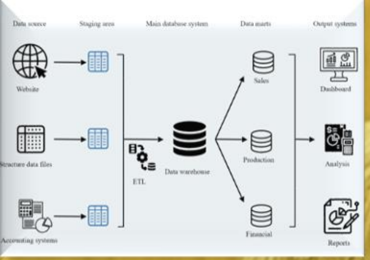


University of Colorado Boulder

Data and Database Architecture

for "*PetsCare Clinic*"

JAN. 2/2023

KEDIR NASIR OMER

Content

- ✓ Project Background
- ✓ Asking Business Questions
- ✓ The Entity Relationship Model (ERM)
- ✓ The Entity Relationship Diagram (ERD)
- ✓ Convert the ERD to the RM
- ✓ Normalize the RM to the Third Normal Form (3NF)
- ✓ Final Output for Implementation
- ✓ Reference
- ✓ Acknowledgment

Project Background

➢ Problem Statement

- ❑ PetsCare is a local clinic that offers care for animals, such as vaccinations, medical services, spay and neuter surgery, dental cleaning and treatment, grooming, and other services. Over the course of more than 20 years, PetsCare has successfully operated as a business while serving the community.

- ❑ The current owner, Claire, who is also the company's founder's daughter, recently came to the conclusion that the outdated file system solution is incredibly ineffective. She and her team are required to put in a lot of time collecting records, requesting data, and maintaining the data.

- ❑ She asked me to design an RDBMS, so that it could keep track of customers, staff, and visitors to the PetsCare clinic.

➢ How to Design a Reliable RDBMS?

- ❑ The main principles for designing a database
  - ✓ Avoid redundant data to conserve space and to keep your data accurate and consistent
  - ✓ Information accuracy and completeness are important
- ❑ Therefore, a good database design is one that
  - ✓ Organize the data into subject-based tables to eliminate redundant information
  - ✓ gives access the data it needs so that it can join the data in the tables as necessary
  - ✓ supports and helps to ensure the integrity and accuracy of your information
  - ✓ meets the needs for data processing and reporting

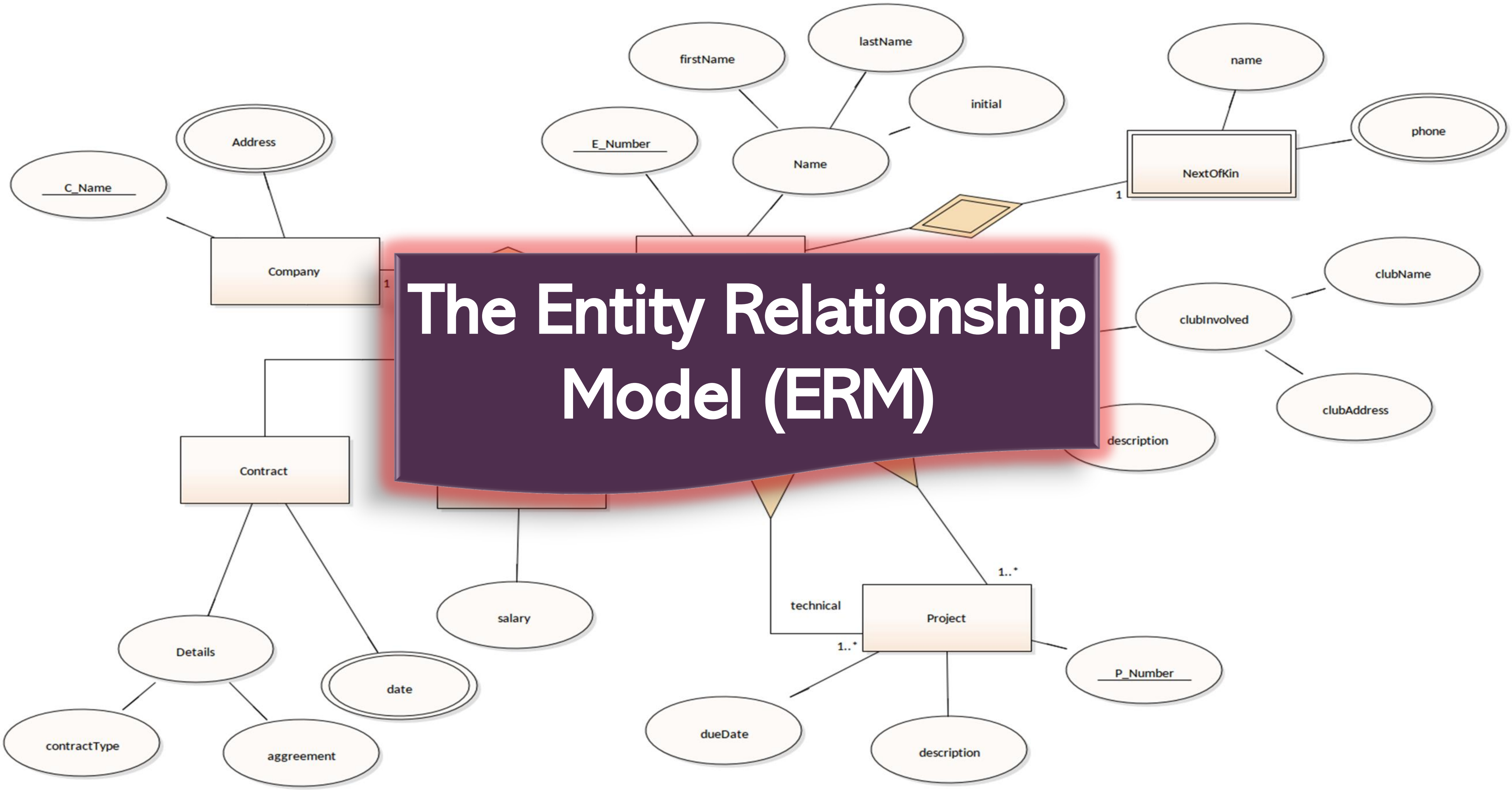
Asking Business Questions?

