INFORMATION SYSTEMS AND DATABASES

DESIGNING A RELATIONAL DATABASE FOR A SMALL BUSINESS

Renato Vivar Orellana

Data Science Engineer

PROBLEM SUMMARY

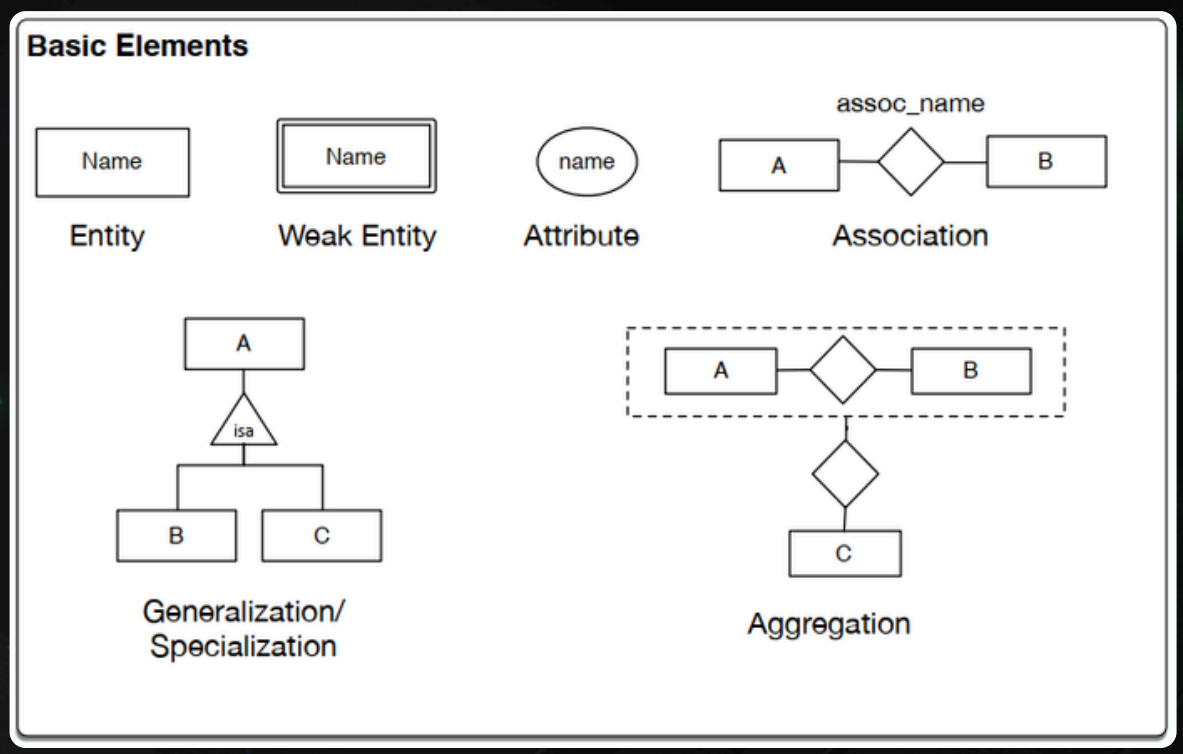
The goal of this project is to design a relational database for a veterinary hospital.

