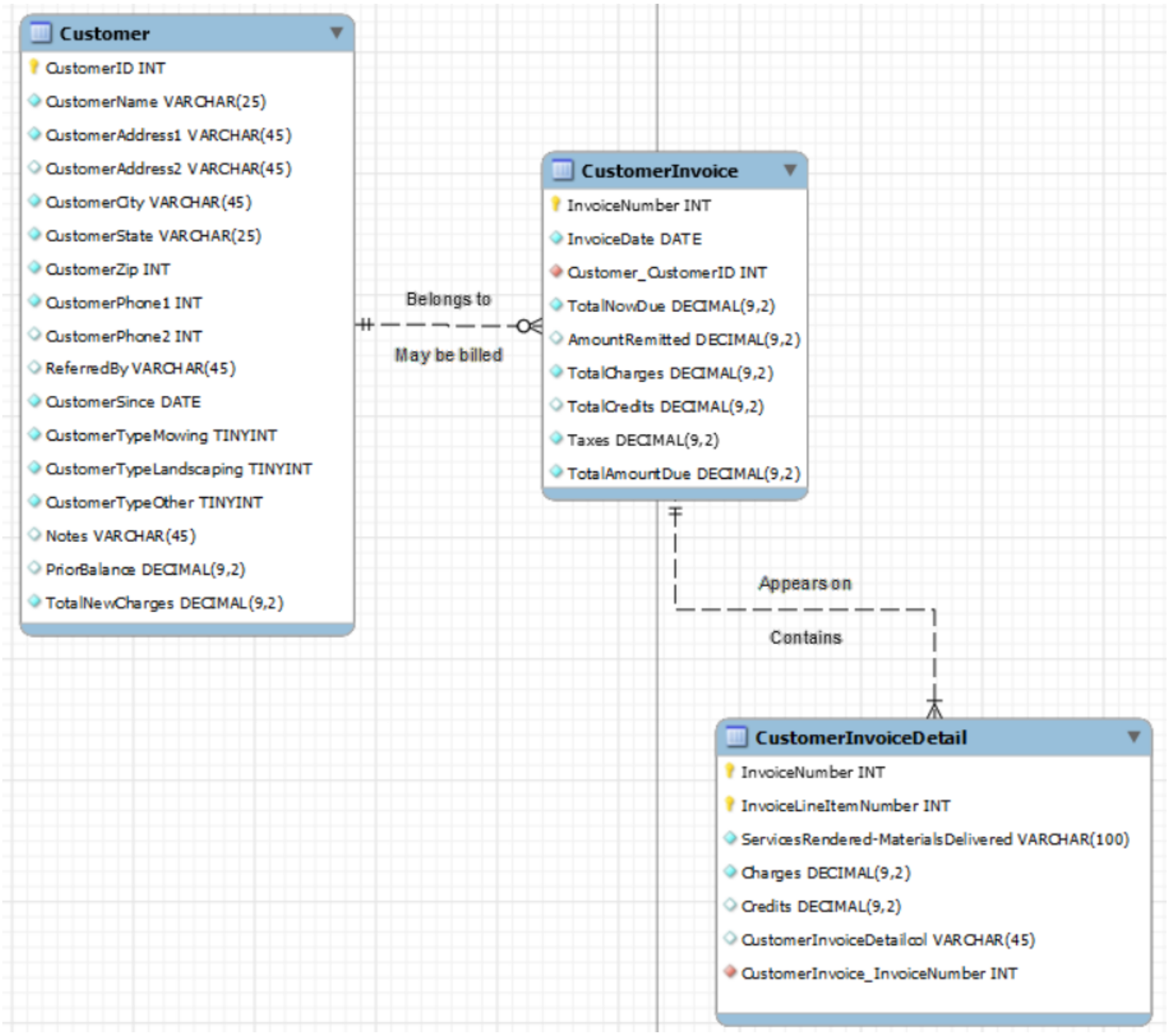
Helen Kim
CSCI 3287-001
Homework 2 - Database Design

Database Model:



Customer

- CustomerID INT
- CustomerName VARCHAR(25)
- CustomerAddress1 VARCHAR(45)
- CustomerAddress2 VARCHAR(45)
- CustomerCity VARCHAR(45)
- CustomerState VARCHAR(25)
- CustomerZip INT
- CustomerPhone1 INT
- CustomerPhone2 INT
- ReferredBy VARCHAR(45)
- CustomerSince DATE
- CustomerTypeMowing TINYINT
- CustomerTypeLandscaping TINYINT
- CustomerTypeOther TINYINT
- Notes VARCHAR(45)
- PriorBalance DECIMAL(9,2)
- TotalNewCharges DECIMAL(9,2)

CustomerInvoice

- InvoiceNumber INT
- InvoiceDate DATE
- Customer_CustomerID INT
- TotalNowDue DECIMAL(9,2)
- AmountRemitted DECIMAL(9,2)
- TotalCharges DECIMAL(9,2)
- TotalCredits DECIMAL(9,2)
- Taxes DECIMAL(9,2)
- TotalAmountDue DECIMAL(9,2)

CustomerInvoiceDetail

- InvoiceNumber INT
- InvoiceLineItemNumber INT
- ServicesRendered-MaterialsDelivered VARCHAR(100)
- Charges DECIMAL(9,2)
- Credits DECIMAL(9,2)
- CustomerInvoiceDetailcol VARCHAR(45)
- CustomerInvoice_InvoiceNumber INT

