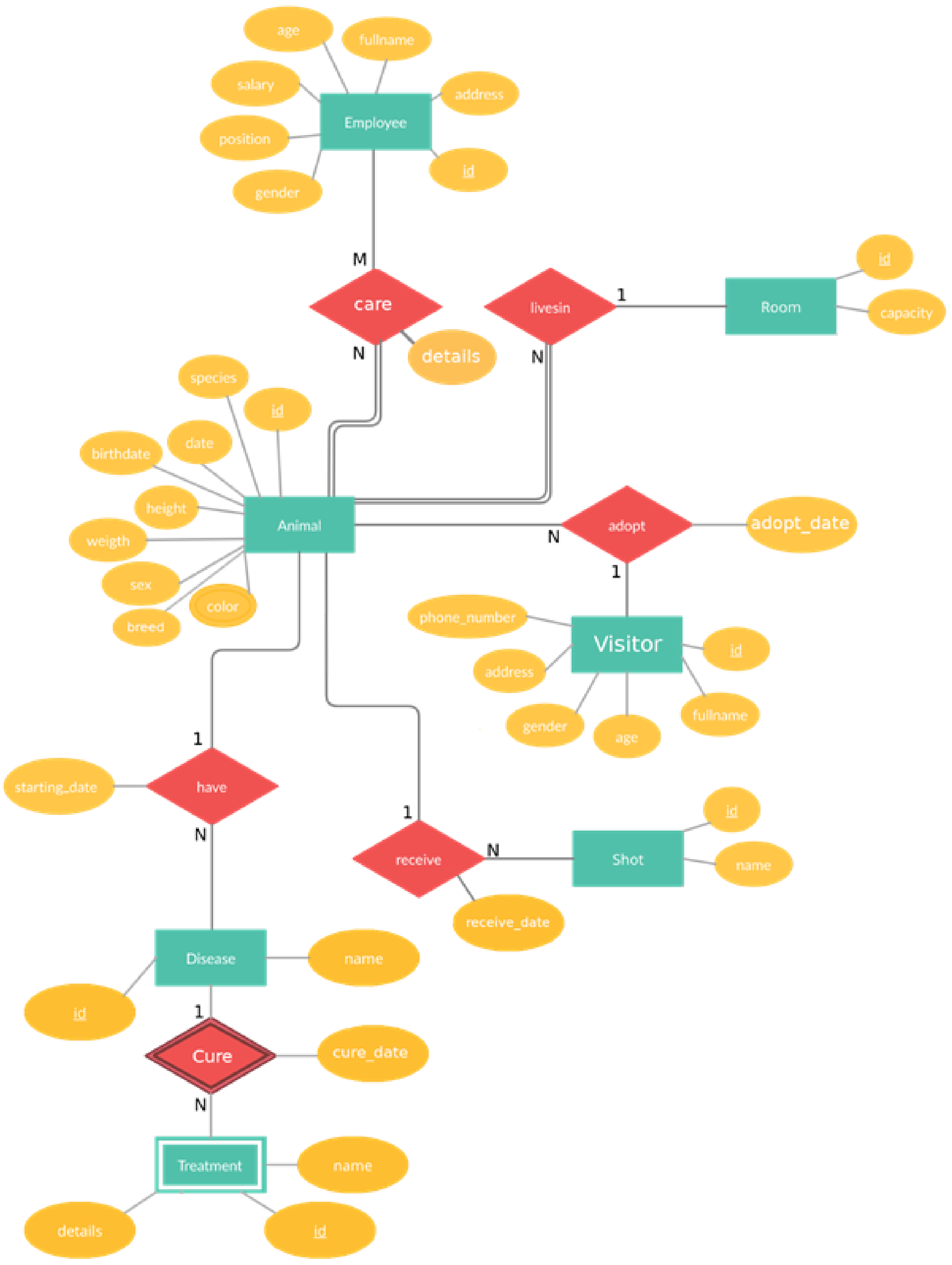
