GG P1 Project

- ist199271 Margarida Estrela (33%)
- ist199891 André Teixeira (33%)
- ist1100096 Tiago Costa (33%)

Prof. Flávio Ramos (Prof. João Marques and Prof. Carlos Felgueiras in the labs)

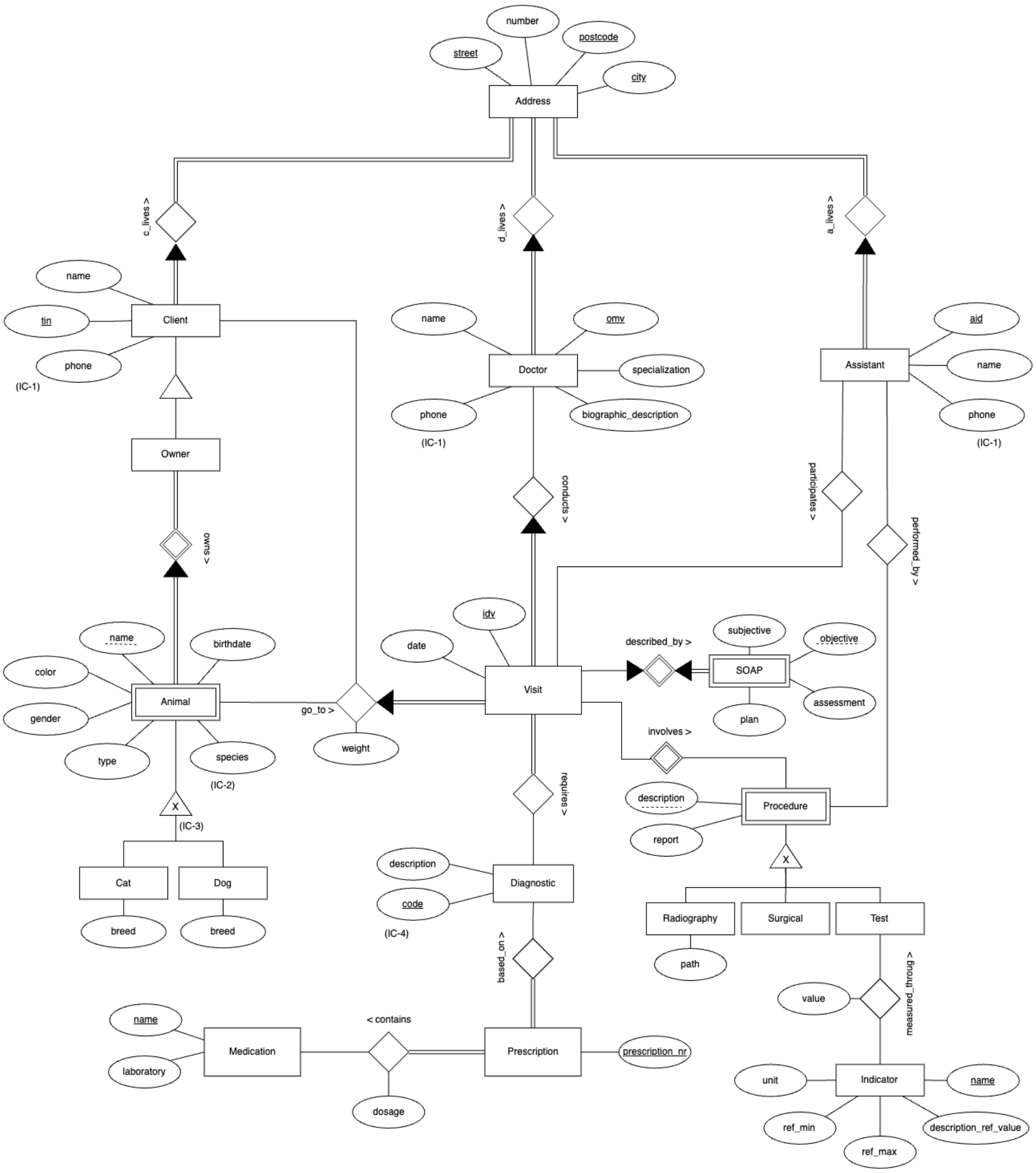
Lab Shift number: PB02

PART I – E-R Model

1. Proposed database design

An E-R model describing your proposed database design. Every design decision that can be captured in the E-R model should be represented in the diagram. Do not forget to include additional integrity constraints not captured in the E-R model. You may provide notes explaining the rationale behind non-trivial design decisions.

