



Islamic University of Technology (IUT)

CSE 4308: Database Management Systems Lab

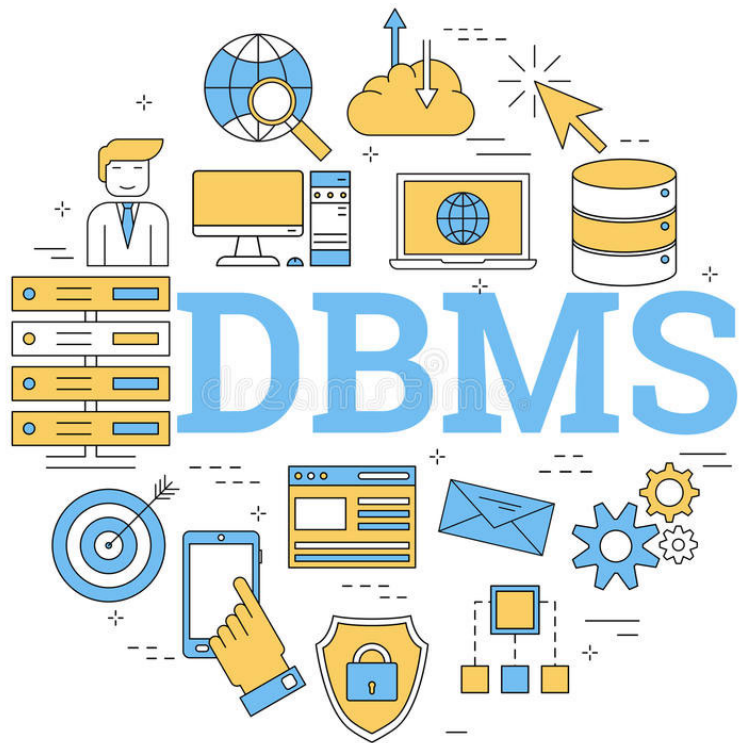
Lab Report # 6

Submitted to:

Md. Bakhtiar Hasan,
Asst. Professor, CSE.
Zannatun Naim Srsity,
Lecturer, CSE.

Submitted by:

M M Nazmul Hossain
ID 200042118
CSE (SWE)



Submission Date:

15.10.2022

Introduction

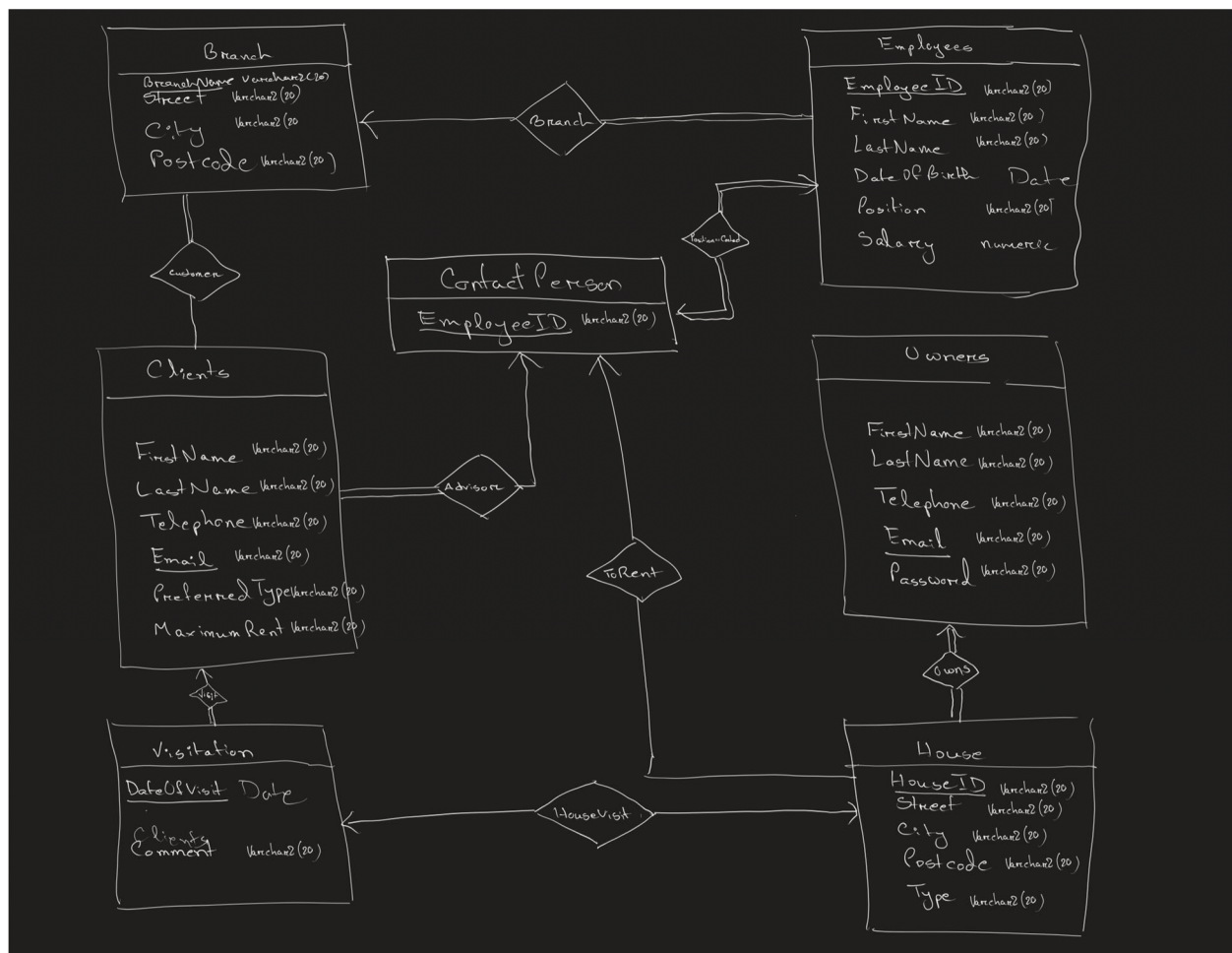
The sixth Database Management Systems Lab was about Entity Relationships. Entity Relationship allows proper organization of Entities and helps avoid data redundancy.