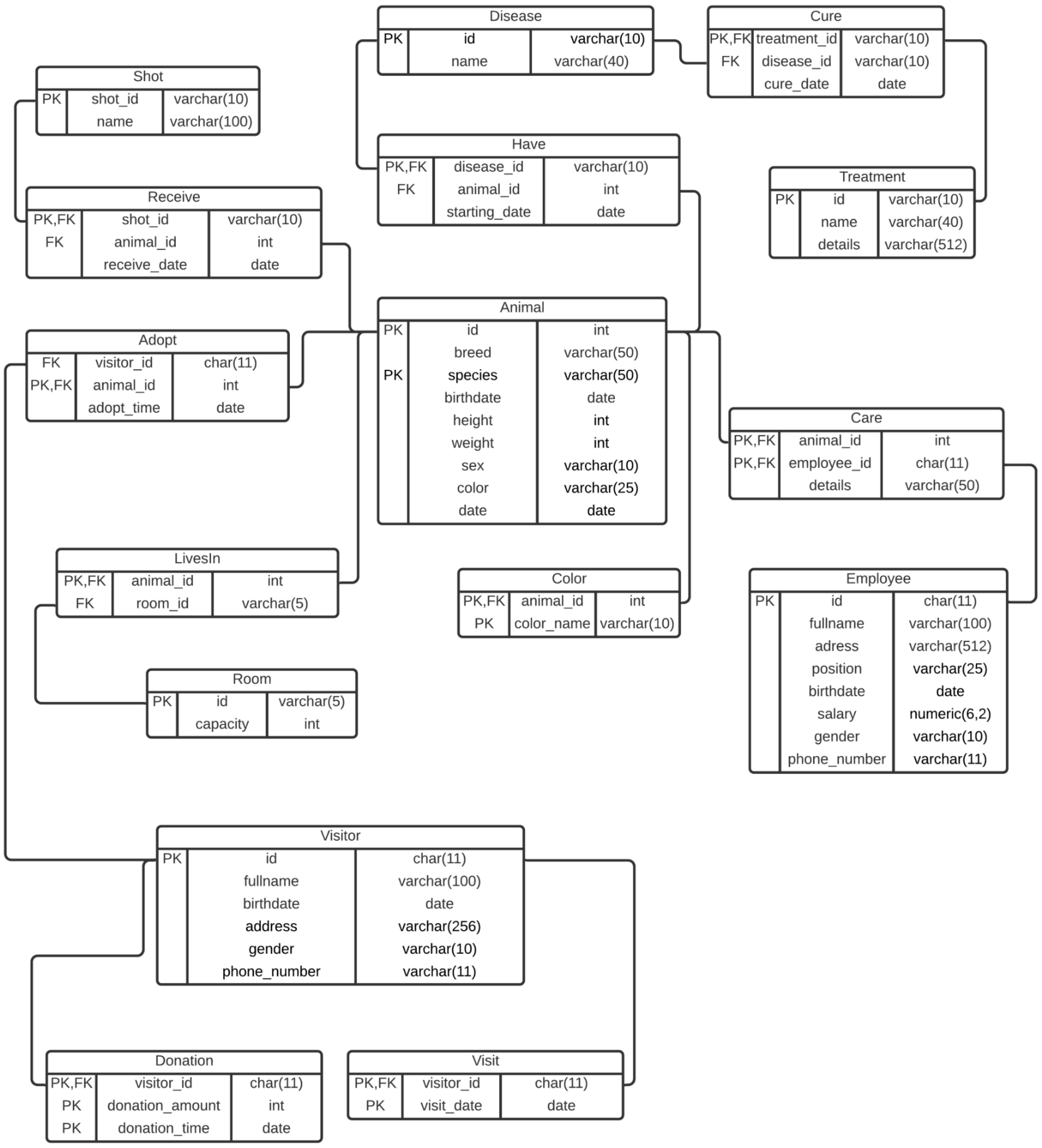
Animal Shelter Management System

