****

**Title: ACIS 5504– Project 1 Database Development**

**Sub title: Milestone 2: Normalization and SQL DDL**

**Due Date: Sunday October 15th at 11:59 PM**

**Jeevan Thapa**

**CS5504 Information Systems Design and Database Concepts**

**Prof Name:**

**Course Instructor: Dr. Steven D. Sheetz**

**Course Distance Learning Iinstructor: Stephen Dargis**

**Course Distance Learning Iinstructor: Dr. David Henderson**

**Normalization:**

It is the process of designing a database table structure that organizes the data to minimize data redundancies, thereby reducing the likelihood of data anomalies. The main objective of normalizing the data is to isolate the information so modifications to the data can be performed in just one table and then propagate throughout the database by means of foreign keys. There are different set of rules in order to perform data normalization. Following are the types of normalization forms:

1. First Normal form (1NF)
2. Second Normal form (2NF)
3. Third Normal form (3NF)
4. Boyce-Codd Normal Form (BCNF)
5. Fourth Normal Form (4NF)

Below are the process I have followed for the GCS database design in 3NF:

1. **First Normal Form (1NF):**
2. Eliminated the repeating groups: Repeating groups are defined as multiple entries of same type can exist for any single key attribute occurrence. I have analyzed possible multi-valued attributes, and created a separate attribute for each single value, so that my attribute contains one single value all the time.
3. Identified the Primary Key: I have identified the primary keys of all entities.
4. Identify all dependencies: I have analyzed the possible functional dependencies (partial and transitive dependency).

**2. Second Normal Form (2NF):**

Conversion to 2NF occurs when the 1NF has a composite primary key. If the 1NF has a single attribute primary key then it is in 2NF.

1. Database tables are in 1NF: I have verified that my design did satisfy the 1NF guidelines.
2. I have made sure my database design is free from partial dependencies but still possible for a table to exhibit transitive dependency: Partial dependency exists when there is a functional dependence in which the determinant is only part of the primary key. If (A, B) is the primary key and (A, B) -> (C, D) and B ->C then B ->C is a partial dependency because only part of the primary key (B) is needed to determine C. I have designed my database so that all the non key attributes depends on composite primary keys.

**3. Third Normal Form (3NF):**

1. Database tables are in 2NF: I have verified my design is in 2NF.
2. It doesn’t contain transitive dependencies: Transitive dependency occurs when there are functional dependencies such that X->Y, Y->Z and X is the primary key. In that case the dependency X->Z is a transitive dependency because X determines value of Z via Y. I have made new tables to eliminate transitive dependencies. I have reassigned corresponding attributes where determinant remains in original table. I created a new table with a determinant being a primary key. I removed the dependents from the original table and placed as a non-prime attributes in the new table.

**SQL DDL that implements the design in the data dictionary:**

-- 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='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema liabrary

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema liabrary

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `liabrary` ;

USE `liabrary` ;

-- -----------------------------------------------------

-- Table `liabrary`.`REGION`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`REGION` (

`REGION\_ID` CHAR(2) NOT NULL,

`REGION\_DESCR` VARCHAR(45) NULL,

PRIMARY KEY (`REGION\_ID`),

UNIQUE INDEX `REGION\_DESCR\_UNIQUE` (`REGION\_DESCR` ASC))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`EMPLOYEE`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`EMPLOYEE` (

`EMP\_ID` INT NOT NULL AUTO\_INCREMENT,

`EMP\_REGION\_ID` CHAR(2) NULL,

`EMP\_LAST\_NAME` VARCHAR(45) NULL,

`EMP\_MI` VARCHAR(1) NULL,

`EMP\_FIRST\_NAME` VARCHAR(45) NULL,

`HIRE\_DT` DATE NULL,

PRIMARY KEY (`EMP\_ID`),

INDEX `REGION\_ID\_idx` (`EMP\_REGION\_ID` ASC),

CONSTRAINT `EMP\_REGION\_ID`

FOREIGN KEY (`EMP\_REGION\_ID`)

REFERENCES `liabrary`.`REGION` (`REGION\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`SKILL`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`SKILL` (

`SKILL\_ID` INT NOT NULL AUTO\_INCREMENT COMMENT 'Skill ID',

`SKILL\_EMP\_ID` INT NULL COMMENT 'Employee ID',

`SKILL\_DESCR` VARCHAR(45) NULL COMMENT 'Skill Description: Employee with skills',