➢ Two Categories Of Questions To Be Asked

- ❑ Running a business involves a complex set of considerations, including the business's interests, the information that needs to be kept, the potential holding entities, and the identity possibilities. By taking all of these factors into account, a business can make informed decisions that will help it to succeed in the long term.

- ❑ In order to create an Entity Relationship Model and resolve ambiguities, it is important to consider the connections between the parties involved, such as which are required to participate, which are optional, and what the cardinalities are. These connections are essential for the successful creation of the Entity Relationship Model.

The Entity Relationship Model (ERM)



➢ The Entities Summary

- ❑ I confirmed the following entities after several rounds of discussion with my client.

  - ✓ **Customers**: CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate. CustomerID is the identifier.

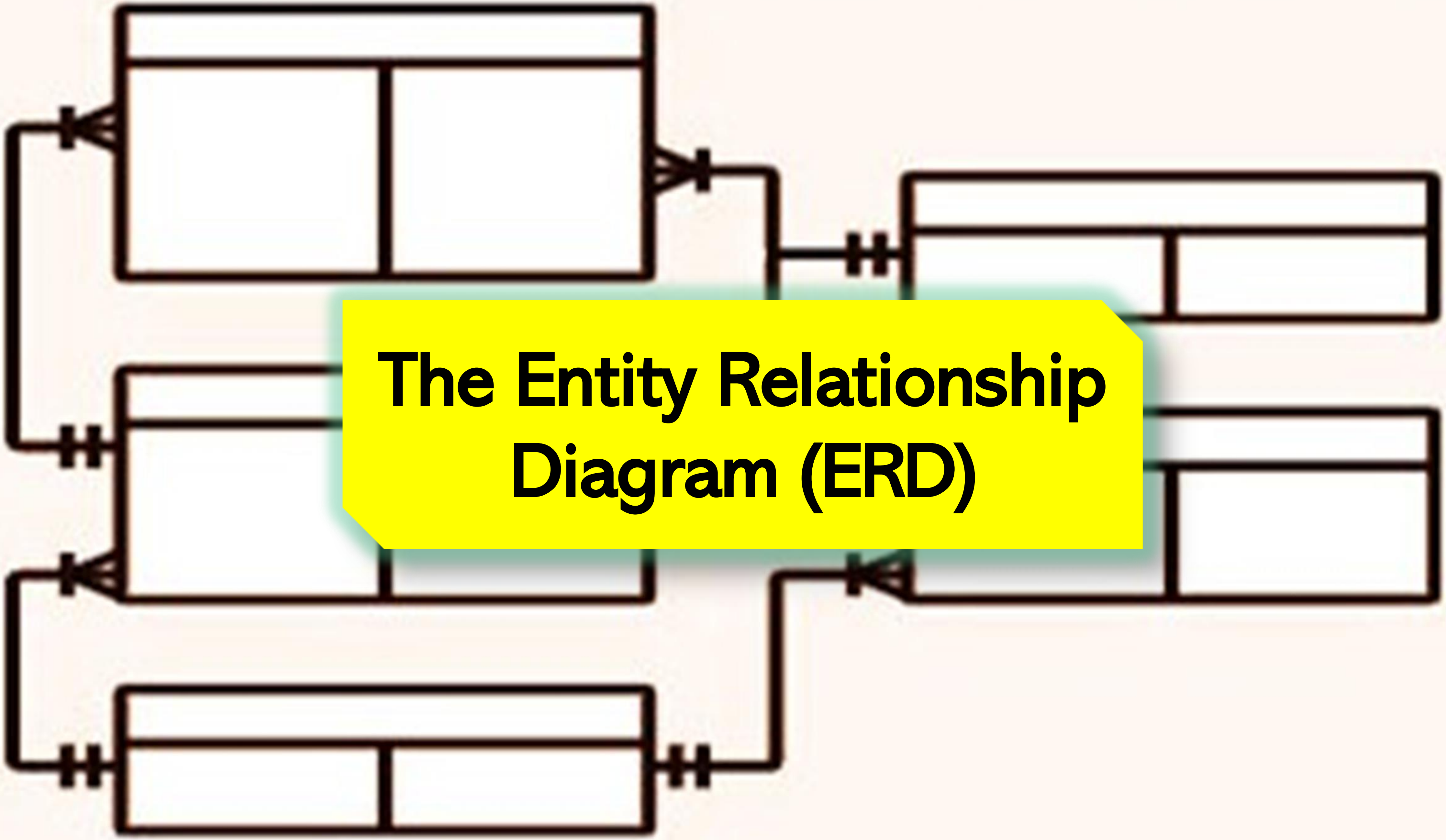
  - ✓ **Pets**: CustomerID, Pet#, NickName, Address, Email, Phone, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes. CustomerID and Pet# together as the identifier.