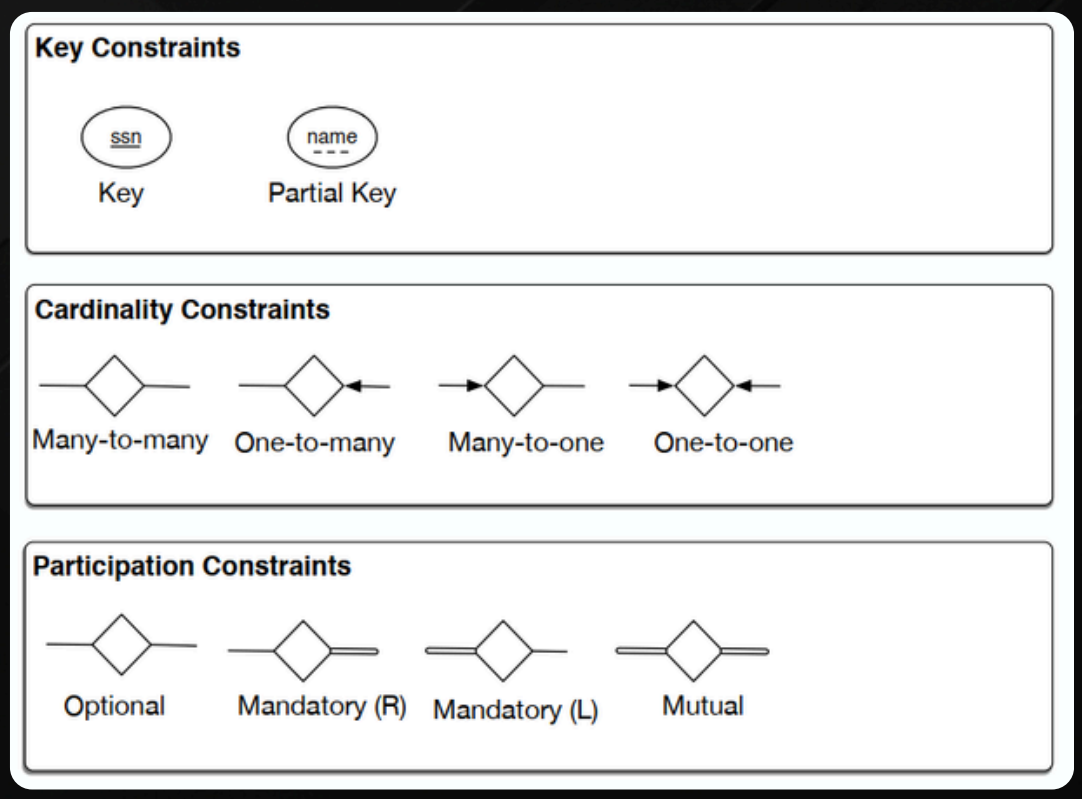
- It must store and organize data related to:
- Doctors (with specialization and biography)
- Clients and their Animals (mammals, birds, reptiles, fish)
- Veterinary Visits (with SOAP notes and diagnostic codes)
- Prescriptions and Medications
- Procedures (tests, radiographies, surgeries, etc.)
- The database supports clinical follow-ups and research by storing structured visit and medical data.

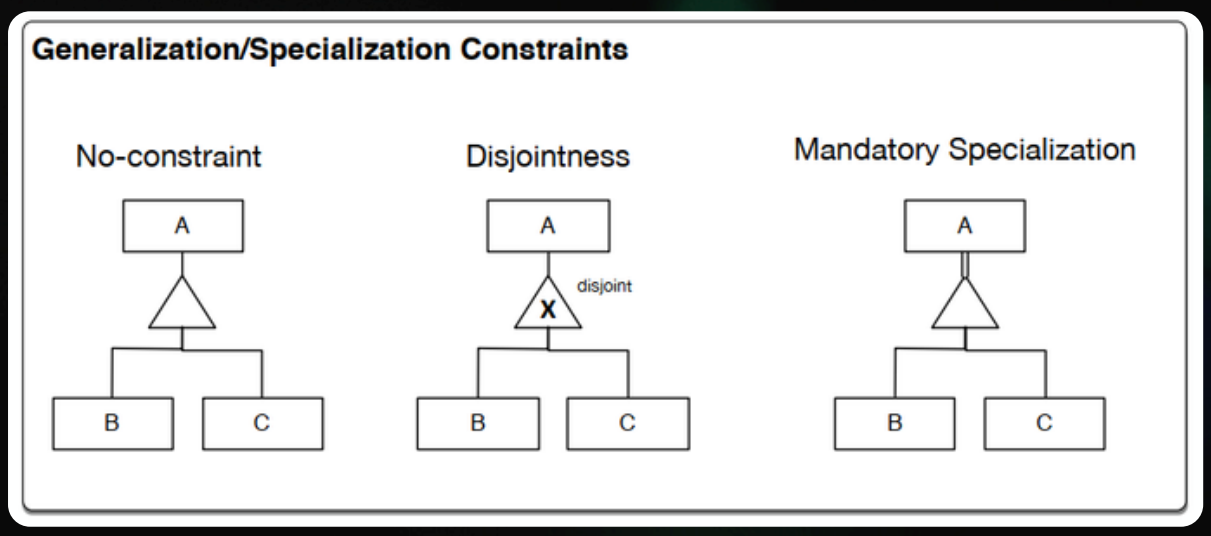
Renato Vivar Orellana
Data Science Engineer