`PAY\_RATE` DECIMAL(13,2) NULL COMMENT 'Pay Rate of Employee',

PRIMARY KEY (`SKILL\_ID`),

INDEX `EMPLOYEE\_ID\_idx` (`SKILL\_EMP\_ID` ASC),

CONSTRAINT `SKILL\_EMP\_ID`

FOREIGN KEY (`SKILL\_EMP\_ID`)

REFERENCES `liabrary`.`EMPLOYEE` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`CUSTOMER`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`CUSTOMER` (

`CUST\_ID` INT NOT NULL AUTO\_INCREMENT,

`CUST\_EMP\_ID` INT NULL,

`CUST\_REGION\_ID` CHAR(2) NULL,

`CUST\_NAME` VARCHAR(45) NULL,

`CUST\_PHN` INT(10) NULL,

PRIMARY KEY (`CUST\_ID`),

INDEX `EMP\_ID\_idx` (`CUST\_EMP\_ID` ASC),

INDEX `REGION\_ID\_idx` (`CUST\_REGION\_ID` ASC),

CONSTRAINT `CUST\_EMP\_ID`

FOREIGN KEY (`CUST\_EMP\_ID`)

REFERENCES `liabrary`.`EMPLOYEE` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `CUST\_REGION\_ID`

FOREIGN KEY (`CUST\_REGION\_ID`)

REFERENCES `liabrary`.`REGION` (`REGION\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`PROJECT`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`PROJECT` (

`PROJECT\_ID` INT NOT NULL AUTO\_INCREMENT,

`PROJECT\_CUST\_ID` INT NULL,

`MANAGER\_ID` INT NULL,

`PROJ\_DESCR` VARCHAR(45) NULL,

`PROJ\_CNTRCT\_DT` DATE NULL,

`EST\_PROJ\_STRT\_DT` DATE NULL,

`EST\_PROJ\_END\_DT` DATE NULL,

`EST\_PROJ\_BUDG` DECIMAL(13,2) NULL,

`ACT\_PROJ\_STRT\_DT` DATE NULL,

`ACT\_PROJ\_END\_DT` DATE NULL,

`ACT\_COST` DECIMAL(13,2) NULL,

PRIMARY KEY (`PROJECT\_ID`),

INDEX `MANAGER\_ID\_idx` (`MANAGER\_ID` ASC),

INDEX `PROJECT\_CUST\_ID\_idx` (`PROJECT\_CUST\_ID` ASC),

CONSTRAINT `MANAGER\_ID`

FOREIGN KEY (`MANAGER\_ID`)

REFERENCES `liabrary`.`EMPLOYEE` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `PROJECT\_CUST\_ID`

FOREIGN KEY (`PROJECT\_CUST\_ID`)

REFERENCES `liabrary`.`CUSTOMER` (`CUST\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`PROJ\_SCH\_TASK`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`PROJ\_SCH\_TASK` (

`TASK\_ID` INT NOT NULL AUTO\_INCREMENT,

`TASK\_PROJECT\_ID` INT NULL,

`TASK\_SKILL\_ID` INT NULL,

`TASK\_EMP\_ID` INT NULL,

`TASK\_DESCR` VARCHAR(45) NULL,

`TASK\_STRT\_DT` DATE NULL,

`TASK\_END\_DT` DATE NULL,

`TOTAL\_EMP\_REQ` INT NULL,

PRIMARY KEY (`TASK\_ID`),

INDEX `PROJECT\_ID\_idx` (`TASK\_PROJECT\_ID` ASC),

INDEX `SKILL\_ID\_idx` (`TASK\_SKILL\_ID` ASC),

INDEX `EMP\_ID\_idx` (`TASK\_EMP\_ID` ASC),

CONSTRAINT `TASK\_PROJECT\_ID`

FOREIGN KEY (`TASK\_PROJECT\_ID`)

REFERENCES `liabrary`.`PROJECT` (`PROJECT\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `TASK\_SKILL\_ID`

FOREIGN KEY (`TASK\_SKILL\_ID`)

REFERENCES `liabrary`.`SKILL` (`SKILL\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `TASK\_EMP\_ID`

FOREIGN KEY (`TASK\_EMP\_ID`)