  - ✓ **Staff**: EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB. EmployeeID is the identifier.

  - ✓ **Visit**: VisitID, Date, Time, Customer, Pet, ServiceID, ServiceName, ServicePrice, ServiceDescription, Staff, Bill, Paid. VisitID is the identifier.

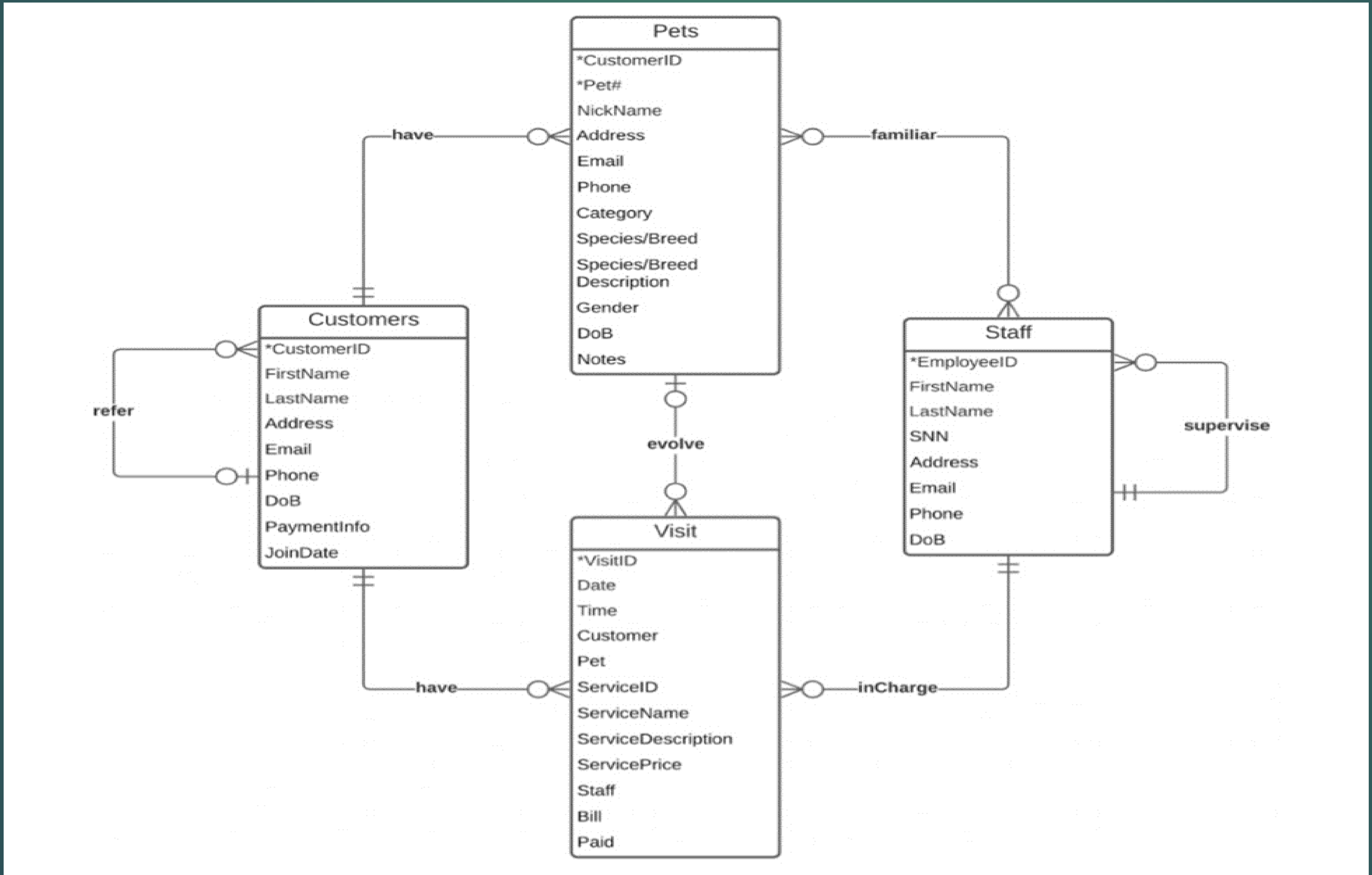
Entities And Their Relationships

- A customer may have one or more (zero or more) pets; A pet must belong to one and only one (exactly one) customer
- A staff may treat one or more pets (zero or more); and A pet may be familiar with one or more (zero or more) staff
- A customer may have one or more (zero or more) visits; and each visit must be done by one and only one (exactly one) customer
- A pet may have one or more (zero or more) visits; and each visit may involve one (zero or one) pet
- A staff must be in charge of one or more (one or more) visits; and a visit must be charged by one and only one (exactly one) staff
