DTSA 5733: Relational Database Design

Professor Di Wu

Spring Semester 2024

by: Rodrigo Tosaki

Table of Contents

Step 1: Describing the Business

My friend's mother is the owner of a home cleaning business, called Laura's Cleaning.
The operation runs as follows:

- My friend's mother, Laura, is the owner of the business
- She has 15 full-time employees to clean houses
- She has approximately 90 clients which employ her for one or more houses
- Staff members are responsible for providing their own transportation
- Houses are charged an arbitrary rate that remains constant unless an additional request is made. (Which increases cost)
- Available schedule times are Monday-Friday 8am-4pm
- Once a house is scheduled it remains at that time slot permanently

Step 2: Assumptions & ER Model

The assumptions that I have made are as follows:

- A client can own many houses
- Once a house is scheduled at a specific weekday and time it remains that way permanently
- Once a house is set a certain dollar rate per sq ft. it stays that way permanently
- Laura, Jenny, and Mindy are always available to work
- For organizational purposes, each member of staff, each client, each house, and each cleaning will have a designated ID# based on their date introduced
- A house may take anywhere from 1-3 staff members to be cleaned in a timely manner
- Employees are paid an even split depending on the "DollarRate" of the house, so salary is **not** relevant
- Since Laura is the creator of the business, she is the only staff member that does NOT need to be supervised. Any additional/current hires follow the supervising rules detailed in the relationships section of this report
- Fields beginning with "active" are used to track if a client, staff member, or house is still involved with the company
- The relationships below **only** encompass clients, houses, and staff that are **ACTIVE**.

ER Model:

Entities:

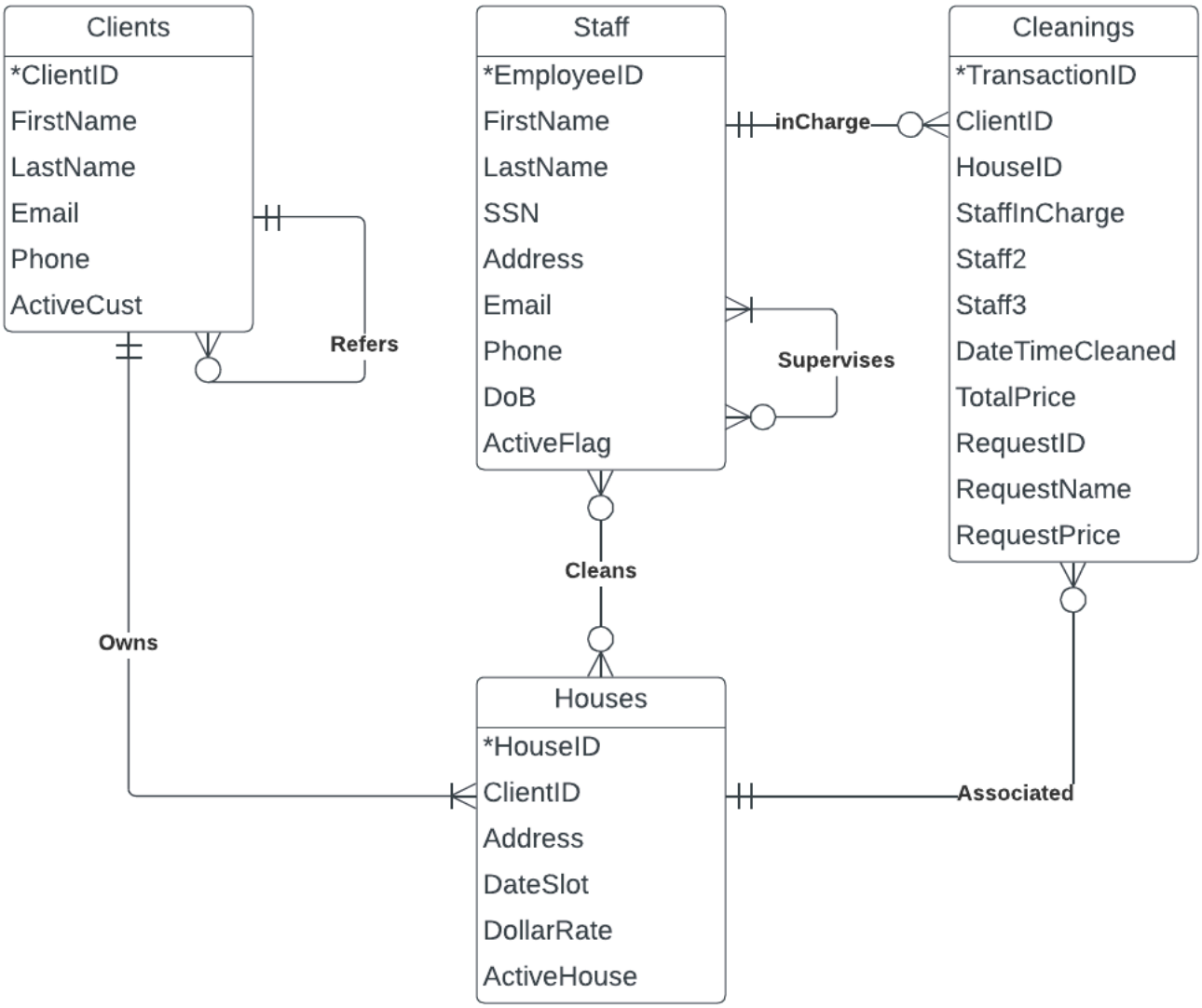
- Staff: EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, ActiveFlag
- Clients: ClientID, FirstName, LastName, Email, Phone, ActiveCust
- Houses: HouseID, ClientID, Address, DateSlot, DollarRate, ActiveHouse
- Cleanings: TransactionID, ClientID, HouseID, StaffInCharge, Staff2, Staff3, DateTimeCleaned, TotalPrice, RequestID, RequestName, RequestPrice