ER MODELING BASICS

ER modeling is used to visually represent the structure of a database. It defines entities, their attributes, relationships, and constraints to ensure data consistency and clarity before implementation.

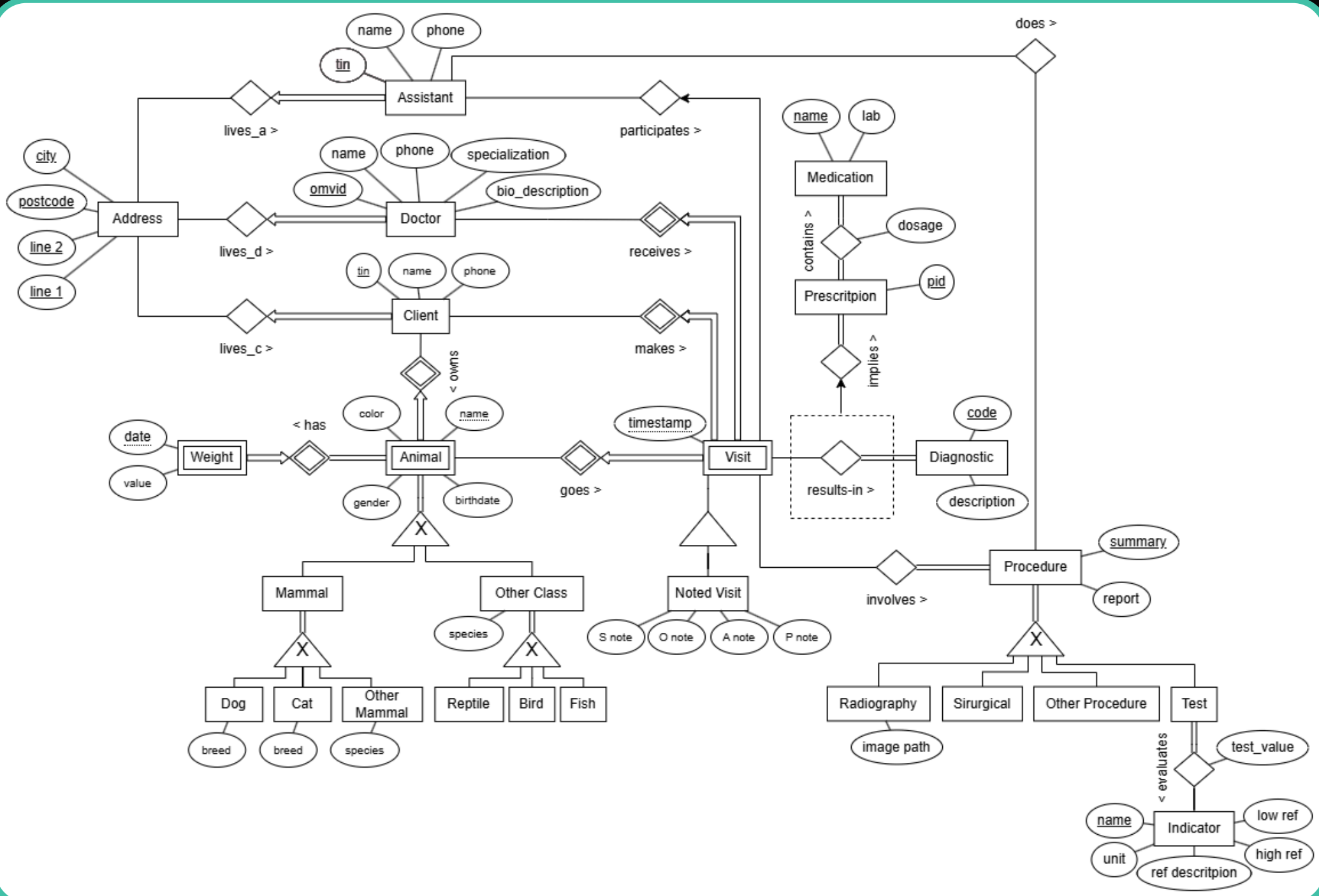






Renato Vivar Orellana
Data Science Engineer

ER DIAGRAM



Renato Vivar Orellana
Data Science Engineer

ER DIAGRAM - KEY POINTS

ER-Model Design Decisions – Key Points

- Address (Strong Entity)
  - All address attributes (line1, line2, postcode, city) form the composite primary key to prevent duplicate entries.
- Assistant (Strong Entity)
  - Uses TIN as the identifier due to lack of other uniquely identifying attributes.
  - TIN is already a reliable identifier for clients, reused here.
- Animal (Weak Entity)
  - Depends on Client (owner) for identity.
  - Allows:
    - Same animal names across different clients.
    - One client owning multiple animals.
- Animal Specialization
  - Superclass: Animal
  - Subclasses: Mammal, Bird, Reptile, Fish
    - Mammals further split into: Dog, Cat, and Other Mammals
  - Breed stored for dogs/cats; species stored for others.
  - Subclasses help capture domain semantics despite shared attributes.
- Weight (Weak Entity)
  - Associated with Animal and includes measurement date.
  - Captures most recent weight, acknowledging changes over time.
- Visit (Weak Entity)
  - Depends on Client, Doctor, and Animal.
  - Identified by a timestamp, supports multiple visits in a day.
  - No need for separate association tables like "goes" or "receives".
  - Optional SOAP notes handled via visit specialization.