- A customer may be referred by one (zero or one) customer; and a customer may refer one or more (one or more) customers
- A staff must have one and only one (exactly one) supervisor; A staff may supervise one or more (zero or more) staff



The Entity Relationship Diagram (ERD)

➢ The ER Illustration

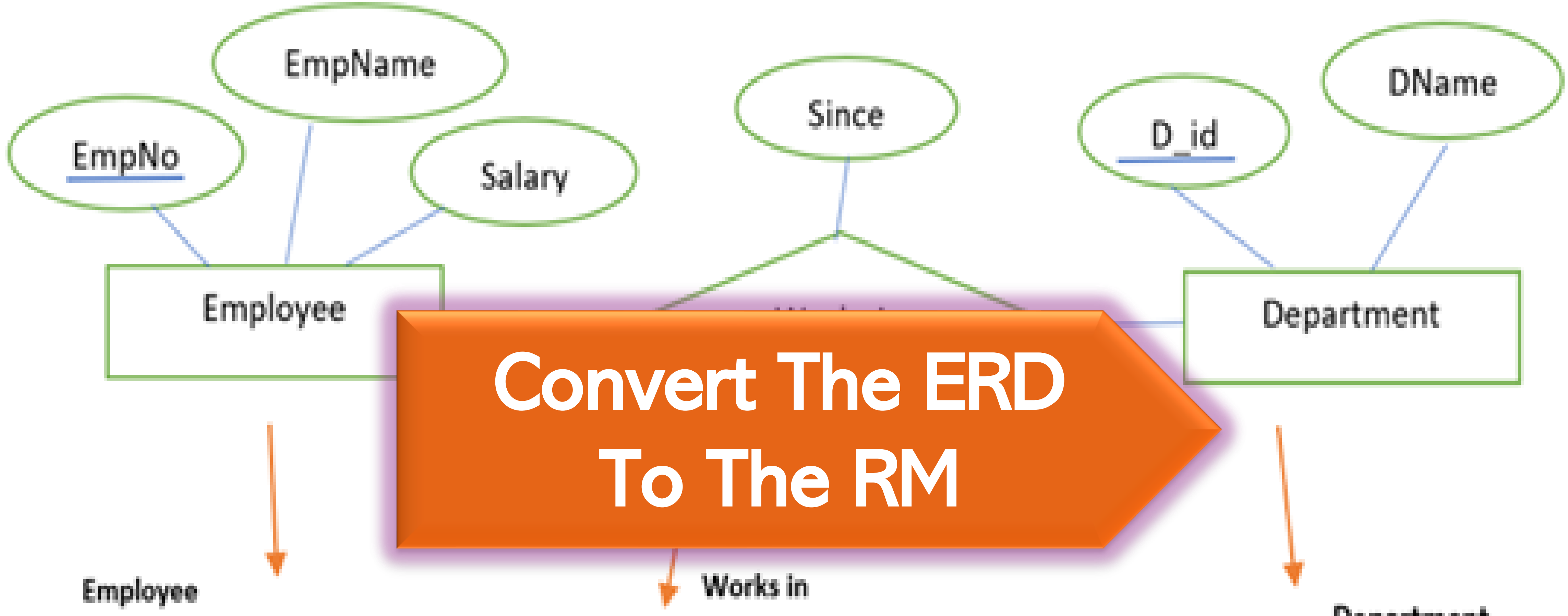


Pets
*CustomerID
*Pet#
NickName
Address
Email
Phone
Category
Species/Breed
Species/Breed Description
Gender
DoB
Notes

Customers
*CustomerID
FirstName
LastName
Address
Email
Phone
DoB
PaymentInfo
JoinDate

Staff
*EmployeeID
FirstName
LastName
SNN
Address
Email
Phone
DoB

Visit
*VisitID
Date
Time
Customer
Pet
ServiceID
ServiceName
ServiceDescription
ServicePrice
Staff
Bill
Paid



Convert The ERD To The RM

Employee

EmpNo	EmpName	Salary

Works in

EmpNo	D_id	Since

Department

D_id	DName

➢ The ERD to RM Conversion

- ❑ Customers (CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate, ReferredByCustomerID(fk)).