Project Group 65
Meltem AKKOCA
Hatice Esra YILMAZ
Uğur KELLECİOĞLU

ER Diagram



Relational Schema



Disease

PK	id	varchar(10)
	name	varchar(40)

Cure

PK,FK	treatment_id	varchar(10)
FK	disease_id	varchar(10)
	cure_date	date

Shot

PK	shot_id	varchar(10)
	name	varchar(100)

Have

PK,FK	disease_id	varchar(10)
FK	animal_id	int
	starting_date	date

Treatment

PK	id	varchar(10)
	name	varchar(40)
	details	varchar(512)

Receive

PK,FK	shot_id	varchar(10)
FK	animal_id	int
	receive_date	date

Animal

PK	id	int
	breed	varchar(50)
PK	species	varchar(50)
	birthdate	date
	height	int
	weight	int
	sex	varchar(10)
	color	varchar(25)
	date	date

Adopt

FK	visitor_id	char(11)
PK,FK	animal_id	int
	adopt_time	date

Care

PK,FK	animal_id	int
PK,FK	employee_id	char(11)
	details	varchar(50)

LivesIn

PK,FK	animal_id	int
FK	room_id	varchar(5)

Color

PK,FK	animal_id	int
PK	color_name	varchar(10)

Employee

PK	id	char(11)
	fullname	varchar(100)
	adress	varchar(512)
	position	varchar(25)
	birthdate	date
	salary	numeric(6,2)
	gender	varchar(10)
	phone_number	varchar(11)

Room

PK	id	varchar(5)
	capacity	int

Visitor

PK	id	char(11)
	fullname	varchar(100)
	birthdate	date
	address	varchar(256)
	gender	varchar(10)
	phone_number	varchar(11)

Donation

PK,FK	visitor_id	char(11)
PK	donation_amount	int
PK	donation_time	date

Visit

PK,FK	visitor_id	char(11)
PK	visit_date	date

Changes

1- In the Phase 1 we created the “animal” table with “color” attribute but now we dropped it by “alter table”, since it is a multivalued variable and has its own table.
2- Before, we created the “room” table with “species” attribute that normally exists in the “animal” table. Now we dropped it by “alter table” because we decided to show just the rooms’ ids and capacities as its features.
