**Project Part 1:**

**Team Members**

**Abhishek Hire (HI6066)**

**Venkat Vamsi Krishna Vanganur (HM7462)**

**GitHub:**[**https://github.com/VenkatVamsiKrishnaVanganur/DSE-6100-Project-Part-1**](https://github.com/VenkatVamsiKrishnaVanganur/DSE-6100-Project-Part-1)

**Youtube Part 1:** [**https://youtu.be/yzGNCkvm4qI**](https://youtu.be/yzGNCkvm4qI)

**Hours Worked on Project Part 1:**

**Abhishek Hire: 6-Hours Login, Registration Page and Changes in Servlet and UserDAO Files.**

**Venkat Vamsi Krishna Vanganur: 6-Hours ER Diagram, Creation of Tables with Assertions and Creation of Tables in MySQL Server.**

**Table Client:**

* ClientID (Primary key)
* FirstName
* LastName
* Address
* CreditCard Info
* Phone
* Email

**SQL Statement:**

CREATE TABLE Client (

ClientID INT PRIMARY KEY,

FirstName VARCHAR (50),

LastName VARCHAR (50),

Address VARCHAR (100),

CreditCardInfo VARCHAR (16),

Phone VARCHAR (15),

Email VARCHAR (100) UNIQUE);

**Table Quote:**

* QuoteID (Primary key)
* TreeID (Foreign Key)
* InitialPrice
* WorkStartDate
* WorkEndDate
* Status
* Note

**SQL Statement:**

CREATE TABLE Quote (

QuoteID INT PRIMARY KEY,

TreeID INT,

InitialPrice DECIMAL (10,2),

WorkStartDate DATE,

WorkEndDate DATE,

Status VARCHAR (20),

Note TEXT,

FOREIGN KEY (TreeID) REFERENCES Tree (TreeID));

**Table Order:**

* OrderID (Primary key)
* QuoteID (Foreign key)
* ClientID (Foreign key)
* TreeID (Foreign key)
* Status

**SQL Statement:**

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

TreeID INT,

ClientID INT,

QuoteID INT,

Status VARCHAR (20),

FOREIGN KEY (TreeID) REFERENCES Tree (TreeID),

FOREIGN KEY (ClientID) REFERENCES Client (ClientID),

FOREIGN KEY (QuoteID) REFERENCES Quote (QuoteID)

);

**Table Bill:**

* BillID (Primary key)
* OrderID (Foreign Key)
* TotalAmount
* Status
* Note

**SQL Statement:**

CREATE TABLE Bill (

BillID INT PRIMARY KEY,

OrderID INT,

TotalAmount DECIMAL(10,2),

Status VARCHAR (20),

Note TEXT,

FOREIGN KEY (OrderID) REFERENCES Order (OrderID)

);

**Table Tree:**

* TreeID (Primary key)
* Size
* Height
* Location
* IsNearHouse
* Note

**SQL Statement:**

CREATE TABLE Tree (

TreeID INT PRIMARY KEY,

ClientID INT,

Size DECIMAL (5,2),

Height DECIMAL (5,2),

Location VARCHAR (100),

IsNearHouse BOOLEAN,

Note TEXT,

FOREIGN KEY (ClientID) REFERENCES Client (ClientID)

);

**ASSERTIONS:**

* Assertion to check Client Email Uniqueness:

CREATE ASSERTION UniqueEmail

CHECK ( (SELECT COUNT(\*) FROM Client) = (SELECT COUNT(DISTINCT Email) FROM Client));

* Assertion to check Initial Price Range between 100 and 10000:

CREATE ASSERTION InitialPriceRange

CHECK (

(SELECT COUNT(\*) FROM Quote WHERE InitialPrice BETWEEN 100 AND 10000) = (SELECT COUNT(\*) FROM Quote));

* Assertion to check if Order Status is valid:

CREATE ASSERTION ValidOrderStatus

CHECK (

(SELECT COUNT(\*) FROM Orders WHERE Status IN ('Pending', 'In Progress', 'Completed')) = (SELECT COUNT(\*) FROM Orders));

* Assertion to check Bill Total amount is non negative:

CREATE ASSERTION NonNegativeTotalAmount

CHECK (

(SELECT COUNT(\*) FROM Bill WHERE TotalAmount >= 0) = (SELECT COUNT(\*) FROM Bill));

* Assertion to check if Tree Location is NotNullLocation:

CREATE ASSERTION NonNullLocation

CHECK (

(SELECT COUNT(\*) FROM Tree WHERE Location IS NOT NULL) = (SELECT COUNT(\*) FROM Tree));