E-R Model



(IC-1): The phone number is unique.
(IC-2): For Cat and Dog species is None.
(IC-3): An Animal may only be a Dog or a Cat only if type is Mammals.
(IC-4): The diagnostic codes should be standardized (controlled vocabulary).

PART II – Relational Model

Database Schema

1. Create the tables and integrity constraints corresponding to the relational database schema obtained.

Do not forget to include additional integrity constraints not captured in the relational model.

Please use database `db`, which should already have been created. If not, follow the instructions in Lab01 to create it.

In [1]:
```
%load_ext sql
%sql postgresql+psycopg://db:db@postgres/db
```

Connecting to 'postgresql+psycopg://db:***@postgres/db'

In [28]:

```
%%sql

DROP TABLE IF EXISTS
    measured_through,
    performed_by,
    indicator,
    radiography,
    surgical,
    test,
    procedure,
    soap,
    contains,
    medication,
    based_on,
    prescription,
    requires,
    diagnostic,
    participates,
    conducts,
    go_to,
    visit,
    a_lives,
    assistant,
    d_lives,
    doctor,
    cat,
    dog,
    animal,
    owner,
    c_lives,
    client,
    address
CASCADE;

CREATE TABLE address (
    street Varchar(255),
    number Char(4) NOT NULL,
    postcode Varchar(10),
    city Varchar(30),
    PRIMARY KEY (street, postcode, city)
);

CREATE TABLE client (
    tin Integer,
    name Varchar(80) NOT NULL,
    phone Numeric(9),
    PRIMARY KEY (tin),
    UNIQUE (phone)
);

CREATE TABLE c_lives (
    tin Integer,
    street Varchar(255),
    postcode Varchar(10),
    city Varchar(30),
    PRIMARY KEY (tin),
    FOREIGN KEY (tin) REFERENCES client(tin),
    FOREIGN KEY (street, postcode, city) REFERENCES address(street, postc
```

```sql
);

CREATE TABLE owner (
    tin Integer,
    PRIMARY KEY (tin),
    FOREIGN KEY (tin) REFERENCES client(tin)
);

CREATE TABLE animal (
    tin Integer,
    name Varchar(80),
    color Varchar(80) NOT NULL,
    gender Varchar(80) NOT NULL,
    type Varchar(80) NOT NULL,
    species Varchar(80) NOT NULL,
    birthdate Date NOT NULL,
    PRIMARY KEY (tin, name),
    FOREIGN KEY (tin) REFERENCES owner(tin)
    -- For Cat and Dog species is None
    -- An Animal may only be a Dog or a Cat only if type is Mammals
);

CREATE TABLE cat (
    tin Integer,
    name Varchar(80),
    breed Varchar(80) NOT NULL,
    PRIMARY KEY (tin, name),
    FOREIGN KEY (tin, name) REFERENCES animal(tin, name)
);

CREATE TABLE dog (
    tin Integer,
    name Varchar(80),
    breed Varchar(80) NOT NULL,
    PRIMARY KEY (tin, name),
    FOREIGN KEY (tin, name) REFERENCES animal(tin, name)
);

CREATE TABLE doctor (
    omv Integer,
    name Varchar(80),
    phone Numeric(9),
    specialization Varchar(80),
    biographic_description Varchar(80),
    PRIMARY KEY (omv),
    UNIQUE (phone)
);

CREATE TABLE d_lives (
    omv Integer,
    street Varchar(255),
    postcode Varchar(10),
    city Varchar(30),
    PRIMARY KEY (omv),
    FOREIGN KEY (omv) REFERENCES doctor(omv),
    FOREIGN KEY (street, postcode, city) REFERENCES address(street, postc
);

CREATE TABLE assistant (
    aid Integer,
    name Varchar(80) NOT NULL,
    phone Numeric(9),
    PRIMARY KEY (aid),
    UNIQUE (phone)
);

CREATE TABLE a_lives (
    aid Integer,
    street Varchar(255),
    postcode Varchar(10),
    city Varchar(30),
    PRIMARY KEY (aid),
    FOREIGN KEY (aid) REFERENCES assistant(aid),
    FOREIGN KEY (street, postcode, city) REFERENCES address(street, postc
);

CREATE TABLE visit (
    idv Integer,
    date Timestamp NOT NULL,
    PRIMARY KEY (idv)
