

Database Management Systems Lab  
Lab 6  
CSE 4308

**A Z Hasnain Kabir**  
200042102  
Software Engineering

19 October 2022

## **Introduction**

An entity relationship diagram helps enterprises and developers document databases, spot inefficiencies, inconsistencies and redundancies in database. Through this process developers can identify design or logic flaws before they are implemented in production.

## **Task 1**

We are to draw an ER diagram for a given problem statement without data redundancy, specifying the cardinality explicitly.

Bhalo Basha Chai (BBC) is a housing agent in Bangladesh that publishes advertisements of properties that can be rented. Previously they stored all their information on paper. Recently they have decided to use a database. They have come up with the following requirements:

- There are many branches of BBC throughout the country. Each branch is located in a street of a city and has a postcode.
- Every branch is maintained by many employees. Upon joining the company, they provide their first name, last name, gender, and date of birth. They are also appointed a position (like manager, salesperson, etc.) in a specific branch. Their salaries are recorded for tax purposes.
- Numerous clients rent houses from BBC. They register by going to a certain branch and providing their first name, last name, telephone number, email, preferred accommodation type, and the maximum amount of rent they can afford. At that time, s/he is also assigned a staff member who is their contact person. A client can register in multiple branches.
- BBC stores information about the property owners who actually own the houses. The owners register by providing their first name, last name, telephone number, email, and password.
- BBC has multiple houses for rent under them. These houses are denoted by street, city, post-code, type, number of available rooms, and rent. Each property is associated with one owner, one contact person who is also a staff member, and the branch the staff works in.
- Each client can visit properties multiple times, but not twice in a day. A client can make some comments about the property during their visit. The date of their visit is also documented.

Figure 1: Problem statement

## Analysis of the problem

After initial reading, we come to the understanding that we are to create Branch table, Employee table, House table, Owner table and Client table. The attributes required for each table are mentioned in the lab task.

**The entity relationship diagram is added later as Figure 2**

## Explanation of the solution

An extra table called Employee position is required. Extra attributes such as E\_ID is added to Employee table. Employee position has a many to one relationship with Employee which we identify with a "is a" diamond. Branch table has an additional Primary key attribute B\_ID. Employee table has a one to many relationship with Branch. Client table has an additional primary key attribute called C\_ID. Client table has a one to many relationship with Employee table. House table has an additional primary key attribute called H\_ID. House table has a one to many relationship with Employee table denoted by "is connected by". House table also has a many to many relationship with Client table. Owner table has an additional primary key attribute called O\_ID. Owner table has a many to one relationship with House

table.

## **Findings**

Entity relationship diagram not only identifies logical design of a database, but also in some cases, ERD identifies the logical constraints of a business model. For example, a client cannot visit properties twice in one day. This is a constraint for the business model but we can accurately specify this using Entity relationship diagram.

**The entity relationship diagram is added later as Figure 2**

## Task 2

We are to write DDL statements for ER diagram.

### Analysis of the problem

We need six main tables and considering many to many relationships we also need three junction tables. We need to add necessary attributes to our table.

### Code

```
CREATE TABLE branch(  
    B_ID INT primary key,  
    Street varchar2(20),  
    city varchar2(20),  
    postcode number(4)  
);  
  
CREATE TABLE employee_Position(  
    position varchar2(20) primary key,  
    salary INT  
);  
  
CREATE TABLE employee(  
    E_ID INT primary key,  
    Firstname varchar2(20),  
    Lastname varchar2(20),  
    Gender varchar2(4),  
    Date_of_birth DATE,  
    branch references branch not null,  
    position references employee_Position not null  
);  
  
CREATE TABLE client(  
    C_ID INT primary key,  
    Firstname varchar2(20),  
    Lastname varchar2(20),  
    Telephone_no varchar2(12),  
    Email varchar2(30),  
    preferred_type varchar2(10),  
    max_rent INT,
```

```

        Staff references employee not null
    );

CREATE TABLE client_registration(
    C_ID references client,
    branch references branch,
    primary key (C_ID, branch)
);

CREATE TABLE House(
    H_ID INT primary key,
    Street varchar2(20),
    city varchar2(20),
    postcode number(4),
    type varchar2(20),
    No_of_rooms number,
    rent number,
    staff references Employee not null
);

CREATE TABLE visits(
    C_ID references client,
    H_ID references House,
    date_of_visit Date not null,
    comments varchar2(100),
    primary key(C_ID, H_ID, date_of_visit)
);

CREATE TABLE owner(
    O_ID INT primary key,
    Firstname varchar2(20),
    Lastname varchar2(20),
    Telephone_no varchar2(12),
    Email varchar2(30),
    Password varchar2(100)
);

ALTER TABLE House ADD owner references owner;

```

## **Explanation of the solution**

We create our tables using normal SQL Data definition commands where the constraints are identified accordingly. Junction tables do not contain one primary key. These tables are dependent on multiple attributes as primary key. The date of client visit and comments are added to the junction table connecting Client and House.

