

-----
-- Table `mydb`.`Invoice`
-----

DROP TABLE IF EXISTS `mydb`.`Invoice` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Invoice` (
  `InvoiceID` INT NOT NULL,
  `InvoiceDate` DATE NOT NULL,
  `CustomerID` INT NOT NULL,
  PRIMARY KEY (`InvoiceID`),
  CONSTRAINT `CustomerID`
    FOREIGN KEY (`CustomerID`)
      REFERENCES `mydb`.`Customer` (`CustomerID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_Invoice_Customer_idx` ON `mydb`.`Invoice` (`CustomerID` ASC) VISIBLE;

-----
-- Table `mydb`.`Service`
-----

DROP TABLE IF EXISTS `mydb`.`Service` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Service` (
  `ServiceID` INT NOT NULL,
  `ServiceTools` VARCHAR(45) NOT NULL,
  `ServiceEquipment` VARCHAR(45) NOT NULL,
  `ServiceStatus` VARCHAR(45) NOT NULL,
  `Description` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ServiceID`))

```

```
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Material`  
-- -----  
  
DROP TABLE IF EXISTS `mydb`.`Material` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Material` (  
  `MaterialID` INT NOT NULL,  
  `UnitOfMeasure` INT NOT NULL,  
  `Size` VARCHAR(10) NOT NULL,  
  `Price` DECIMAL(3,2) NOT NULL,  
  `Quantity` INT NOT NULL,  
  `Description` VARCHAR(45) NULL,  
  PRIMARY KEY (`MaterialID`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`LiveStock`  
-- -----  
  
DROP TABLE IF EXISTS `mydb`.`LiveStock` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`LiveStock` (  
  `LivestockID` INT NOT NULL,  
  `UnitOfMeasure` INT NOT NULL,  
  `Size` VARCHAR(10) NOT NULL,  
  `Price` DECIMAL(3,2) NOT NULL,  
  `Quantity` INT NOT NULL,  
  `Description` VARCHAR(45) NULL,  
  PRIMARY KEY (`LivestockID`))  
ENGINE = InnoDB;
```

```

-----
-- Table `mydb`.`InvoiceDetail`
-----

DROP TABLE IF EXISTS `mydb`.`InvoiceDetail` ;

CREATE TABLE IF NOT EXISTS `mydb`.`InvoiceDetail` (
  `InvoiceDateLineNumber` INT NOT NULL,
  `InvoiceID` INT NOT NULL,
  `Cost` INT NOT NULL,
  `Quantity` INT NOT NULL,
  `TYPE` VARCHAR(3) NOT NULL,
  `ServiceID` INT NULL,
  `MaterialID` INT NULL,
  `LivestockID` INT NULL,
  `Description` VARCHAR(45) NULL,
  PRIMARY KEY (`InvoiceDateLineNumber`, `InvoiceID`),
  CONSTRAINT `InvoiceID`
    FOREIGN KEY (`InvoiceID`)
      REFERENCES `mydb`.`Invoice` (`InvoiceID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ServiceID`
    FOREIGN KEY (`ServiceID`)
      REFERENCES `mydb`.`Service` (`ServiceID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `MaterialID`
    FOREIGN KEY (`MaterialID`)
      REFERENCES `mydb`.`Material` (`MaterialID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `LivestockID`
    FOREIGN KEY (`LivestockID`)

```

```

REFERENCES `mydb`.`LiveStock` (`LivestockID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE INDEX `fk_InvoiceDetail_Service1_idx` ON `mydb`.`InvoiceDetail` (`ServiceID`
ASC) VISIBLE;

CREATE INDEX `InvoiceID_idx` ON `mydb`.`InvoiceDetail` (`InvoiceID` ASC) VISIBLE;

CREATE INDEX `MaterialID_idx` ON `mydb`.`InvoiceDetail` (`MaterialID` ASC) VISIBLE;