Relationships:

- A client must **own** one or more houses; A house must be owned by one and only one client
- A member of staff may **supervise** one or more members of staff; A member of staff must be supervised by one or more members of staff
- A member of staff may be **in charge** of one or more cleanings; A cleaning must have only one member of staff in charge
- A member of staff may **clean** one or more houses; a house may be cleaned by one or more members of staff
- A referring client may **refer** one or more clients; a referred client must be referred by one and only one client
- A house may be **associated** with one or more cleanings; a cleaning must be associated with one and only one house

Step 3: ER Diagram

ER Diagram:



Step 4: Relational Model

Relational Model:

- Staff(EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, ActiveFlag, SupervisorID(fk))
- Clients(ClientID, FirstName, LastName, Email, Phone, ActiveCust, ReferredBy(fk))
- Houses: (HouseID, ClientID(fk), Address, DateSlot, DollarRate, ActiveHouse)
- Cleanings: (TransactionID, ClientID(fk), HouseID(fk), DateTimeCleaned, StaffInCharge(fk), Staff2(fk), Staff3(fk), TotalPrice, RequestID, RequestName, RequestPrice)

When creating this relational model, the first step taken was to select the correct candidate key as the primary key for each relation. In this case, we did not need to consider candidate keys containing greater than one attribute for any relation.

The next action was to identify any foreign keys (fk) within the displayed relations. However, in order to *accurately* and *exhaustively* account for all fk's, we must consider the relationships that we previously defined.

Accuracy Example:

- **Clients:Houses** represents a 1:N relationship meaning that we must place the foreign key "ClientID" under the **Houses** relation to avoid creating redundant client tuples.

Exhaustive Example:

- The **Staff:Staff** relationship represents a 1:1 relationship that is not explicitly defined in the ER Model or the ER Diagram, so without looking at the relationship, that level of information is lost in translation to the relational model. We create a fk in the **Staff** relation called SupervisorID that can be used to identify who is being supervised by who.