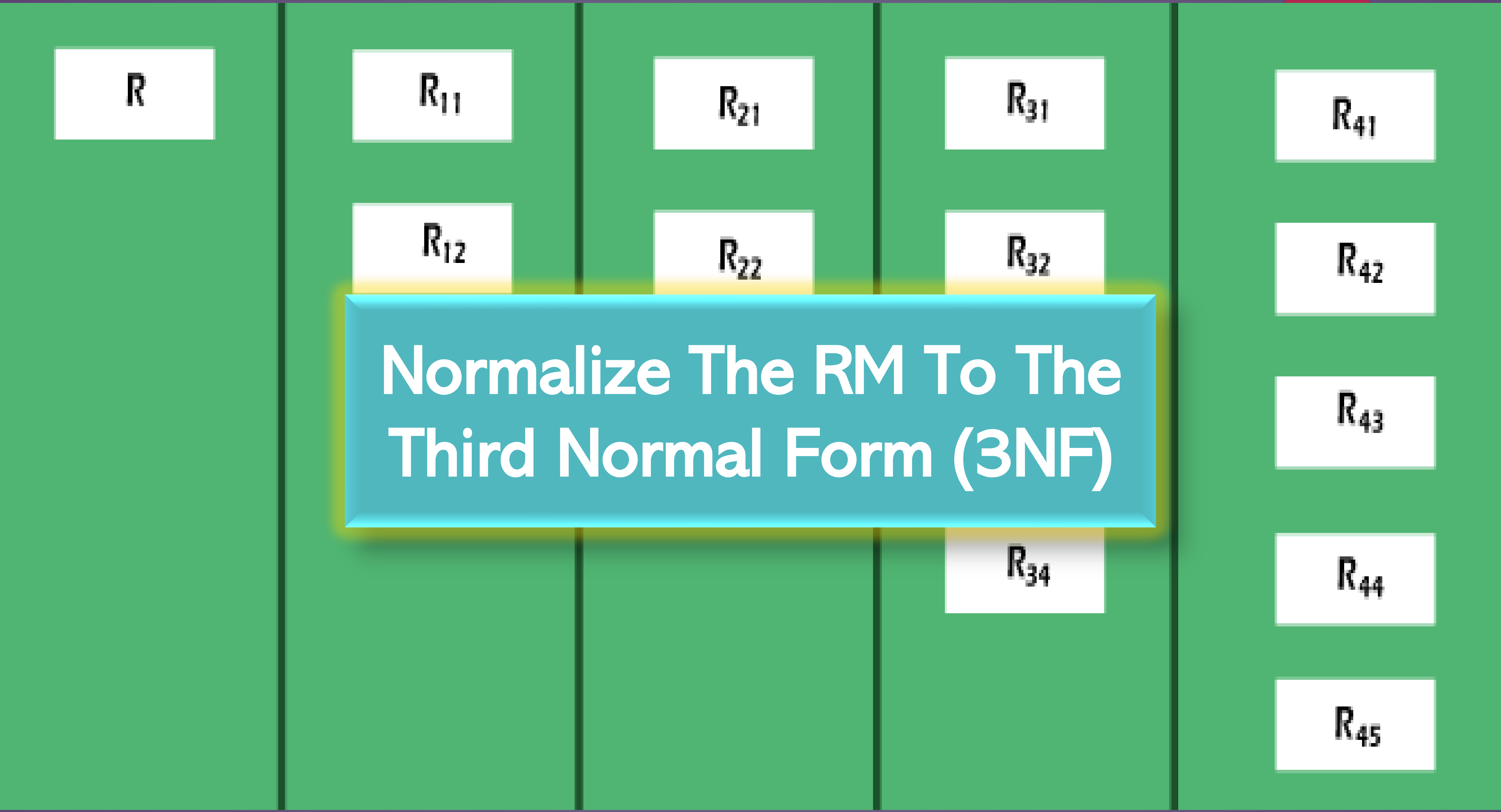
- ❑ Pets (CustomerID(fk), Pet#, NickName, Address, Email, Phone, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes).

- ❑ Staff (EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, SupervisorID(fk)).