Method

Task 1

The lab task provided a scenario where the relation between many groups was described. Entity Relationships from those had to be identified and a complete Entity Relationship Diagram had to be drawn which didn't have any data redundancy.

To implement that the following Entity Relationship Diagram was drawn:



Task 2

The following task was to covert the previously implemented Entity Relationship Diagram and Define the Schema through Database Definition Language. The following working code is the submitted solution:

```
drop table Branch cascade constraints;
drop table Employee cascade constraints;
drop table ContactPerson cascade constraints;
drop table Clients cascade constraints;
drop table House cascade constraints;
drop table Owner cascade constraints;
drop table Visitation cascade constraints;
drop table Customer cascade constraints;

create table Branch
    (branch_name varchar2(20),
     street varchar2(20),
     city varchar2(20),
     post_code varchar2(20),
     primary key (branch_name)
    );

create table Employee
    (employee_id varchar2(20),
     first_name varchar2(20),
     last_name varchar2(20),
     date_of_birth date,
     position varchar2(20),
     salary numeric(10,0),
     branch_name varchar2(20),
     primary key (employee_id),
     foreign key (branch_name) references Branch
    );

create table ContactPerson
    (employee_id varchar2(20) unique,
     first_name varchar2(20),
     last_name varchar2(20),
     primary key (employee_id),
     foreign key (employee_id) references Employee
    );

create table Clients
    (first_name varchar2(20),
     last_name varchar2(20),
     telephone varchar2(20),
```

```

    email varchar2(20),
    preferred_type varchar2(20),
    max_rent varchar2(20),
    employee_id varchar2(20),
    primary key (email),
    foreign key (employee_id) references ContactPerson
);
create table Customer
    (branch_name varchar2(20),
    email varchar2(20),
    foreign key (email) references Clients,
    foreign key (branch_name) references Branch(branch_name)
);
create table Owner
    (first_name varchar2(20),
    last_name varchar2(20),
    telephone varchar2(20),
    password varchar2(20),
    email varchar2(20),
    primary key (email)
);
create table House
    (house_id varchar2(20),
    street varchar2(20),
    city varchar2(20),
    postcode varchar2(20),
    type varchar2(20),
    email varchar2(20),
    employee_id, varchar2(20),
    primary key (house_id),
    foreign key (email) references Owner(email),
    foreign key (employee_id) references ContactPerson
);
create table Visitation
    (date_of_visit date,
    client_comment varchar2(20),
    client_email varchar2(20),
    house_id varchar2(20) unique,
    primary key (date_of_visit),
    foreign key (client_email) references Clients(email),
    foreign key (house_id) references House(house_id)
);

```

Explanation of Solution

The ER Diagram contains the database schema of Bhalo Basha Chai (BBC). The solution Diagram has 7 tables and 1 junction table.