- Diagnostic & Prescription
  - Visit may have multiple diagnostics.
  - Prescriptions only exist if diagnostics exist (optional relationship).
  - Each prescription belongs to exactly one Visit-Diagnostic pair.
- Medication
  - Only medications linked to prescriptions are stored.
  - No standalone medication catalog.
- Procedures
  - Identified by unique summary (used as primary key).
  - Avoids using full report text for cost reasons.
  - Specialized into types like tests or radiography.
  - May involve assistants and associated data (e.g., image path, test indicators).

Renato Vivar Orellana
Data Science Engineer

SQL IMPLEMENTATION

EXAMPLE FOR A ENTITY

```sql
CREATE TABLE animal (
    tin NUMERIC(9),
    name VARCHAR(80),
    color VARCHAR(50) NOT NULL,
    gender VARCHAR(20) NOT NULL,
    birthdate DATE NOT NULL,

    PRIMARY KEY (tin, name),
    FOREIGN KEY (tin) REFERENCES client(tin)

    -- Every animal must exist either in the table 'mammal' or in the table 'other_class'.
    -- No animal can exist at the same time in the both the table 'mammal' or in the table 'other_class'.
);
```

This animal table stores basic information about animals owned by clients. Each animal is uniquely identified by the combination of the owner's TIN and the animal's name. The table includes attributes like color, gender, and birthdate. A foreign key constraint ensures that each animal is linked to a valid client.

Additionally, a business rule (commented) enforces that every animal must belong to either the mammal or other_class table—but not both—ensuring proper specialization.