- ❑ Visit (VisitID, Date, Time, CustomerID(fk), Pet(fk), ServiceID, ServiceName, ServicePrice, ServiceDescription, EmployeeID(fk), Bill, Paid).

- ❑ Pets_Staff(CustomerID(fk), Pet#(fk),EmployeeID(fk))

Normalize The RM To The Third Normal Form (3NF)



➢ Functional Dependencies

❑ I asked Claire about the Functional Dependencies of the relations before normalization. Here's what she told me:

- ✓ Customers (CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate, ReferredByCustomerID(fk)).
  - FD1: CustomerID → FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate, ReferredByCustomerID
- ✓ Pets (CustomerID(fk), Pet#, NickName, Address, Email, Phone, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes).
  - FD1: CustomerID, Pet# → NickName, Address, Email, Phone, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes.
  - FD2: CustomerID → Address, Email, Phone
  - FD3: Species/Breed → Species/Breed Description

Cont…

- Staff (EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, SupervisorID(fk)).
  - FD1: EmployeeID → FirstName, LastName, SSN, Address, Email, Phone, DoB, SupervisorID
- Visit (VisitID, Date, Time, CustomerID(fk), Pet(fk), ServiceID, ServiceName, ServicePrice, ServiceDescription, EmployeeID(fk), Bill, Paid).
  - FD1: VisitID → Date, Time, CustomerID, Pet, ServiceID, ServiceName, ServicePrice, ServiceDescription, EmployeeID, Bill, Paid
  - FD2: ServiceID → ServiceName, ServicePrice, ServiceDescription.
- Pets_Staff(CustomerID(fk), Pet#(fk),EmployeeID(fk))
  - There is no non-primary-key attribute.

➢ The Normalization Process

- ❑ Customers, Staff, and Pets_Staff relations are in 3NF, because they are in 1NF; they have no partial functional dependencies so they are in 2NF; and they have no transitive functional dependencies so they are in 3NF.

- ❑ Pets relation is in 1NF. However, it is not in 2NF because FD2: CustomerID → Address, Email, Phone. CustomerID as part of the primary key, determines non-primary-key attributes. This leads to a partial functional dependency. We need to normalize Pets to 2NF:
  - Create a new relation to put CustomerID, Address, Email, Phone. Since Customer relation has these attributes, we can simply remove them from Pets, and keep CustomerID as a foreign key.
  - Pets (CustomerID(fk), Pet#, NickName, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes).
    - FD1: CustomerID, Pet# → NickName, Category, Species/Breed, Species/Breed Description, Gender, DoB, Notes.
    - FD2: Species/Breed → Species/Breed Description

➢ Cont…

❑ Pets relation now is in 2NF. However, it is not in 3NF because of FD2: Species/Breed → Species/Breed Description. (CustomerID, Pet#) →Species/Breed, and Species/Breed → Species/Breed Description is a transitive functional dependency. We need to normalize Pets to 3NF:

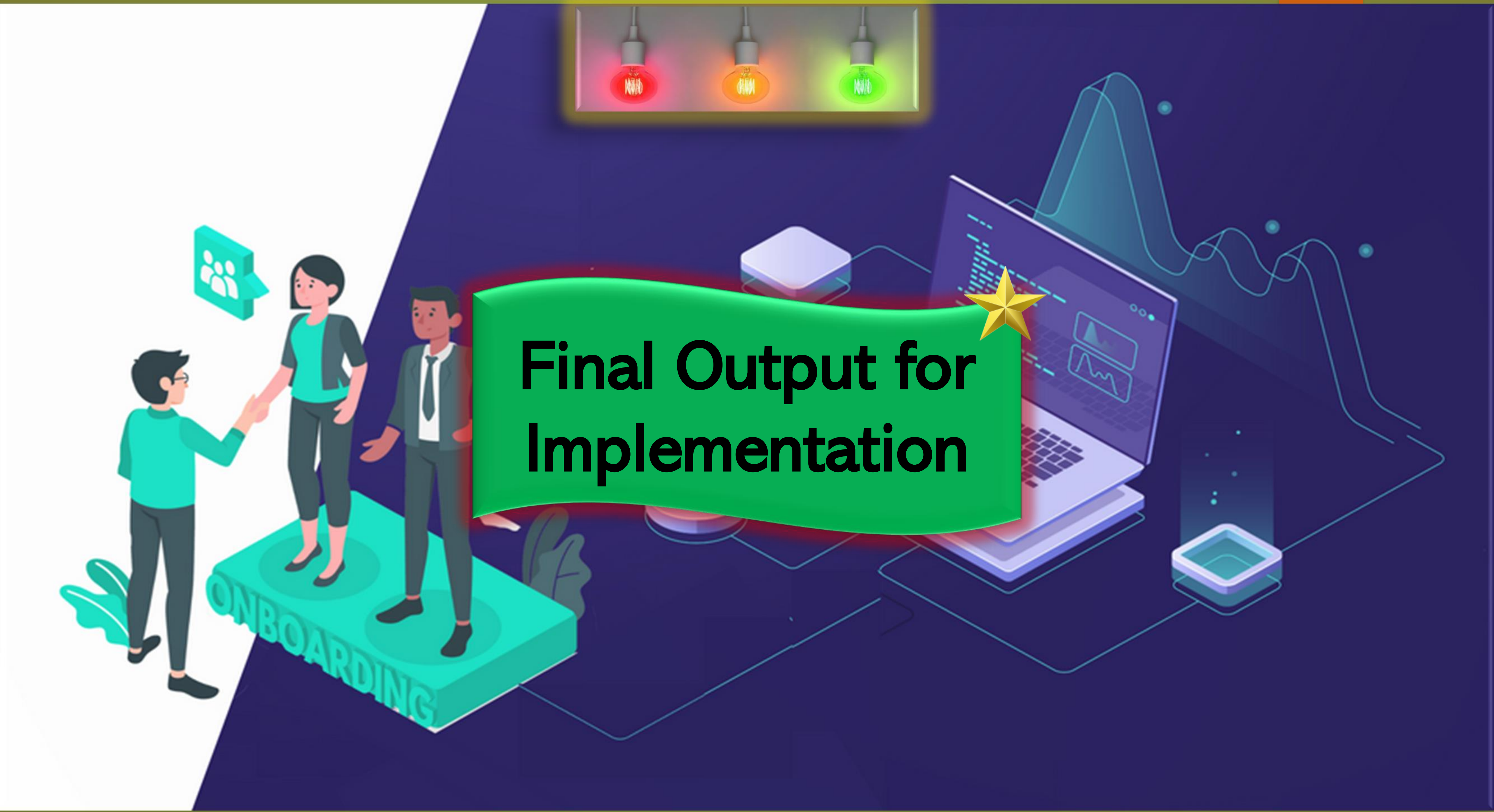
- Create a new relation to put Species/Breed and Species/Breed Description and modify Pets:
- Species (Species/Breed, Species/Breed Description)
  - FD1: Species/Breed → Species/Breed Description
- Pets (CustomerID(fk), Pet#, NickName, Category, Species/Breed(fk), Gender, DoB, Notes).
  - FD1: CustomerID, Pet# → NickName, Category, Species/Breed, Gender, DoB, Notes.

❑ Now Pets relation is in 3NF.

Cont…

- ❑ Visit relation is in 1NF, and 2NF. However, it is not in 3NF because of FD2: ServiceID → ServiceName, ServicePrice, ServiceDescription. VisitID → ServiceID, and ServiceID → ServiceName, ServicePrice, ServiceDescription is a transitive functional dependency. We need to normalize Visit to 3NF:
  - Create a new relation to put ServiceID, ServiceName, ServicePrice, ServiceDescription and modify visit.
  - Service (ServiceID, ServiceName, ServicePrice, ServiceDescription)
    - FD1: ServiceID → ServiceName, ServicePrice, ServiceDescription
  - Visit (VisitID, Date, Time, CustomerID(fk), Pet(fk), ServiceID(fk), EmployeeID(fk), Bill, Paid).
    - FD1: VisitID → Date, Time, CustomerID, Pet, ServiceID, EmployeeID, Bill, Paid

Final Output for Implementation



➢ Final Relational Model In 3NF

❑ Customers (CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate, ReferredByCustomerID(fk)).

- FD1: CustomerID → FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate, ReferredByCustomerID

❑ Pets (CustomerID(fk), Pet#, NickName, Category, Species/Breed(fk), Gender, DoB, Notes).

- FD1: CustomerID, Pet# → NickName, Category, Species/Breed, Gender, DoB, Notes.

❑ Species (Species/Breed, Species/Breed Description)

- FD1: Species/Breed → Species/Breed Description

❑ Staff (EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, SupervisorID(fk)).

- FD1: EmployeeID → FirstName, LastName, SSN, Address, Email, Phone, DoB, SupervisorID