Step 5: Normalization to 3NF

The main goal in normalizing relational databases is to minimize data redundancy to store only what we need. This translates to quicker processing times when querying, lower storage costs, and lower maintenance lead times when performing CRUD operations.

Good design minimizes the need for normalization, but for the sake of the exercise we have included a scenario which breaks one of the normal form rules.

The first step in normalizing the relational model to 3NF is identifying all functional dependencies within each relation.

Staff

1. EmployeeID → FirstName, LastName, SSN, Address, Email, Phone, DoB, ActiveFlag, SupervisorID

Clients

1. ClientID → FirstName, LastName, Email, Phone, ActiveCust, ReferredBy

Houses

1. HouseID → ClientID, Address, DateSlot, DollarRate, ActiveHouse

Cleanings

1. TransactionID → ClientID, HouseID, DateTimeCleaned, StaffInCharge, Staff2, Staff3, TotalPrice, RequestID, RequestName, RequestPrice
2. RequestID → RequestName, RequestPrice

After identifying all functional dependencies, let us identify which, if any, normal form each relation satisfies.

- **Staff** is in 3NF because:
    a. It has only one value in each attribute and is a relation, satisfying **1NF** requirements
    b. It has no partial dependencies, satisfying **2NF** requirements
    c. It has no transitive dependencies, satisfying **3NF** requirements
- **Clients** is in 3NF because:
    a. It has only one value in each attribute and is a relation, satisfying **1NF** requirements
    b. It has no partial dependencies, satisfying **2NF** requirements
    c. It has no transitive dependencies, satisfying **3NF** requirements
- **Houses** is in 3NF because:
    a. It has only one value in each attribute and is a relation, satisfying **1NF** requirements
    b. It has no partial dependencies, satisfying **2NF** requirements
    c. It has no transitive dependencies, satisfying **3NF** requirements
- **Cleanings** is in 3NF because:

a. It has only one value in each attribute and is a relation, satisfying **1NF** requirements
b. It has no partial dependencies, satisfying **2NF** requirements
c. It DOES, however, have transitive dependencies, break the required conditions for **3NF** to be achieved.

We can see that **Cleanings** is the only relation in which normalization is required. The transitive dependencies determined by the attribute "RequestID" should really be stored in their own separate relation so that the information only needs to be recorded once. *(especially varchar values that take up more space such as "RequestName")*

Let us begin the normalization process for the **Cleanings** relation from 2NF to 3NF:

*Current State (2NF) of **Cleanings:***

- Staff(EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, ActiveFlag, SupervisorID(fk))
- Clients(ClientID, FirstName, LastName, Email, Phone, ActiveCust, ReferredBy(fk))
- Houses: (HouseID, ClientID(fk), Address, DateSlot, DollarRate, ActiveHouse)
- Cleanings: (TransactionID, ClientID(fk), HouseID(fk), DateTimeCleaned, StaffInCharge(fk), Staff2(fk), Staff3(fk), TotalPrice, RequestID, RequestName, RequestPrice)

1. First, let us remove any transitive dependencies from **Cleanings.** This involves removing RequestName and RequestPrice from the **Cleanings** relation to a new relation that we will call **SpecialRequests**
2. Now, we must label RequestID as a foreign key (fk) in **Cleanings** and add it to **SpecialRequests** as a primary key (pk) in our relational model.
3. With that we are done normalizing and our entire relational model is now implementable for use with a DBMS!

Step 6: Finalize the Relational Model for Implementation

3rd Normal Form (3NF) State:

- Staff(EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, ActiveFlag, SupervisorID(fk))
- Clients(ClientID, FirstName, LastName, Email, Phone, ActiveCust, ReferredBy(fk))
- Houses: (HouseID, ClientID(fk), Address, DateSlot, DollarRate, ActiveHouse)
- Cleanings: (TransactionID, ClientID(fk), HouseID(fk), DateTimeCleaned, StaffInCharge(fk), Staff2(fk), Staff3(fk), TotalPrice, RequestID(fk))
- SpecialRequests:(RequestID, RequestName, RequestPrice)

Bonus: Finalized ERD