REFERENCES `liabrary`.`EMPLOYEE` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`ASSIGNMENT`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`ASSIGNMENT` (

`ASSIGN\_ID` INT NOT NULL AUTO\_INCREMENT COMMENT 'Assignment ID',

`ASSIGN\_EMP\_ID` INT NULL,

`ASSIGN\_PROJ\_TASK\_ID` INT NULL COMMENT 'Project Task ID',

`ASSIGN\_STRT\_DT` DATE NULL COMMENT 'Assign Start Date',

`ASSIGN\_END\_DT` DATE NULL COMMENT 'Assign End Date',

PRIMARY KEY (`ASSIGN\_ID`),

INDEX `EMP\_ID\_idx` (`ASSIGN\_EMP\_ID` ASC),

INDEX `ASSIGN\_PROJ\_TASK\_ID\_idx` (`ASSIGN\_PROJ\_TASK\_ID` ASC),

CONSTRAINT `ASSIGN\_EMP\_ID`

FOREIGN KEY (`ASSIGN\_EMP\_ID`)

REFERENCES `liabrary`.`EMPLOYEE` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `ASSIGN\_PROJ\_TASK\_ID`

FOREIGN KEY (`ASSIGN\_PROJ\_TASK\_ID`)

REFERENCES `liabrary`.`PROJ\_SCH\_TASK` (`TASK\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`INVOICE`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`INVOICE` (

`INVOICE\_ID` INT NOT NULL AUTO\_INCREMENT,

`INVOICE\_CUST\_ID` INT NULL,

`INVOICE\_DT` VARCHAR(45) NULL,

PRIMARY KEY (`INVOICE\_ID`),

INDEX `INVOICE\_CUST\_idx` (`INVOICE\_CUST\_ID` ASC),

CONSTRAINT `INVOICE\_CUST`

FOREIGN KEY (`INVOICE\_CUST\_ID`)

REFERENCES `liabrary`.`CUSTOMER` (`CUST\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `liabrary`.`WORK\_LOG`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `liabrary`.`WORK\_LOG` (

`WRK\_LOG\_ID` INT NOT NULL AUTO\_INCREMENT,

`ASSIGN\_ID` INT NULL,

`INVOICE\_ID` INT NULL,

`CHRG\_DT` DATE NULL,

`WRKD\_HRS` INT NULL,

`CHRG\_RT` DECIMAL(13,2) NULL,

PRIMARY KEY (`WRK\_LOG\_ID`),

INDEX `ASSIGN\_ID\_idx` (`ASSIGN\_ID` ASC),

INDEX `INVOICE\_ID\_idx` (`INVOICE\_ID` ASC),

CONSTRAINT `ASSIGN\_ID`

FOREIGN KEY (`ASSIGN\_ID`)

REFERENCES `liabrary`.`ASSIGNMENT` (`ASSIGN\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `INVOICE\_ID`

FOREIGN KEY (`INVOICE\_ID`)

REFERENCES `liabrary`.`INVOICE` (`INVOICE\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