3- We turned the “LivesIn” relationship between the “room” and “animal” tables into a table in order to match the animals and rooms according to their ids.
4- Since we show their match in the “LivesIn” table, we dropped the “room id” column in the “animal” table by “alter table”.

Descriptions of Tables

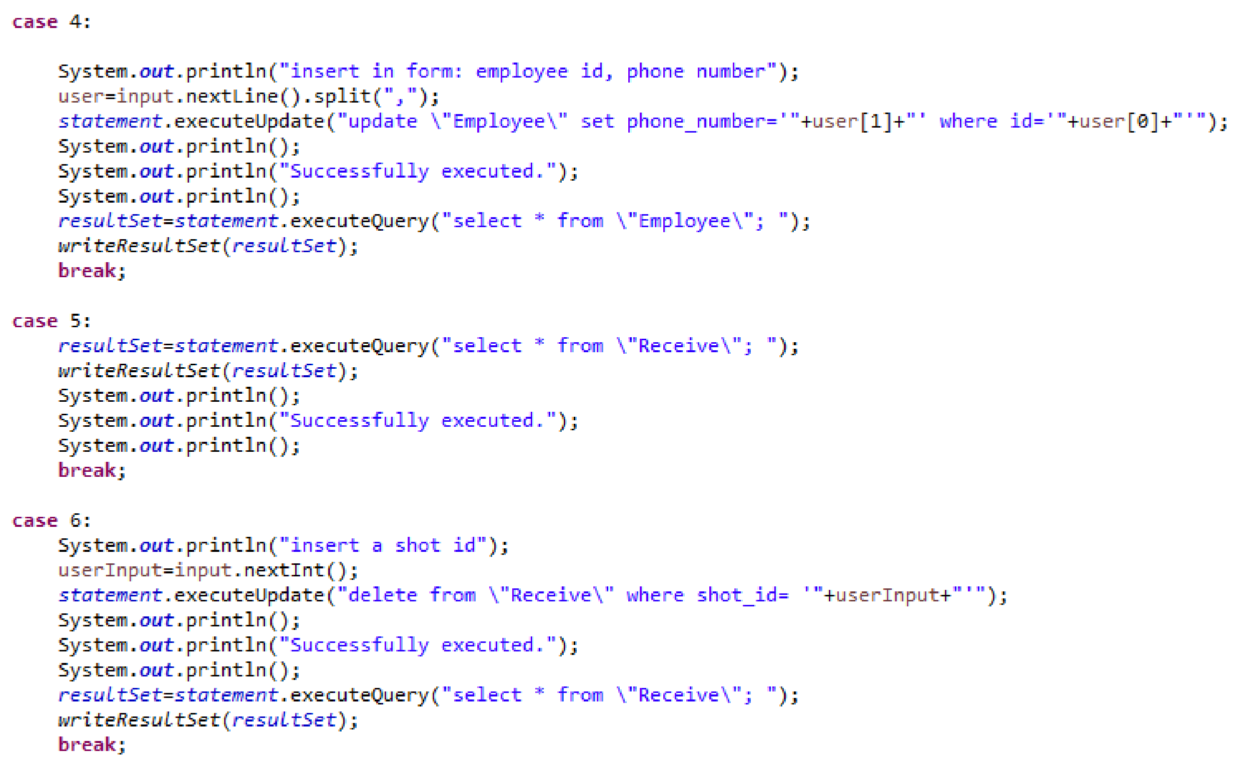
1- **Animal** is an entity which is known by id (primary key), species, breed, birthdate, sex, colors which is a multivalued attribute with its own attributes that are animal id (foreign key, primary key) and color name(primary key), the date when animals came to the shelter, the height and weight that they have at the first day in the shelter.
2- There is an **employee** entity which is defined by unique id (primary key), full name, address, age, salary, position and gender.
3- Between animal and employee entities there is a **care** relation. This relation contains animal id and employee id as foreign keys and both are primary keys. In this relation each animal must be taken care of by employees in terms of food, washing, health as the details attribute. Employees can look after many animals or they can be security guards, cleaning staff, managers etc.
4- The shelter has a **room** entity which the animals **live in** (the relation between animals and rooms), with the attributes that are the rooms’ ids (primary key) and capacities.
5- In the **“LivesIn”** entity there are animal ids as foreign key and primary key, and room ids as foreign key, in order to match the rooms and animals.
6- Animals might have **diseases**, which is an entity that is described by diseases’ unique ids (primary key) and names. Also, there is a starting date attribute of animal’s disease. Moreover, a disease can impact only one animal because of the disease's special id.