DDL for Database designed:

-- MySQL Script generated by MySQL Workbench
-- Sun Sep 26 13:30:58 2021
-- 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_FOR_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 AUTO_INCREMENT,
  `CustomerName` VARCHAR(25) NOT NULL,
  `CustomerAddress1` VARCHAR(45) NOT NULL,
  `CustomerAddress2` VARCHAR(45) NULL,
  `CustomerCity` VARCHAR(45) NOT NULL,
  `CustomerState` VARCHAR(25) NOT NULL,
  `CustomerZip` INT NOT NULL,
  `CustomerPhone1` INT NOT NULL,
  `CustomerPhone2` INT NULL,
  `ReferredBy` VARCHAR(45) NULL,
  `CustomerSince` DATE NOT NULL,
  `CustomerTypeMowing` TINYINT NOT NULL,
  `CustomerTypeLandscaping` TINYINT NOT NULL,
  `CustomerTypeOther` TINYINT NOT NULL,
  `Notes` VARCHAR(45) NULL,
  `PriorBalance` DECIMAL(9,2) NULL,
  `TotalNewCharges` DECIMAL(9,2) NOT NULL,
  PRIMARY KEY (`CustomerID`))
```

```
ENGINE = InnoDB;


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

CREATE TABLE IF NOT EXISTS `mydb`.`CustomerInvoice` (
  `InvoiceNumber` INT NOT NULL,
  `InvoiceDate` DATE NOT NULL,
  `Customer_CustomerID` INT NOT NULL,
  `TotalNowDue` DECIMAL(9,2) NOT NULL,
  `AmountRemitted` DECIMAL(9,2) NULL,
  `TotalCharges` DECIMAL(9,2) NOT NULL,
  `TotalCredits` DECIMAL(9,2) NULL,
  `Taxes` DECIMAL(9,2) NOT NULL,
  `TotalAmountDue` DECIMAL(9,2) NOT NULL,
  PRIMARY KEY (`InvoiceNumber`),
  INDEX `fk_Invoice_Customer_idx` (`Customer_CustomerID` ASC) VISIBLE,
  CONSTRAINT `fk_Invoice_Customer`
    FOREIGN KEY (`Customer_CustomerID`)
    REFERENCES `mydb`.`Customer` (`CustomerID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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

CREATE TABLE IF NOT EXISTS `mydb`.`CustomerInvoiceDetail` (
  `InvoiceNumber` INT NOT NULL,
  `InvoiceLineItemNumber` INT NOT NULL,
  `ServicesRendered-MaterialsDelivered` VARCHAR(100) NOT NULL,
  `Charges` DECIMAL(9,2) NOT NULL,
  `Credits` DECIMAL(9,2) NULL,
  `CustomerInvoiceDetailcol` VARCHAR(45) NULL,
  `CustomerInvoice_InvoiceNumber` INT NOT NULL,
  PRIMARY KEY (`InvoiceNumber`, `InvoiceLineItemNumber`),
  INDEX `fk_CustomerInvoiceDetail_CustomerInvoice1_idx`
(`CustomerInvoice_InvoiceNumber` ASC) VISIBLE,
  CONSTRAINT `fk_CustomerInvoiceDetail_CustomerInvoice1`
    FOREIGN KEY (`CustomerInvoice_InvoiceNumber`)
    REFERENCES `mydb`.`CustomerInvoice` (`InvoiceNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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

Database Relation explanations:

Customer:

- Declared the primary key to be the customer ID since name isn’t always unique but ID would be, which is why a surrogate primary key was made for this table.
- Made most information not null as it would seem to be required information, but for info like phone number 2 or address two it was made to not necessarily have to be not null it is not required information most of the time as it is not something the majority of the people have compared to their first (main) address
- The CustomerType such as mowing would be not null since it is a statement that the service people would know as to whether it is desired or not, especially since the variable type is boolean (which defaulted to tiny int in the data table)
- Notes can be null as there may not be any extra info needed overall other than what is already displayed on the datatable
- Prior balance can also be null as they could be a first time customer or just don’t have any previous balance history but total new charge would have to be not null since the current transaction exists even if the balance ends up being 0

CustomerInvoice:

- Like for the customer table, none of the existing attributes are always unique so we need an ID for the invoice as well and not just reuse something like customer ID since a customer could have multiple invoices. Hence why it is a 1:many relationship as well from customer to invoice
- The above is also why customer ID is a foreign key

- The relationship is “CustomerInvoice belongs to Customer” and CustomerInvoice may be billed to Customer”

CustomerInvoiceDetail

- We have two invoice detail primary keys, where one we need to identify which invoice document since the customer can have multiple invoices and we need to specify which one. Then invoiceLineItemNumber is also required for which of the services/materials are being billed in that specific invoice, hence why we need two primary keys. Both of these are surrogate keys because there are no other attributes that are unique to identify specific tables of data that would be used for the different invoice detail tables of each customer
- And we have the the CustomerInvoice as a primary key InvoiceID as a primary key due to us using that for identify which of the invoices the customer is referring to as stated before where the line item number refers to that (due to the fact that there may be repeat services or materials but it happens on different days or time)