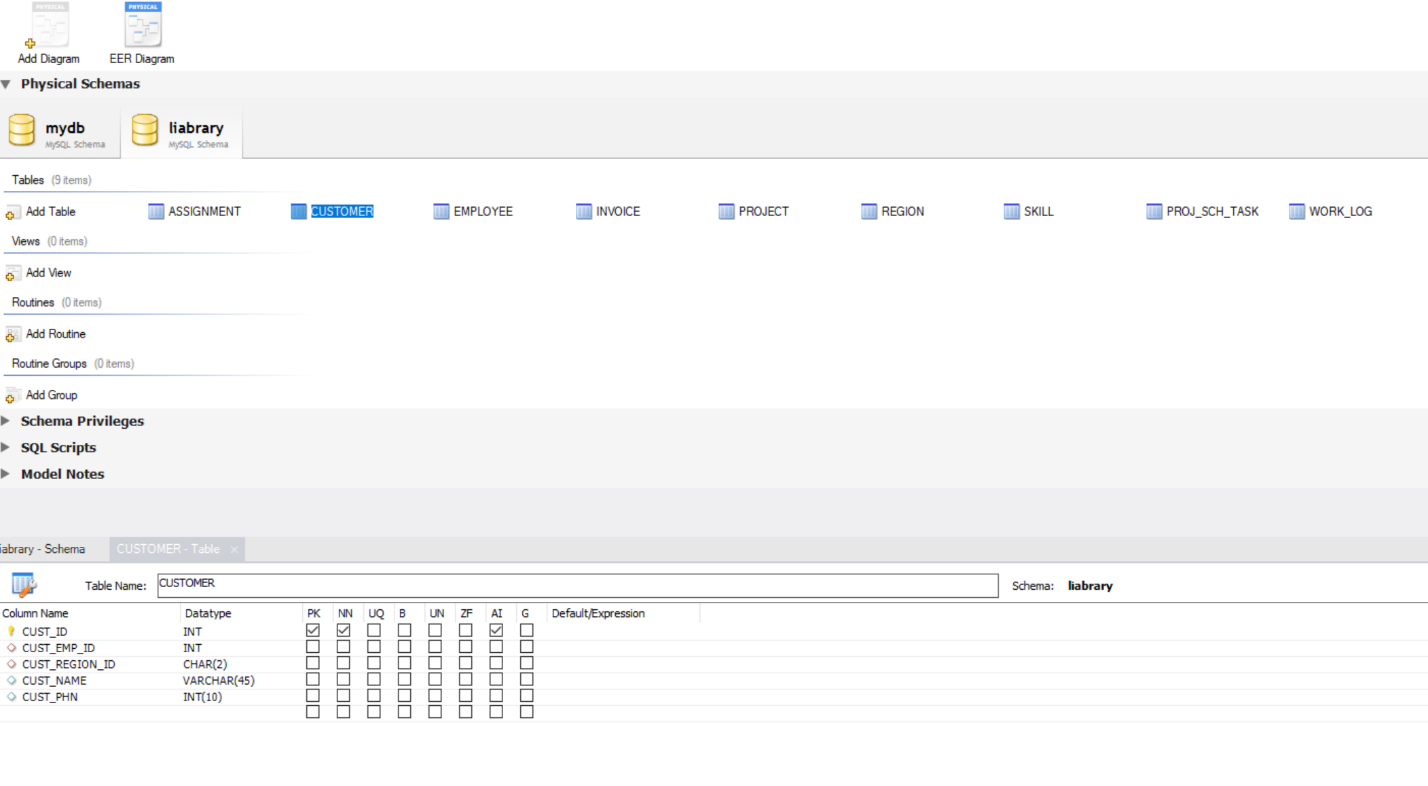
ENGINE = InnoDB;

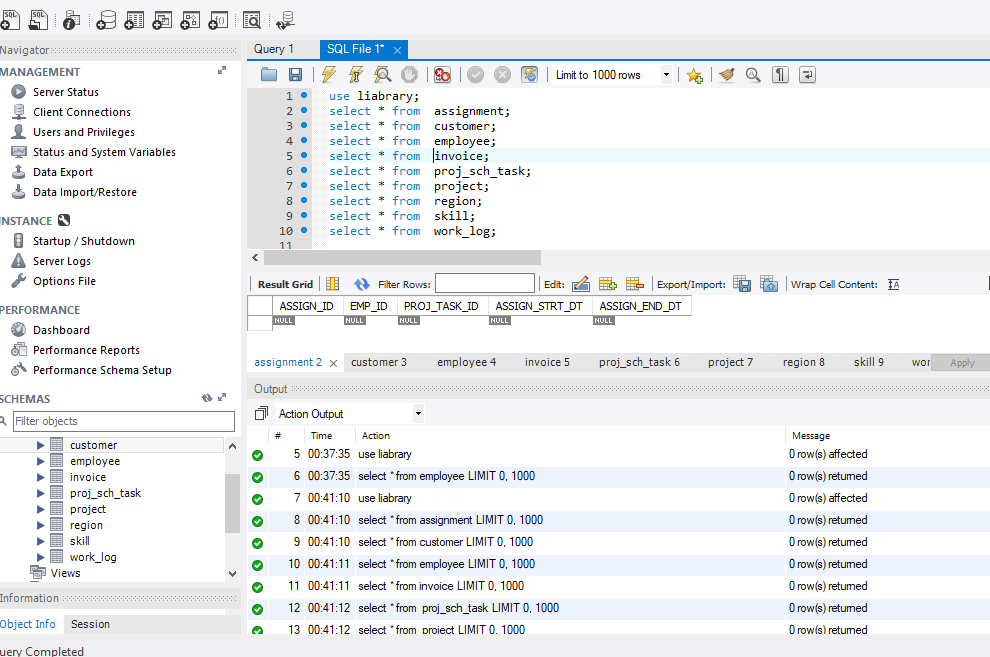
SET SQL\_MODE=@OLD\_SQL\_MODE;

