**DATABASE MANAGEMENT SYSTEM-1**

1. **Project Part-1**
2. **Project Partners:**

* Name- Geetanjali Kulkarni (Access ID- hk2745)
* Name- Soumyadeep Chatterjee (Access ID- hq8682)

**Contribution made by each partner-**

* Geetanjali Kulkarni- ER diagram and coding.
* Soumyadeep Chatterjee- ER diagram and coding.

**Number of hours worked together-** Total 17 hours.

We have equally contributed in part1.

1. **URL to github repo-**

<https://github.com/Soumya98-dev/Database-Management-Systems-I.git>

**ER diagram-**

In the ER diagram, there are a total of ten entities we found after reading requirements. Each entity has its own attributes and also, relationships between each entity are shown below.

A diagram of a company

Description automatically generated

**Assumptions and justifications-**

1. Reading the requirements, we figured out that we have to create accounts for client, admin and contractor David Smith. Also, we found multiple entity creation for tree cutting service. We have created tables for all the entities which we have shared below. Also, we have created .java, DAO.java, control servlet, and .jsp files to make the requirements fulfil.
2. Here, a contractor David Smith takes contracts of tree cutting service. He gives estimated quotes for tree cutting services requested by clients. Clients can accept, reject or negotiate quotes depending on the quotes they have received.
3. Here, clients can login to the web page by adding their username and password. Also, a contractor and admin can login to their own accounts.

**A set of create table statements-**

CREATE DATABASE Project1;

USE Project1;

# Stores information about registered clients. ClientID is generated as a unique identifier

CREATE TABLE Client

(

Client\_ID VARCHAR(10) PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Address VARCHAR(200),

CreditCardInfo VARCHAR(10),

PhoneNumber VARCHAR(10),

Email VARCHAR(50)

);

ALTER TABLE Client

ADD COLUMN Password VARCHAR(100) AFTER Client\_ID;

# Stores information about tree-cutting requests submitted by clients. The Status field indicates whether the request is pending, accepted, or rejected

CREATE TABLE TreeCuttingRequest

(

RequestID VARCHAR(10) PRIMARY KEY,

Client\_ID VARCHAR(10) REFERENCES Client (Client\_ID),

RequestDate DATE,

Status VARCHAR(100),

Note VARCHAR(200)

);

# Stores details about each tree in a request, linked to the corresponding request.

CREATE TABLE TreeInfo

(

TreeID VARCHAR(10) PRIMARY KEY,

RequestID VARCHAR(10) REFERENCES TreeCuttingRequest (RequestID),

Size VARCHAR(10),

Height DECIMAL(2, 1),

Location VARCHAR(100),

ProximityToHouse FLOAT

);

# Stores quotes provided by David Smith in response to client requests. The Quote table is linked to the TreeCuttingRequest table

CREATE TABLE Quote

(

QuoteID VARCHAR(10) PRIMARY KEY,

RequestID VARCHAR(10) REFERENCES TreeCuttingRequest (RequestID),

Price DECIMAL(2, 2),

WorkPeriodFrom DATE,

WorkPeriodTo Date,

ResponseDate Date,

Note VARCHAR(200)

);

# Stores information about orders of work created when a client accepts a quote. It links to the original request and the associated quote

CREATE TABLE OrderOfWork

(

OrderID VARCHAR(10) PRIMARY KEY,

RequestID VARCHAR(10) REFERENCES TreeCuttingRequest (RequestID),

QuoteID VARCHAR(10) REFERENCES Quote (QuoteID),

StartDate DATE,

EndDate DATE,

Status VARCHAR(50)

);

# Stores bills generated based on completed work. It links to the order of work and indicates the status of the bill (e.g., pending, in dispute)

CREATE TABLE Bill

(

BillID VARCHAR(10) PRIMARY KEY,

OrderID VARCHAR(10) REFERENCES OrderOfWork (OrderID),

Amount FLOAT,

DueDate DATE,

Status VARCHAR(50)

);

# Stores responses from clients to quotes. It is linked to the Quote table and the Client table

CREATE TABLE ResponseToQuote

(

ResponseToQuoteID VARCHAR(10) PRIMARY KEY,

QuoteID VARCHAR(10) REFERENCES Quote (QuoteID),

Client\_ID VARCHAR(10) REFERENCES Client (Client\_ID),

ResponseDate DATE,

Note VARCHAR(50)

);

# Stores responses from clients to bills. It is linked to the Bill table and the Client table

CREATE TABLE ResponseToBill

(

ResponseToBillID VARCHAR(10) PRIMARY KEY,

BillID VARCHAR(10) REFERENCES Bill (BillID),

Client\_ID VARCHAR(10) REFERENCES Client (Client\_ID),

ResponseDate DATE,

Note VARCHAR(50)

);

# Stores login credentials for administrators with access to the system

CREATE TABLE Administrator

(

AdminID VARCHAR(10) PRIMARY KEY,

Username VARCHAR(50),

Password VARCHAR(50)

);

ALTER TABLE Administrator

ADD COLUMN Email VARCHAR(50) AFTER AdminID;

# Stores records of revenue generated for specific time periods

CREATE TABLE Revenue

(

RevenueID VARCHAR(10) PRIMARY KEY,

Date DATE,

Amount FLOAT

);