CREATE INDEX `LivestockID_idx` ON `mydb`.`InvoiceDetail` (`LivestockID` ASC) VISIBLE;

-- -----
-- Table `mydb`.`Supplier`
-- -----

DROP TABLE IF EXISTS `mydb`.`Supplier` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Supplier` (
  `SupplierID` INT NOT NULL,
  `SupplierName` VARCHAR(45) NOT NULL,
  `SupplierAddress1` VARCHAR(45) NOT NULL,
  `SupplierAddress2` VARCHAR(45) NOT NULL,
  `SupplierCity` VARCHAR(45) NOT NULL,
  `SupplierState` VARCHAR(30) NOT NULL,
  `SupplierZip` INT NOT NULL,
  `SupplierPhone` VARCHAR(12) NOT NULL,
  `SupplierFax` VARCHAR(45) NOT NULL,
  `Notes` TEXT(1024) NULL,
  PRIMARY KEY (`SupplierID`))
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`Shipment`
-- -----

DROP TABLE IF EXISTS `mydb`.`Shipment` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Shipment` (
  `ShipmentID` INT NOT NULL,
  `SupplierID` INT NOT NULL,
  `ShipmentName` VARCHAR(45) NOT NULL,
  `ShipmentDate` DATE NOT NULL,
  `BillofLading` TEXT(1000) NOT NULL,
  PRIMARY KEY (`ShipmentID`),
  CONSTRAINT `SupplierID`
    FOREIGN KEY (`SupplierID`)
      REFERENCES `mydb`.`Supplier` (`SupplierID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `SupplierID_idx` ON `mydb`.`Shipment` (`SupplierID` ASC) VISIBLE;

-- -----
-- Table `mydb`.`MaterialSupplier`
-- -----

DROP TABLE IF EXISTS `mydb`.`MaterialSupplier` ;

CREATE TABLE IF NOT EXISTS `mydb`.`MaterialSupplier` (
  `SupplierID` INT NOT NULL,
  `MaterialID` INT NOT NULL,
  PRIMARY KEY (`SupplierID`, `MaterialID`),
  CONSTRAINT `MaterialID`
    FOREIGN KEY (`MaterialID`)

```

```

REFERENCES `mydb`.`Material` (`MaterialID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `SupplierID`
FOREIGN KEY (`SupplierID`)
REFERENCES `mydb`.`Supplier` (`SupplierID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `MaterialID_idx` ON `mydb`.`MaterialSupplier` (`MaterialID` ASC) VISIBLE;

-- -----
-- Table `mydb`.`LiveStockSupplier`
-- -----

DROP TABLE IF EXISTS `mydb`.`LiveStockSupplier` ;

CREATE TABLE IF NOT EXISTS `mydb`.`LiveStockSupplier` (
  `LivestockID` INT NOT NULL,
  `SupplierID` INT NOT NULL,
  PRIMARY KEY (`LivestockID`, `SupplierID`),
  CONSTRAINT `LivestockID`
    FOREIGN KEY (`LivestockID`)
    REFERENCES `mydb`.`LiveStock` (`LivestockID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `SupplierID`
    FOREIGN KEY (`SupplierID`)
    REFERENCES `mydb`.`Supplier` (`SupplierID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```
CREATE INDEX `fk_LiveStockSupplier_Supplier1_idx` ON `mydb`.`LiveStockSupplier`  
(`SupplierID` ASC) VISIBLE;
```

```
-- -----  
-- Table `mydb`.`MaterialShipment`  
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`MaterialShipment` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`MaterialShipment` (  
  `MaterialID` INT NOT NULL,  
  `ShipmentID` INT NOT NULL,  
  PRIMARY KEY (`MaterialID`, `ShipmentID`),  
  CONSTRAINT `MaterialID`  
    FOREIGN KEY (`MaterialID`)  
      REFERENCES `mydb`.`Material` (`MaterialID`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `ShipmentID`  
    FOREIGN KEY (`ShipmentID`)  
      REFERENCES `mydb`.`Shipment` (`ShipmentID`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
CREATE INDEX `ShipmentID_idx` ON `mydb`.`MaterialShipment` (`ShipmentID` ASC) VISIBLE;
```

