# Database Design in 3NF

owners (id, first_name, last_name, street, city, state, zip, phone, email, active)

pets (id, name, animal_id, owner_id, female, date_of_birth, active)

visits (id, pet_id, date, weight, overnight_stay, total_charge)

animals (id, name, active)

medicines (id, name, description, stock_amount, method, unit, vaccine)

medicine_costs (id, medicine_id, cost_per_unit, start_date, end_date)

animal_medicines (id, animal_id, medicine_id, recommended_num_of_units)

visit_medicines (id, visit_id, medicine_id, units_given, discount)

procedures (id, name, description, length_of_time, active)

treatments (id, visit_id, procedure_id, successful, discount)

procedure_costs (id, procedure_id, cost, start_date, end_date)

notes (id, notable_type, notable_id, title, content, user_id, date)

users (id, first_name, last_name, role, username, password_digest, active)

**Underlines**:
Solid underlined fields are primary keys;
Dotted underlined fields are foreign keys;
Double underlined fields are composite keys that are both primary and foreign keys.

---

*Database Design Notes:*

1. Strictly speaking, having zip code in the owners table creates a transitive dependency, but given the limited size of the system (the greater Pittsburgh area) there is no need to normalize and move zip code and primary city & state into its own table.

2. To minimize the impact of rounding errors associated with floats, we will follow the industry norm of recording all costs as integers with the base monetary unit being cents, not dollars.

3. In a similar vien, length_of_time in the procedures table is recorded as an integer representing the number of minutes the procedure is expected to take.

4. A medicine's current cost is determined by finding the one medcine_cost record that has a NULL value in end_date. Similarly a procedure's current cost is found by identifying the one record that has a null end_date. Triggers will need to set up to automatically add the end date to the record prior to adding a new record to these tables.

5. Since usernames are used for uniquely identifying users during login, the values of username must be unique even though it is not a primary key.

6. All phone numbers are saved as a string of numbers without any other characters. Phone numbers include area code, prefix, and suffix as one 10-digit numerical string.

7. Units in the medicines table represents the measurement unit (e.g., mililiters, ounces, tablets) used to apportion that particular medicine. The stock_amount is an integer which represents the number of units currently on hand in the PATS office. This amount ought to be decremented automatically whenever a new visit_medicine record is inserted.

8. In the medicines table the field method is a string indicating the method by which a medicine is administered. At this time this is limited to one of three string options: 'injection', 'oral', 'intravenous'. It should be noted that new methods of administration are unlikely to be made available in the future so it would be prudent to constrain input to one of these values.

9. Discounts in the visit_medicine and treatments tables is by default set to zero (no discount given) and has a maximum value of 1, which represents a 100% discount (making the medicine or treatment free). The multiplier used in calculating cost is (1 – discount). The discount is stored as a two-digit decimal. Under no circumstances will this field ever fall outside the prescribed boundaries.

10. In the visits table, total charge is a summary field which is calculated after the visit details are recorded. Total charge is as follows:

    *sum for each medicine administered during the visit the medicine's unit cost multiplied by the number of units multiplied by (1 – discount) for that particular medicine given,*

    plus …

    *sum for each procedure given during the visit the procedure's cost multiplied by (1 – discount) given for that particular treatment.*

    This field is calculated via a function in the database that is executed after inserting or updating visit details.

11. In the visits table, overnight_stay is a summary field which is calculated after the visit details are recorded. If the sum of length_of_time for each procedure administered is greater than 720 minutes (i.e., 12 hours) then this flag is set to true; otherwise it is false.

12. Because we will later connect this database to a Rails application, we eschew the use of composite keys in favor of the Rails convention of specifying a new primary key for each table simply called 'id'.

13. All foreign key constraints will cascade updates but restrict any deletions *except* in the following cases:

    a. in the case of deleting procedures, deletions are allowed if a procedure has never been administered as a treatment; otherwise it is restricted. Moreover, if a procedure is deleted, then any costs associated with the procedure should also be deleted.

    b. in the case of deleting medicines, deletions are allowed if the medicine has never been administered during a visit; otherwise it is restricted. Moreover, if a medicine is deleted, then any costs associated with the medicine should also be deleted.

14. Notes have been made into polymorphic assocations with owners, pets and visits. This will allow later flexibility to attach note functionality at a later point to other entities such as medicines, procedures, treatments, visit_medicines and even possibly to procedure_ and medicine_costs as needed/desired. The string value in the notable_type field represents the name of the table that the notable_id is referencing.

15. All fields in all tables are required *except* the following:

    a. procedure_costs(end_date) *[current cost for given procedure is where end_date is null]*
    b. medicine_costs(end_date) *[current cost for given medicine is where end_date is null]*
    c. procedures(description) *[simple procedures may have names that say it all…]*
    d. owners(phone, email) *[owners may not wish to disclose this information]*
    e. pets(date_of_birth) *[unknown for some pets, such as adopted strays]*
    f. treatments(successful) *[in some cases success may not be known for some time]*
    g. visits(weight) *[in emergency situations, weight might not get recorded]*
    h. animal_medicines(recommended_num_of_units) *[unknown in some cases]*

16. All active fields are by default set to true unless otherwise specified.

17. The vaccine field in the medicines table is assumed to be false when a new medicine is created unless otherwise specified.

18. The state field in the owners table is assumed to be 'PA' when a new owner is created unless otherwise specified.

19. Because of the need for full text searching on medicines(description), it is important to add a GIN index on this field using the to_tsvector() function.

20. Because of the difficulties associate with deleting visit_medicine records (the medicine can't be taken back and the medicine amount cannot be returned), we will remove delete privileges on that table for the 'pats' user. Since treatments have similar difficulties, the same privilege restrictions will be applied to that table. Finally, we will restrict the ability of the 'pats' user to edit the units_given field in the visit_medicines table.

21. For security purposes, all database users who are not superusers are limited to SELECT access only on the users table.