ZOO MANAGEMENT

SAKSHI SANCHETI B 207266

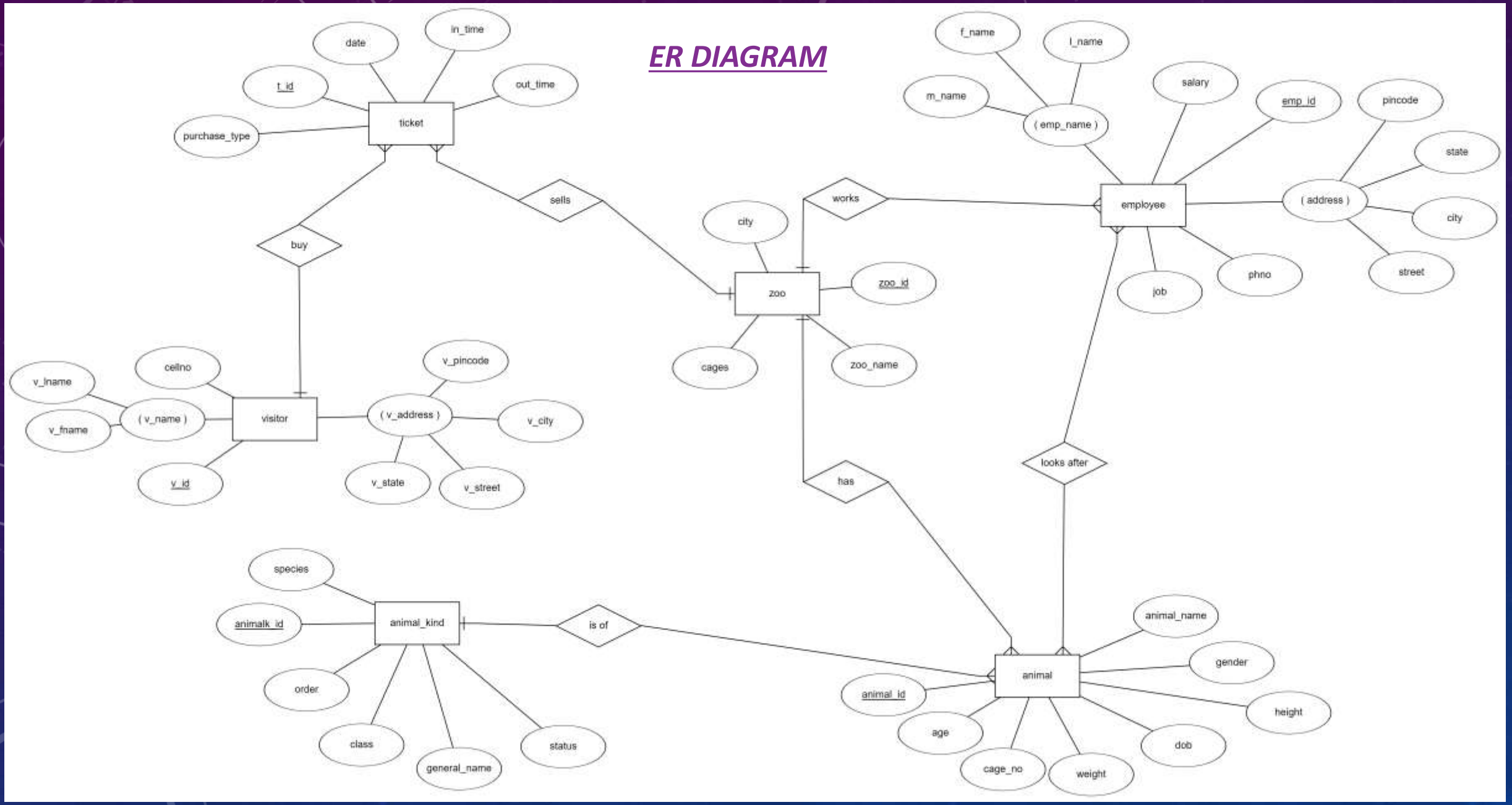
SUMAIYYA ANWAR B 207273

NIKITHA PUCHALA B 207256

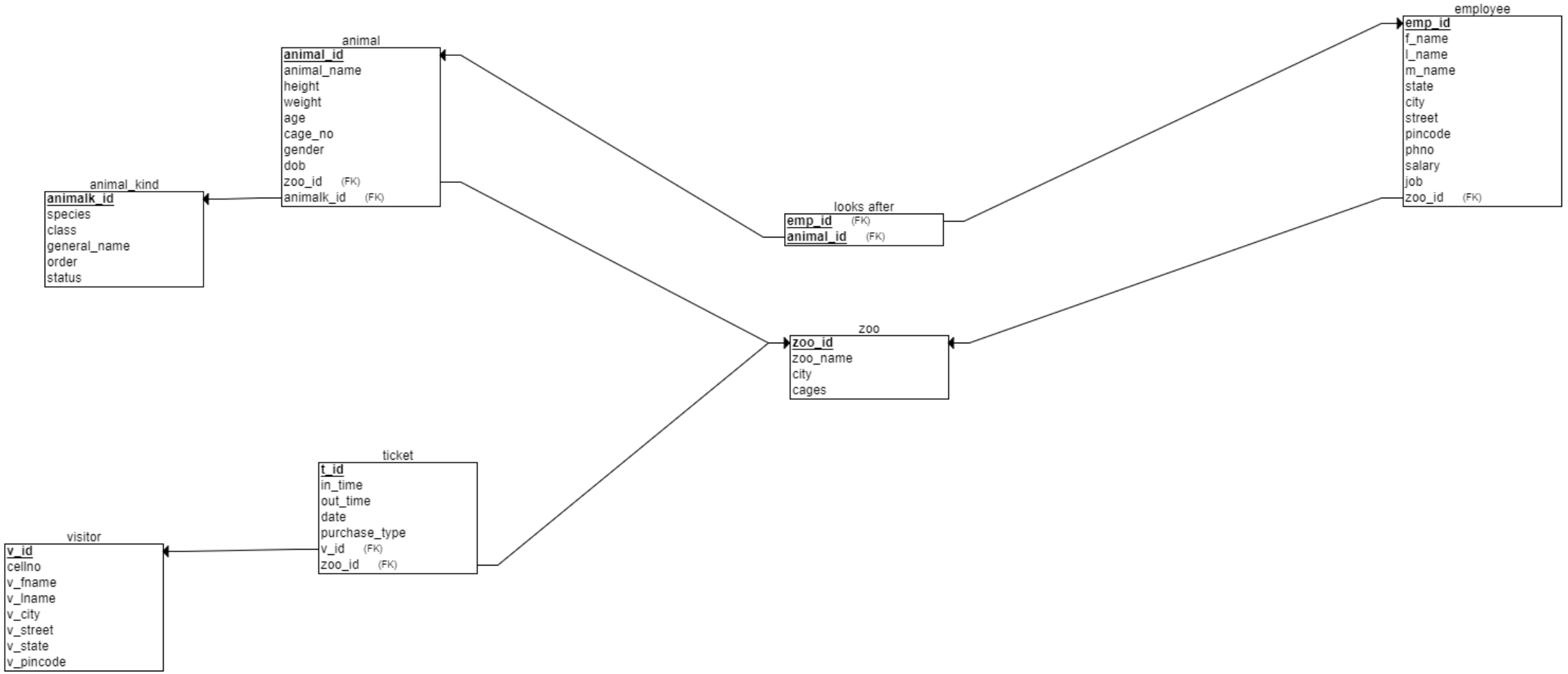
PROBLEM STATEMENT:

To help for conservation of the Wildlife and to raise awareness amongst the future generation about the preservation of fauna, create a database for zoos. This database has information about animals present in zoo, employees who work and take care of animals at the zoo. Also, the databases has information about the visitors who visit zoo.

ER DIAGRAM


date
in_time
t_id
out_time
purchase_type
ticket
sells
buy
f_name
l_name
m_name
(emp_name)
salary
emp_id
pincode
state
city
street
(address)
employee
works
city
zoo
zoo_id
cages
zoo_name
job
phno
cellno
v_pincode
v_lname
(v_name)
v_fname
visitor
(v_address)
v_city
v_id
v_state
v_street
has
looks after
species
animalk_id
animal_kind
is of
order
class
general_name
status
animal_name
gender
animal_id
animal
height
age
dob
cage_no
weight

RELATIONAL SCHEMA



Features of Database:

- Basic information about the animals is available and new species can be included into database, updates, deletions of existing animals is possible.
- A track of visitors is maintained with some information about them.
- New employees can be added into zoos and updates about them and deletion can be done
- Tickets bought for an individual zoo can be recorded with also the type of payment method.
- Information about every animal present in a zoo is stored with options to add new animals into zoos, update the changes like change of caretaker and deletion of animal.

NORMALISATION

Entity: Animal Kind

- Animak_id
- General_name
- species
- Order
- Class
- Status

This makes the Data one 1NF,2NF as all the columns are atomic, partial dependencies and transitive dependency are not present in data. The presence of functional dependency between the columns, order and class where neither of both are candidate keys break the 3NF, thus to reduce redundancy and to safeguard again anomalies the table is broken into 2 parts.

Entity: Taxonomy

- Order
- Class

This makes the data in 3NF, BCNF as the existing functional dependency between order and class is moved to another table.

Entity: Animal

- Animal_ID
- Animal_name
- Origin
- Gender
- Weight
- Height
- Cage_NO
- Age

This data follows all the normal forms 1NF, 2NF, 3NF and BCNF are there are no columns with non-atomic values, and each column has unique name, no partial, transitive dependencies.

Entity: Employee

- Employee ID
- Employee First name
- Employee Last name
- Salary
- Role
- Phone number
- Pin Code
- Street
- City
- State

The 1 st normal form is violated in column phone number, as an employee maybe possess more than 1 contact number, distinguishing his personal contact number and work. Thus, the table to broken with new table consisting contact numbers of each personnel. The 3NF, BCNF requirements are not meet too, as the street depends on pin code, City depends on Street, State depends on City, i.e., there is a transitive dependency among the columns where none of them are candidate keys. Street, City, State are moved to another table with primary key as pin code.

Table: Contact

- Employee ID
- Phone number

Table: Address

- Pin code
- Street
- City

Table: Location

- City
- State

This breakdown of table into smaller table gives it all 4 normal forms namely 1NF, 2NF, 3NF, BCNF.

Entity: Zoo

- Zoo ID
- Zoo name
- City
- Capacity

The data is in all 1NF, 2NF, 3NF, BCNF. Therefore, needing no breaking down of tables for normalization of data.