);

CREATE TABLE go_to (
    client_tin Integer,
    owner_tin Integer,
    name Varchar(80),
    idv Integer,
    weight Numeric(3,2) NOT NULL,
    PRIMARY KEY (idv),
    FOREIGN KEY (idv) REFERENCES visit(idv),
    FOREIGN KEY (owner_tin, name) REFERENCES animal(tin, name),
    FOREIGN KEY (client_tin) REFERENCES client(tin)
);

CREATE TABLE conducts (
    idv Integer,
    omv Integer,
    PRIMARY KEY (idv),
    FOREIGN KEY (idv) REFERENCES visit(idv),
    FOREIGN KEY (omv) REFERENCES doctor(omv)
);

CREATE TABLE participates (
    idv Integer,
    aid Integer,
    PRIMARY KEY (idv),
    FOREIGN KEY (idv) REFERENCES visit(idv),
    FOREIGN KEY (aid) REFERENCES assistant(aid)
);

CREATE TABLE diagnostic (
    code Char(10),
    description Varchar(120) NOT NULL,
    PRIMARY KEY (code)
    -- Every code must be standardized
);

CREATE TABLE requires (
    idv Integer,
    code Varchar(10),
    PRIMARY KEY (idv),
    FOREIGN KEY (idv) REFERENCES visit(idv),
    FOREIGN KEY (code) REFERENCES diagnostic(code)
);

CREATE TABLE prescription (
    prescription_nr Integer,
    PRIMARY KEY (prescription_nr)
);

CREATE TABLE based_on (
    code Varchar(10),
    prescription_nr Integer,
    PRIMARY KEY (code),
    FOREIGN KEY (code) REFERENCES diagnostic(code),
    FOREIGN KEY (prescription_nr) REFERENCES prescription(prescription_nr
);

CREATE TABLE medication (
    name Varchar(80),
    laboratory Varchar(80) NOT NULL,
    PRIMARY KEY (name)
);

CREATE TABLE contains (
    prescription_nr Integer,
    name Varchar(80),
    dosage Numeric(3,2) NOT NULL,
    PRIMARY KEY (prescription_nr),
    FOREIGN KEY (prescription_nr) REFERENCES prescription(prescription_nr
    FOREIGN KEY (name) REFERENCES medication(name)
);

CREATE TABLE soap (
    idv Integer,
    subjective Varchar(120) NOT NULL,
    objective Varchar(120),
    assessment Varchar(120) NOT NULL,
    plan Varchar(120) NOT NULL,
    PRIMARY KEY (idv, objective),
    FOREIGN KEY (idv) REFERENCES visit(idv)
);

CREATE TABLE procedure (
    idv Integer,
    description Varchar(80),
    report Varchar(200) NOT NULL,
    PRIMARY KEY (idv, description),
    FOREIGN KEY (idv) REFERENCES visit(idv)
);

CREATE TABLE radiography (
    idv Integer,
    description Varchar(80),
    path Varchar(80) NOT NULL,
    PRIMARY KEY (idv, description),
    FOREIGN KEY (idv, description) REFERENCES procedure(idv, description)
);

CREATE TABLE surgical (
    idv Integer,
    description Varchar(80),
    PRIMARY KEY (idv, description),
    FOREIGN KEY (idv, description) REFERENCES procedure(idv, description)
);

CREATE TABLE test (
    idv Integer,
    description Varchar(80),
    PRIMARY KEY (idv, description),
    FOREIGN KEY (idv, description) REFERENCES procedure(idv, description)
);

CREATE TABLE indicator (
    name Varchar(80),
    unit Varchar(5) NOT NULL,
    ref_min Numeric(3,2) NOT NULL,
    ref_max Numeric(3,2) NOT NULL,
    description_ref_value Varchar(80) NOT NULL,
    PRIMARY KEY (name)
);

CREATE TABLE measured_through (
    idv Integer,
    description Varchar(80),
    name Varchar(80),
    value Numeric(3,2) NOT NULL,
    PRIMARY KEY (idv, description, name),
    FOREIGN KEY (idv, description) REFERENCES test(idv, description),
    FOREIGN KEY (name) REFERENCES indicator(name)
);

CREATE TABLE performed_by (
    aid Integer,
    idv Integer,
    description Varchar(80),
    PRIMARY KEY (idv, description),
    FOREIGN KEY (idv, description) REFERENCES procedure(idv, description)
    FOREIGN KEY (aid) REFERENCES assistant(aid)
);
```

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

Out[28]: