PET SHOP (ER DIAGRAM & NORMALIZATION)

OBJECTIVE OF THE PROJECT

- The objectives of the project is to provide web based interface to a **petshop** owner to manages his **petshop** activities.
- To provide an option for storing and managing the basic information about pets and pet products in the shop.
- To provide an option for storing and managing the sales details of the shop.
- To provide an option for storing and managing the basic information about the customer.
- To track the information about sold pets and products to a customer

IMPLEMENTATION

ER DIAGRAM

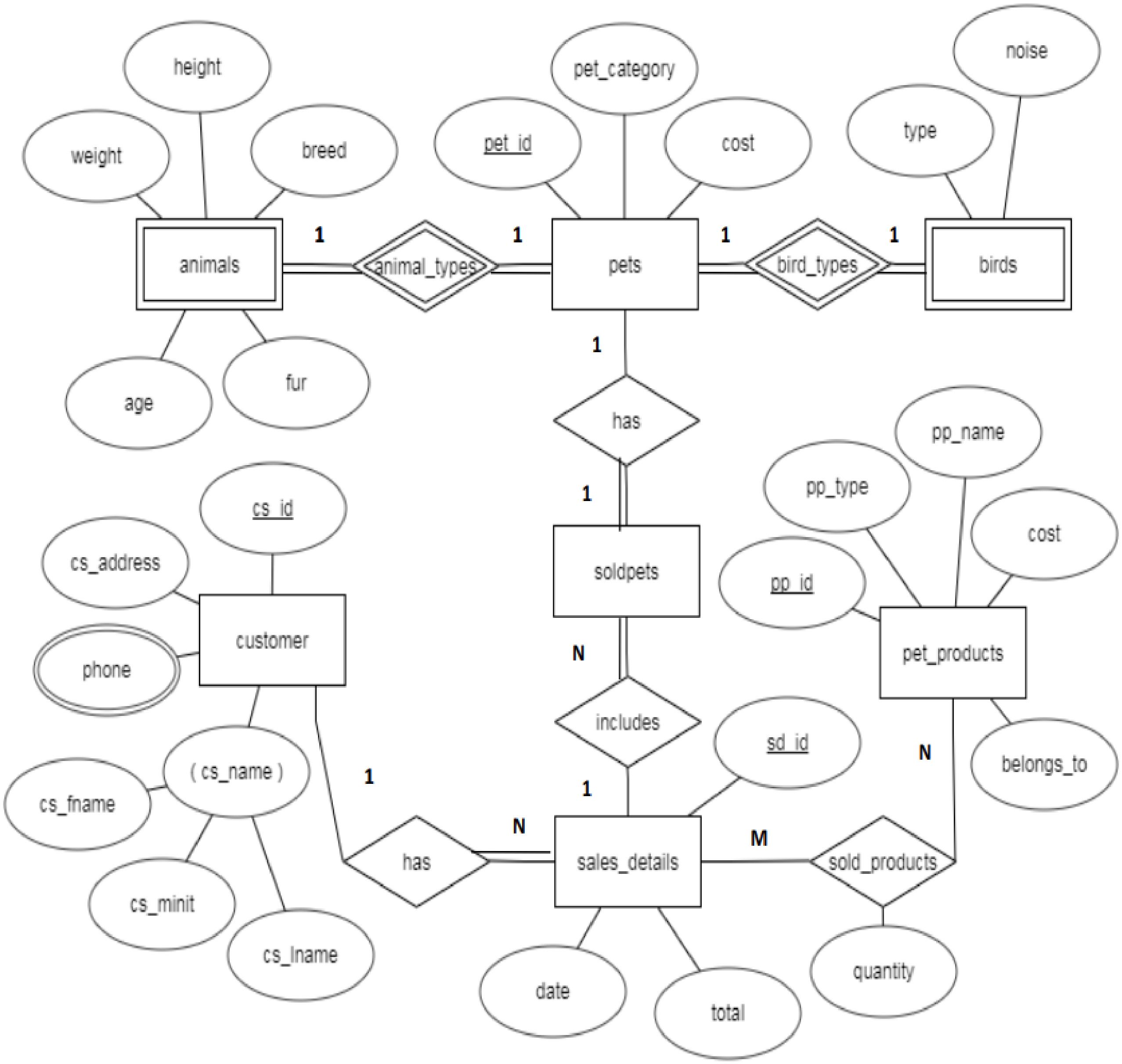
An entity-relationship model (ER model) describes inter-related things of Interest in a specific domain of knowledge. An ER model is composed of entity types(which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

The main components of ER model are: entity set and relationship set.
Here are the geometric shapes and their meaning in an ER Diagram

Rectangle : Represents Entity sets.
Ellipses : Attributes.

Diamonds: Relationship set.
Lines : They link attributes to Entity Sets and this to Relationship Set.



ER DIAGRAM OF PETSHOP MANAGEMENT SYSTEM

NORMALIZE THE RELATIONS

Normalization is a process of organizing the data in database to avoid data redundancy,
insertion anomaly, update anomaly & deletion anomaly.
There are three main types of normal forms:
a) First normal form(1NF)
b) Second normal form(2NF)
c) Third normal form(3NF)

1. First normal form (1NF):

a) As per the rule of first normal form, an attribute (column) of a table cannot hold multiple
values.
b) It should hold only atomic values.
This table holds only the atomic values company id and the company name and no multiple
values are stored in this table so it can be considered as the 1NF.

2.Second normal form (2NF):

A table is said to be in 2NF if both the following conditions hold:
a) Table is in 1NF (First normal form)
b) No non-prime attribute is dependent on the proper subset of any candidate key of table.
c) An attribute that is not part of any candidate key is known as non-prime attribute.

3. Third Normal form (3NF):

A table design is said to be in 3NF if both the following conditions hold:
a) Table must be in 2NF
b) Transitive functional dependency of non-prime attribute on any super key should be
removed.
c) An attribute that is not part of any candidate key is known as non-prime attribute.
In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each
functional dependency X-> Y at least one of the following conditions hold:
X is a super key of table

Y is a prime attribute of table
An attribute that is a part of one of the candidate keys is known as prime attribute.
The relations are already in the normalized form in the schema diagram without any
redundancy.

CREATION OF TABLES

1. CREATION OF PETS TABLE

```
create table pets( pet_id varchar(9) not null,
pet_category varchar(15) not null,
cost int(11) not null,
primary key(pet_id));
```

2. CREATION OF ANIMALS TABLE

```
create table animals(pet_id varchar(9) not null,
breed varchar(30) not null,
weight float not null,
height float not null,
age int(11) not null,
fur varchar(15) not null,
primary key(pet_id),
forgien key(pet_id) references pets(pet_id) on delete cascade);
```

2. CREATION OF BIRDS TABLE

```
create table birds(pet_id varchar(9) not null,
type varchar(25) not null,
noise varchar(10) not null,
primary key(pet_id),
forgien key(pet_id) references pets(pet_id) on delete cascade);
```

3. CREATION OF PET_PRODUCTS TABLE

```
create table pet_products(pp_id varchar(9) not null,
pp_name varchar(30) not null,
pp_type varchar(20) not null,
cost int(11) not null,
belongs_to varchar(20) not null,
primary key(pp_id));
```

5.CREATION OF CUSTOMER TABLE

```
create table customer(cs_id varchar(9) not null,
cs_fname varchar(10) not null,
cs_minit varchar(10) not null,
cs_lname varchar(10) not null,
cs_address varchar(30)not null,
primary key(cs_id));
```

6.CREATION OF PHONE TABLE

```
create table phone (cs_id varchar(9) not null,
cs_phone bigint(10) not null,
primary key(cs_id,cs_phone),
foreign key(cs_id) references customer(cs_id)on delete cascade);
```

7.CREATION OF SALES_DETAILS TABLE

```
create table sales_details(sd_id varchar(9) not null,
cs_id varchar(9) not null,
date date not null,
total int(11) not null,
primary key(sd_id,cs_id),
foreign key(cs_id)references customer(cs_id)on delete cascade);
```

8.CREATION OF SOLD_PETS TABLE

```
create table sold_pets(sd_id varchar(9) not null,
pet_id varchar(9) not null,
primary key(pet_id),
foreign key(sd_id)references sales_details(sd_id)on delete cascade,
foreign key(pet_id)references pets(pet_id)on delete cascade);
```