➢ Cont…

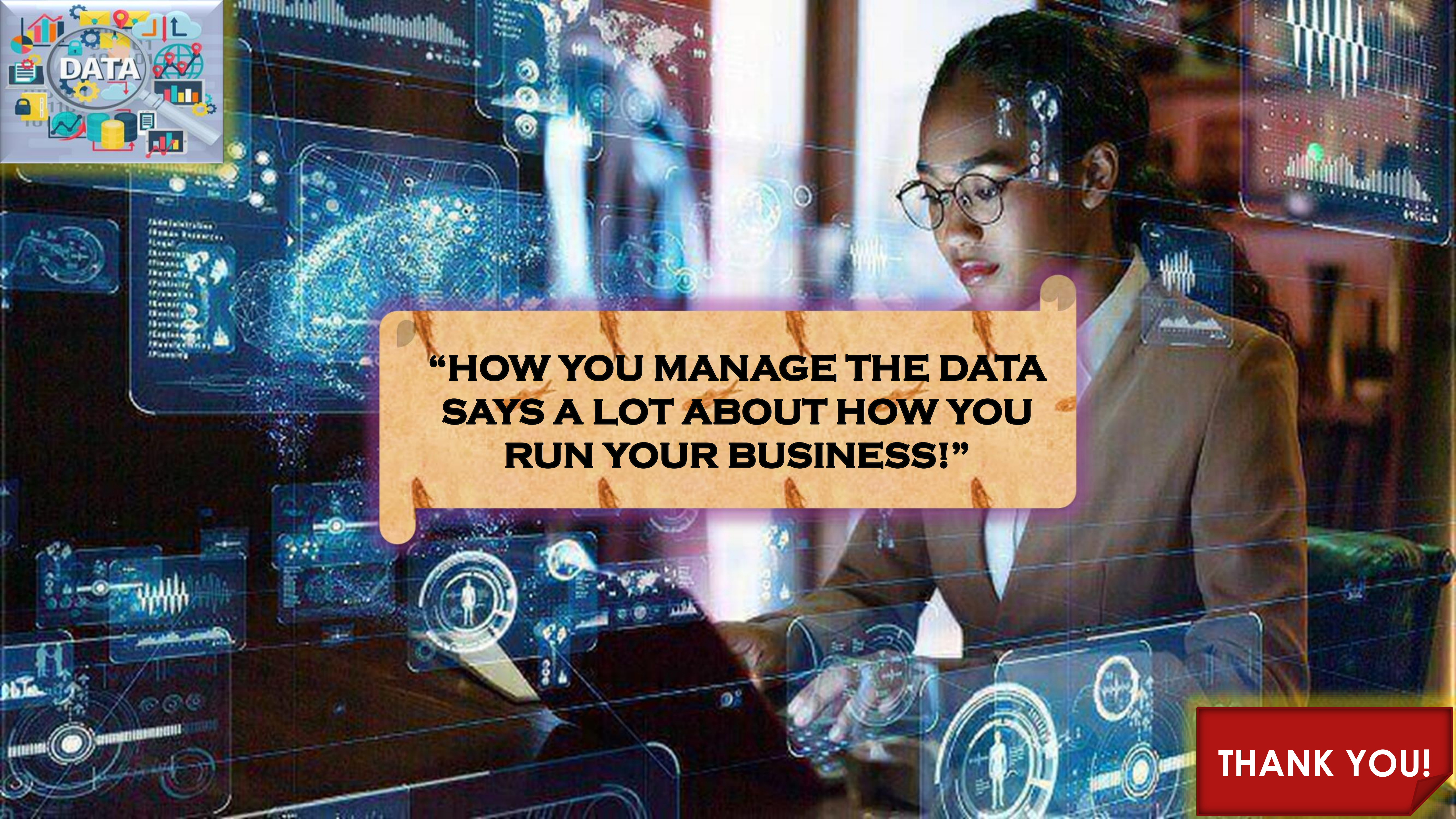
- ❑ Visit (VisitID, Date, Time, CustomerID(fk), Pet(fk), ServiceID(fk), EmployeeID(fk), Bill, Paid).
  - FD1: VisitID → Date, Time, CustomerID, Pet, ServiceID, EmployeeID, Bill, Paid
- ❑ Service (ServiceID, ServiceName, ServicePrice, ServiceDescription
  - FD1: ServiceID → ServiceName, ServicePrice, ServiceDescription
- ❑ Pets_Staff(CustomerID(fk), Pet#(fk),EmployeeID(fk))
  - There is no non-primary-key attribute.
- ❑ I successfully designed a reliable RDBMS and hand it over to the owner Claire, which effectively keep track of customers, staff, and visitors to the PetsCare clinic.

Reference

- ❑ **University Of Colorado Boulder- Databases for Data Scientists Specializations**
  - ✓ Visit My GitHub Portfolio: https://github.com/kedibeki
  - ✓ Visit the Specializations Page: https://www.coursera.org/specializations/databases-for-data-scientists

Acknowledgment

- ❑ **University Of Colorado Boulder and Coursera- Databases for Data Scientists Specializations Instructors**
  - ✓ **Di Wu-** Instructor, Data Science
  - ✓ **Alan Paradise-** Teaching Professor, Computer Science



“HOW YOU MANAGE THE DATA SAYS A LOT ABOUT HOW YOU RUN YOUR BUSINESS!”

THANK YOU!