EXAMPLE FOR AN ASSOCIATION

```sql
CREATE TABLE involves (
    summary VARCHAR(255),
    tin NUMERIC(9),
    omvid INTEGER,
    name VARCHAR(80),
    date TIMESTAMP,
    tin_owner NUMERIC(9),

    PRIMARY KEY (tin, omvid, name, date, tin_owner, summary),
    FOREIGN KEY (tin, omvid, name, date, tin_owner) REFERENCES visit(tin, omvid, name, date, tin_owner),
    FOREIGN KEY (summary) REFERENCES procedures(summary)
);
```

The involves table represents the association between a visit and a procedure. Each row links one procedure (by its summary) to a specific visit, which is identified by a composite key: client TIN, doctor ID (omvid), animal name, visit date, and animal owner's TIN. This design ensures that: Every procedure is tied to an existing visit, only procedures recorded in the procedures table can be associated and The primary key includes all attributes to guarantee uniqueness for each visit-procedure link.

Renato Vivar Orellana
Data Science Engineer

SQL IMPLEMENTATION - TESTING

INSERT RANDOM VALUES

```sql
-- Insert a client
INSERT INTO client (tin, name, phone)
VALUES (123456789, 'Maria Gomez', 912345678);

-- Insert an address
INSERT INTO adress (line_1, line_2, postcode, city)
VALUES ('Rua das Flores', 'Apt 3', '1000-123', 'Lisbon');

-- Insert a doctor
INSERT INTO doctor (omvid, name, phone, specialization, bio_description)
VALUES (101, 'Dr. Sousa', 911112222, 'Surgery', 'Experienced in small animal surgery.');
```

QUERY DATA TO TEST

```sql
%%sql

SELECT c.name AS client_name, a.name AS animal_name, a.color, a.gender
FROM client c
JOIN animal a ON c.tin = a.tin;
```

Last executed at 2025-07-31 15:00:21 in 15ms

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

client_name	animal_name	color	gender
Maria Gomez	Bobby	Brown	Male

```sql
%%sql

SELECT m.name AS medication_name, m.lab, con.dosage
FROM visit v
JOIN results_in ri ON ri.tin = v.tin AND ri.omvid = v.omvid AND ri.name = v.name
                  AND ri.date = v.date AND ri.tin_owner = v.tin_owner
JOIN implies i ON i.tin = ri.tin AND i.omvid = ri.omvid AND i.name = ri.name
              AND i.date = ri.date AND i.tin_owner = ri.tin_owner AND i.code = ri.code
JOIN contains con ON con.pid = i.pid
JOIN medication m ON m.name = con.name
WHERE v.name = 'Bobby' AND v.date = '2025-07-31 10:00:00';
```

Last executed at 2025-07-31 15:01:00 in 20ms

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

medication_name	lab	dosage
Antihistamine	VetPharma	5mg/day

Renato Vivar Orellana
Data Science Engineer

KEY TAKEAWAYS

Real-world modeling:

- Applied database modeling principles to a veterinary hospital, covering people, animals, visits, diagnostics, and treatments.

ER design with reasoning: Carefully considered entity strength, specialization, and relationships, including:

- Weak entities (e.g., animal, visit, weight)
- Specializations (e.g., mammal → dog/cat/other, procedure → radiography/test)
- Optional attributes (e.g., SOAP notes as a specialization of visits)

Referential integrity enforced:

- All associations use foreign keys to maintain data consistency and support cascading relations across entities.

Business rules in design:

- Logical constraints like "an animal must belong to only one class" were documented to reflect domain-specific rules.

SQL Schema implementation:

- Translated the ER model into SQL with:
  - Composite primary keys
  - Clear foreign key constraints
  - Use of PostgreSQL syntax for table creation and integrity

Full data lifecycle:

- The model supports data from initial registration (clients, animals) through diagnostics, prescriptions, and procedures—ensuring a complete digital archive.

Renato Vivar Orellana
Data Science Engineer

GITHUB + CONTACT INFO

Renato Vivar Orellana

Data Science Engineer