1. Branch: branch_name is the primary key of this table along with the relevant attributes. It has a many-to-many relation with Clients which is assisted by the junction table Customer. It also has a one-to-many relation with Employees. Since it was the first relation, it didn't have any foreign keys.
 - a. Primary Key: branch_name
 - b. Foreign Key: (none)
 - c. Relationships:
 - i. One-to-one: (none)
 - ii. One-to-many: Employee
 - iii. Many-to-many: Clients
2. Employee: employee_id is the primary key of this table along with the relevant attributes. It has Many-to-one relation with the Branch as previously stated. It has a One-to-one relation with ContactPerson, where essentially, the Contact person is an Employee whose position is Contact Person.
 - a. Primary Key: employee_id
 - b. Foreign Key: branch_name from Branch table
 - c. Relationships:
 - i. One-to-one: ContactPerson
 - ii. Many-to-one: Branch
 - iii. Many-to-many: (none)
3. ContactPerson: employee_id is both the primary and foreign key of the ContactPerson table which references the Employees. Contact Person essentially serves to reduce clutter as the scenarios provided involved only Contact person employees in different scenarios.

- a. Primary Key: employee_id
 - b. Foreign Key: employee_id from Employees
 - c. Relationships:
 - i. One-to-one: Employee
 - ii. Many-to-one: Branch, Clients, and Houses
 - iii. Many-to-many: (none)
4. Clients: email attribute of a client is being used as the primary key as it would be unique for all different clients. It had other attributes including the foreign key of employee_id from ContactPerson.
- a. Primary Key: email
 - b. Foreign Key: employee_id from ContactPerson
 - c. Relationships:
 - i. One-to-one: (none)
 - ii. Many-to-one: ContactPerson
 - iii. One-to-Many: Visitation
 - iv. Many-to-many: Branch
5. Customer: It is essentially the junction table designed to build the Many-to-Many relation between the Branch and Clients table. It has only two attributes, both of which are foreign keys referencing both the Branch and Clients table.
- a. Primary Key: (none)
 - b. Foreign Key:
 - i. branch_name from Branch table
 - ii. email from Clients
6. Owner: email is the primary key of this table along with the relevant attributes. It has a One-to-Many relation with the Houses.
- a. Primary Key: email
 - b. Foreign Key: (none)
 - c. Relationships:
 - i. One-to-one: (none)

- ii. One-to-many: Houses
 - iii. Many-to-many: (none)
- 7. Houses: house_id is the primary key for this relation. It has all the other necessary attributes. Two of which are foreign keys referencing the Owner and another referencing the ContactPerson.
 - a. Primary Key: house_id
 - b. Foreign Key:
 - i. email from Owner
 - ii. employee_id from ContactPerson
 - c. Relationships:
 - i. One-to-one: (none)
 - ii. Many-to-one: ContactPerson and Owner
 - iii. Many-to-many: (none)
- 8. Visitation: DateOfVisit is the primary key of this relation. It has other attributes, two of which are foreign keys referencing Clients and Houses.
 - a. Primary Key: DateOfVisit
 - b. Foreign Key:
 - i. Clients_email references Clients
 - ii. House_id references Houses
 - c. Relationships:
 - i. One-to-one: Houses
 - ii. Many-to-one: Clients
 - iii. Many-to-many: none

Problems

Figuring out the relationship between the different entities, from just the given scenario was quite challenging. Maintaining both cardinalities, while ensuring no data redundancy took place proved to be a difficult task. Many entities would end up being weak entities and I had to introduce new attributes. Without more scenarios and examples, proof-checking the cardinality and relations established

isn't possible within the short amount of time provided. To reduce complications, I had to introduce a new ContactPerson table to establish relations with other entities which might help reduce data redundancy.

Findings

To reduce data redundancy, figuring out the correct cardinality and schema of the database is imperative. For DDL, a properly implemented ER Diagram is also a must since it allows for the sequential declaration of the entities alongside establishing their relations.

Conclusion

The lab was a fruitful experience, which will help develop our understanding of Entity Relations.