```
-- -----  
-- Table `mydb`.`LiveStockShipment`  
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`LiveStockShipment` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`LiveStockShipment` (  
  `MaterialID` INT NOT NULL,  
  `ShipmentID` INT NOT NULL,  
  PRIMARY KEY (`MaterialID`, `ShipmentID`),  
  CONSTRAINT `MaterialID`  
    FOREIGN KEY (`MaterialID`)  
      REFERENCES `mydb`.`Material` (`MaterialID`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `ShipmentID`  
    FOREIGN KEY (`ShipmentID`)  
      REFERENCES `mydb`.`Shipment` (`ShipmentID`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

`LivestockID` INT NOT NULL,
`ShipmentID` INT NOT NULL,
PRIMARY KEY (`LivestockID`, `ShipmentID`),
CONSTRAINT `LivestockID`
    FOREIGN KEY (`LivestockID`)
    REFERENCES `mydb`.`LiveStock` (`LivestockID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `ShipmentID`
    FOREIGN KEY (`ShipmentID`)
    REFERENCES `mydb`.`Shipment` (`ShipmentID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `ShipmentID_idx` ON `mydb`.`LiveStockShipment` (`ShipmentID` ASC)
VISIBLE;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

PART 3: Assumptions/Notes

- One assumption I had to make during this modeling process was how much information I wanted to include where. For instance, Invoice and Customers is relatively empty because I felt as if what was really most important to the scope was Invoice Detail, and how it connected to Services, Materials, and Livestock, and the way Quantity factored into it.
- I had to assume various data types for some of the table entities. For example I can assume that by size, they'd want to input something like "14 inches" if it's a bush, or "5 feet" if it's a tree. Maybe for a plank of wood, size refers to "4x4", or "12x8", so these sizes can be represented uniquely in characters.
- Another assumption I had to make was regarding what entities to leave as NULL, and what not to leave as NULL. In my opinion, for the most part a description can be NULL, but in the instance of services, I think having a description is very important. Other than that though, it was just following the business on what details are crucial and should mandatorily be in the database, and some personal touches on what I believed was important information.
- I had to assume in my services table that there is an entity for "service status". What this handles is the case that, say that landscaping provides a fencing service, and all the machines happen to be in repair. Status for that type of service would be "Not Available", and it would become "Available" as soon as the machines are fixed. It's crucial for the business to know what services they are fully ready to perform, and if things do break down, the status will change dependent on that and they'll be able to answer to customers' needs more efficiently.
- I had to assume different cardinalities for the table relationships, like assuming a customer can have multiple invoices... or assuming that suppliers have multiple shipments, assuming the cardinality between invoice details to services, materials, and livestock is one to one, respectively.
- I had to create associative entities to resolve the many to many issues between materials, shipments and suppliers, and between livestock, shipments and suppliers. So for those, 4 new tables were added to the model.
- Assuming that Invoice Detail is what mainly affects the inventory processes. In Scope.
- Cost is how much you pay the vendor, whereas Price is how much customer pays you.
- **NOTE:** In my version of SQL, for some reason the **RED KEY** symbol for foreign keys is not displayed. In the tables, "MaterialSupplier, MaterialShipment, LiveStockSupplier, LiveStockShipment", the entities are displayed WITHOUT a symbol. My version of SQL just happens to for some reason forgo that symbol, so I just want to make it clear that the entities in there are foreign keys. You will also notice other entities in my tables without symbols. Same applies for them. If they are at the most-left side of the table without a key or diamond symbol, they are a foreign key, just missing the red key symbol which usually appears by them.