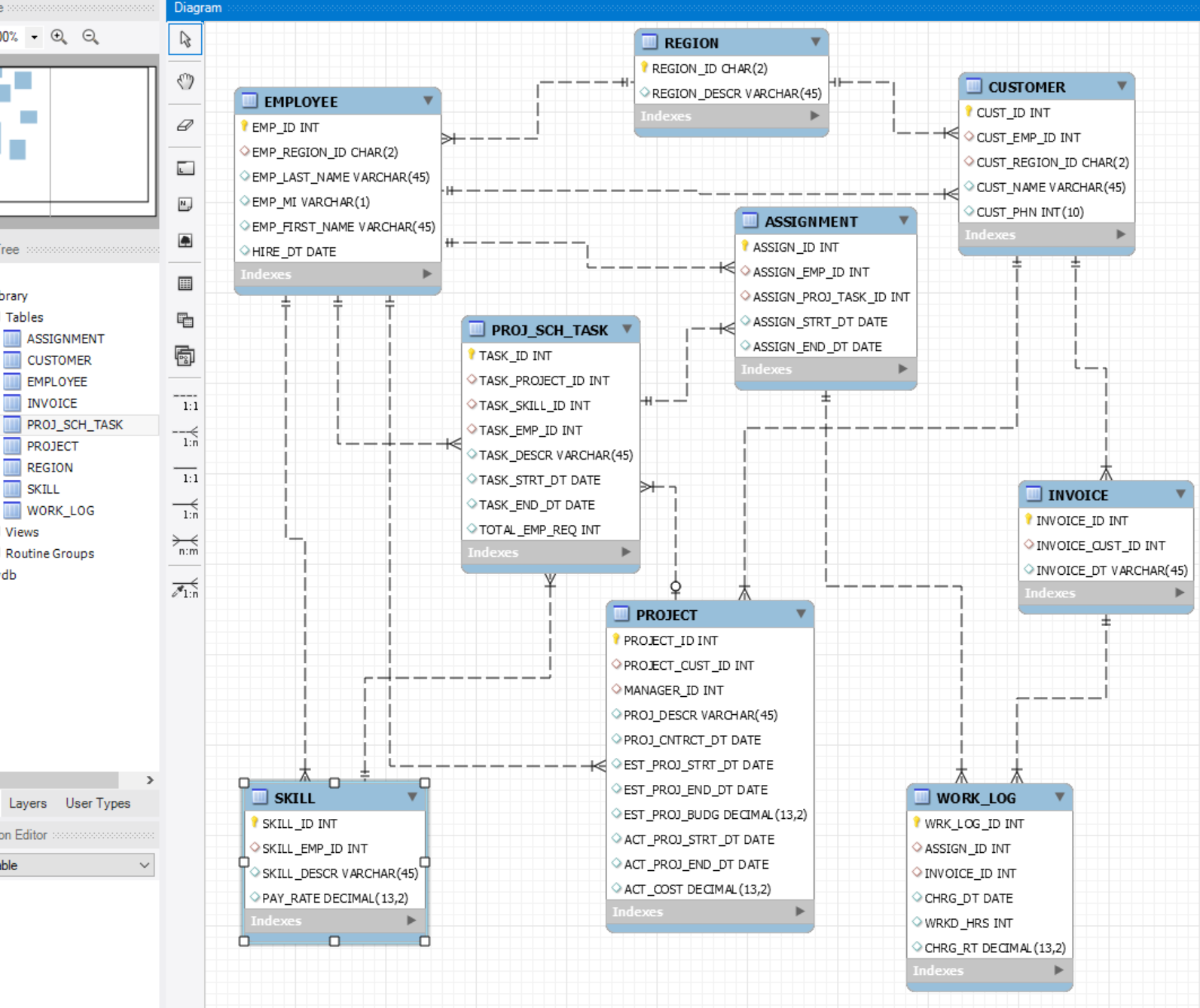
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Evidence of the creation of the tables using a MySQL DBMS:**

****

****

****

**Appendix:**

****