## **Findings**

We had to create the tables that reference each other differently as constraints cannot be instantly added. We had to create House table first, then we had to create Owner table which references House table then we had to alter House table and add constraint owner which references owner.

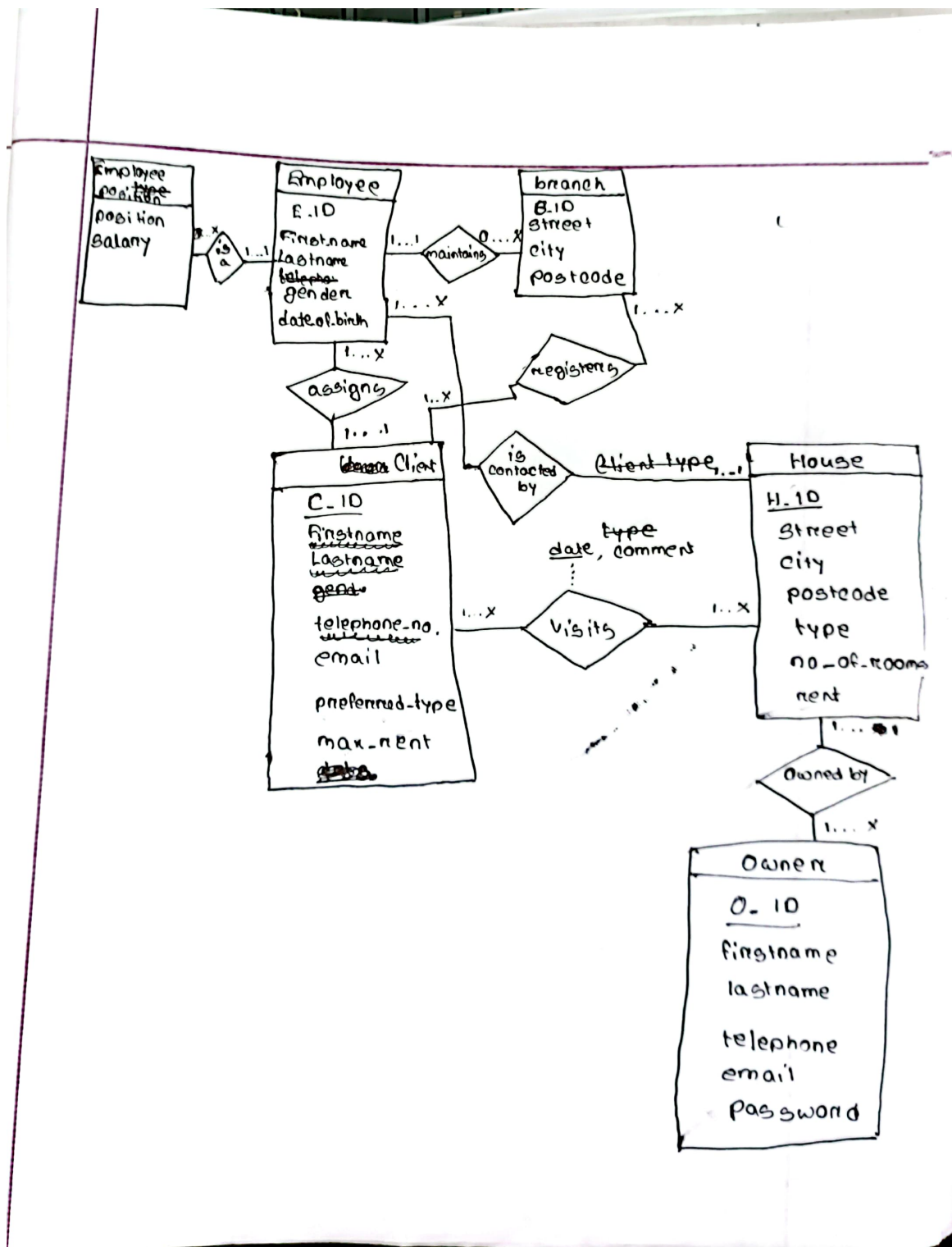


Figure 2: Entity relationship diagram