7- Between animal and disease there is a **“have”** relation which has disease id (primary key, foreign key), animal id (foreign key) and starting date of disease.
8- There is a **treatment** entity which depends on the existence of disease, which makes it a weak entity. A disease can be cured by many treatments but a treatment can be applied on one disease. A treatment has a special id (primary key), name and details.
9- **Cure** is a relation that connects disease and treatment entities with its attributes that are treatment id(primary key, foreign key), disease id(foreign key) and cure date.
10- There is a **shot** entity with shot id (primary key) and name attributes. Animals may receive (the relation between them) many shots but one shot can be given to only one animal. **Receive** relation has shot id (foreign key and primary key), animal id (foreign key) and receive date attributes.
11- There is a **visitor** entity which is defined by visitors’ unique id (primary key), full name, age, gender, address, visit dates which is a multivalued attribute with visitor id (primary key, foreign key) and visit date (primary key) attributes, and donation which is also a multivalued attribute with visitor id (primary key, foreign key), donation amount (primary key) and donation time (primary key) attributes.
12- Visitors can also **adopt** (the relation between visitor and animal entities) many animals, which is specified by animal id (primary key, foreign key), visitor id (foreign key) and adopt time.

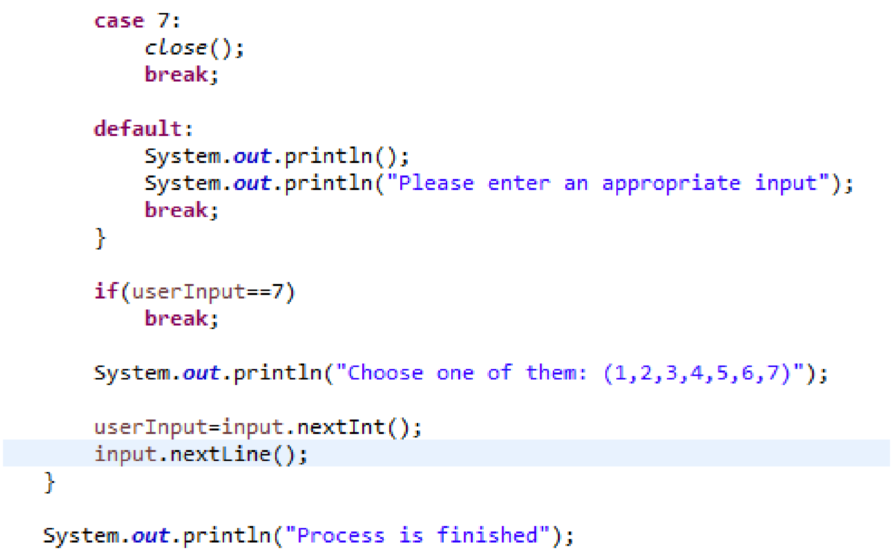
Java Application Program

1- Firstly, we selected the “animal”, “employee” and “receive” tables.
2- We offered some choices to the user to list the information of the three tables or to make changes on them. These changes are inserting an animal to the “animal” table, updating an employee’s phone number through the employee’s id, and deleting from the “receive” table through a shot’s id.
3- According to the user’s choice we created a switch-case statement.
4- If the user makes a change, after that we list the new version of the table in the related case statement.
5- We put the switch-case statement in a while loop. Every time, we ask the choice question until the user chooses to exit.

```java
int userInput;
String[] user;

System.out.println("Choose one of them: (1,2,3,4,5,6,7)");
System.out.println("1-List the information of animals");
System.out.println("2-Insert an animal");
System.out.println("3-List the information of employees");
System.out.println("4-Update phone number of an employee");
System.out.println("5-List the information of receive table");
System.out.println("6-Delete from receive table");
System.out.println("7-Exit from the program");

userInput=input.nextInt();
input.nextLine();


while(userInput!=7) {

    switch(userInput) {

    case 1:
        resultSet=statement.executeQuery("select * from \"Animal\"; ");
        writeResultSet(resultSet);
        System.out.println();
        System.out.println("Successfully executed.");
        System.out.println();
        break;

    case 2:

        System.out.println("insert in form: id, breed, species, birthdate, height, weight, sex, date ");
        user=input.nextLine().split(",");

        statement.executeUpdate("insert into \"Animal\" values( " +user[0]+ " , '"+user[1]+"' , '"+user[2]+"' , "
            + " date '"+user[3]+"', "+user[4]+", "+user[5]+", '"+user[6]+"', date '"+user[7]+"')");
        System.out.println();
        System.out.println("Successfully executed.");
        System.out.println();
        resultSet=statement.executeQuery("select * from \"Animal\"; ");
        writeResultSet(resultSet);
        break;

    case 3:
        resultSet=statement.executeQuery("select * from \"Employee\"; ");
        writeResultSet(resultSet);
        System.out.println();
        System.out.println("Successfully executed.");
        System.out.println();
        break;
```



```java
case 4:

    System.out.println("insert in form: employee id, phone number");
    user=input.nextLine().split(",");
    statement.executeUpdate("update \"Employee\" set phone_number='"+user[1]+"' where id='"+user[0]+"'");
    System.out.println();
    System.out.println("Successfully executed.");
    System.out.println();
    resultSet=statement.executeQuery("select * from \"Employee\"; ");
    writeResultSet(resultSet);
    break;

case 5:
    resultSet=statement.executeQuery("select * from \"Receive\"; ");
    writeResultSet(resultSet);
    System.out.println();
    System.out.println("Successfully executed.");
    System.out.println();
    break;

case 6:
    System.out.println("insert a shot id");
    userInput=input.nextInt();
    statement.executeUpdate("delete from \"Receive\" where shot_id= '"+userInput+"'");
    System.out.println();
    System.out.println("Successfully executed.");
    System.out.println();
    resultSet=statement.executeQuery("select * from \"Receive\"; ");
    writeResultSet(resultSet);
    break;
```



```java
      case 7:
          close();
          break;

      default:
          System.out.println();
          System.out.println("Please enter an appropriate input");
          break;
      }

      if(userInput==7)
          break;

      System.out.println("Choose one of them: (1,2,3,4,5,6,7)");

      userInput=input.nextInt();
      input.nextLine();
  }

  System.out.println("Process is finished");
```