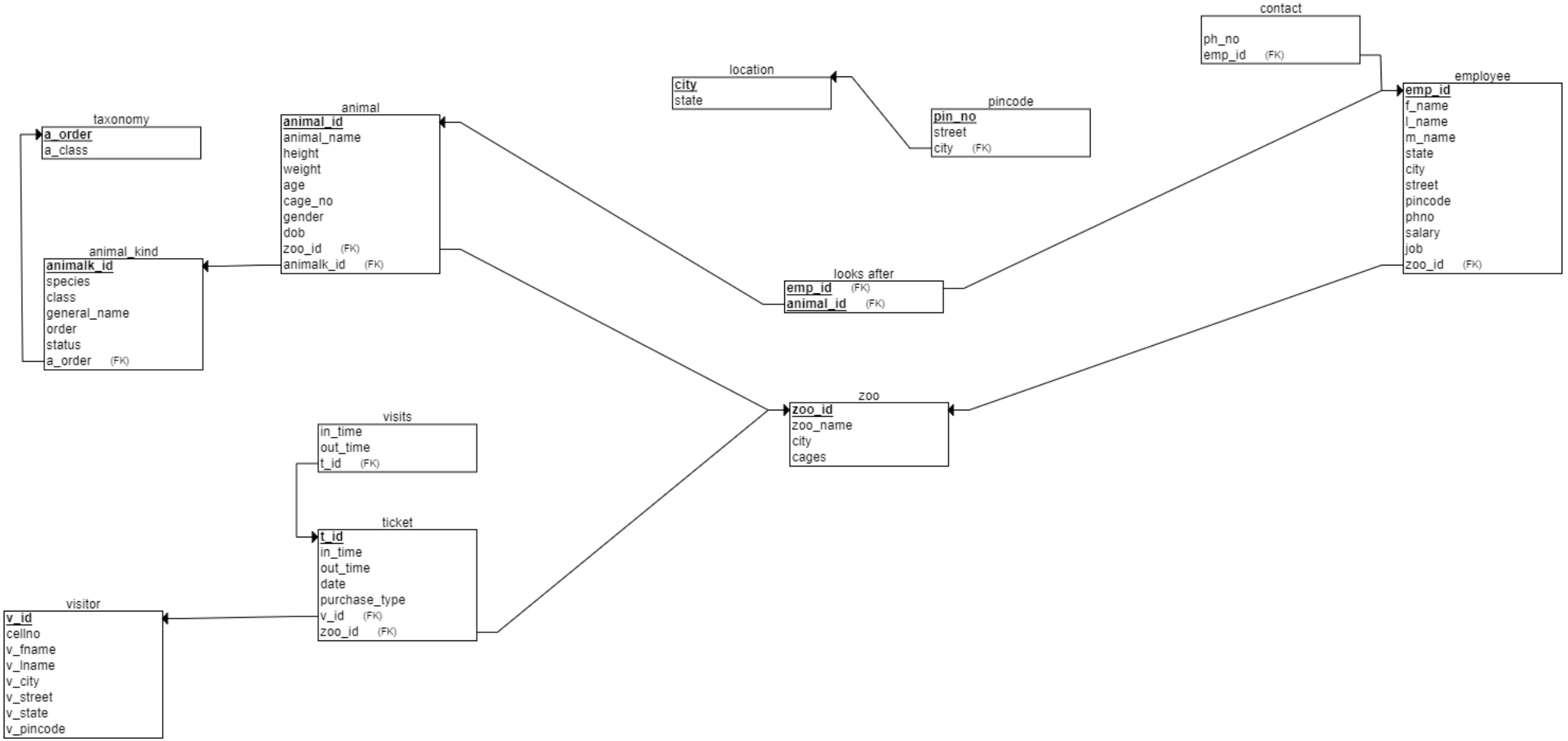
Entity: Visitor

- Visitor ID
- Visitor First Name
- Visitor Last Name
- Phone number
- Pin code
- Street
- City
- State

But creation of new tables is redundant as it already exists from breakdown of employee entity.

Relation Name	Type	Explanation
BUY	Many to 1	1 visitor can buy more than 1 ticket. and 1 ticket can be given to 1 visitor only
WORKS	Many to 1	A zoo has many working employees and many employees can work in a single zoo
SELLS	Many to 1	A zoo has many tickets and many tickets can be sold by one zoo
IS_OF	Many to 1	Many animals belong to 1 kingdom(animal species) and single animal have 1 animal kind
LOOKS_AFTER	Many to Many	Many Employees can look after one animal Many animals can be looked upon by single employee
HAS	Many to 1	One Zoo has many animals and many animals can live in One zoo

Relational Schema After Normalisation


contact
ph_no
emp_id (FK)
location
city
state
pincode
pin_no
street
city (FK)
employee
emp_id
f_name
l_name
m_name
state
city
street
pincode
phno
salary
job
zoo_id (FK)
taxonomy
a_order
a_class
animal
animal_id
animal_name
height
weight
age
cage_no
gender
dob
zoo_id (FK)
animalk_id (FK)
animal_kind
animalk_id
species
class
general_name
order
status
a_order (FK)
looks after
emp_id (FK)
animal_id (FK)
zoo
zoo_id
zoo_name
city
cages
visits
in_time
out_time
t_id (FK)
ticket
t_id
in_time
out_time
date
purchase_type
v_id (FK)
zoo_id (FK)
visitor
v_id
cellno
v_fname
v_lname
v_city
v_street
v_state
v_pincode

CREATION OF TABLES AND INSERTION OF DATA

```
create table zoo
(
zoo_id number primary key,
zoo_name varchar(100),
city varchar(100),
cages number
);

INSERT INTO zoo VALUES (10003, 'Sri Venkateswara Zoological Park', 'Tirupati', 69);
 INSERT INTO zoo VALUES (10004, 'Kakatiya Zoological Park', 'Hanmakonda', 102);
 INSERT INTO zoo VALUES (10007, 'Nehru Zoological Park', 'Hyderabad', 43);
INSERT INTO zoo VALUES (10009, 'Indira Gandhi Zoological Park', 'Visakhapatnam',39);
```

ZOO_ID	ZOO_NAME	CITY	CAGES
10009	Indira Gandhi Zoological Park	Visakhapatnam	39
10003	Sri Venkateswara Zoological Park	Tirupati	69
10004	Kakatiya Zoological Park	Hanmakonda	102
10007	Nehru Zoological Park	Hyderabad	43

```
create table taxonomy(
    a_order varchar(100) primary key,
  a_class varchar(100)
);
INSERT INTO taxonomy VALUES ('Anura', 'Amphibia’);
INSERT INTO taxonomy VALUES ('Artiodactyla', 'Mammalia’);
 INSERT INTO taxonomy VALUES ('Cardiida', 'invertebrate’);
INSERT INTO taxonomy VALUES ('Carnivora', 'Mammalia’);
 INSERT INTO taxonomy VALUES ('Casuariiformes', 'Aves’);
 INSERT INTO taxonomy VALUES ('Ciconiiformes', 'Aves’);
INSERT INTO taxonomy VALUES ('Crocodilia', 'Reptilia’);
INSERT INTO taxonomy VALUES ('Cyclopoida', 'Hexanauplia’);
 INSERT INTO taxonomy VALUES ('Decapoda', 'invertebrate’);
INSERT INTO taxonomy VALUES ('Galliformes', 'Aves’);
```

A_ORDER	A_CLASS
Anura	Amphibia
Artiodactyla	Mammalia
Cardiida	invertebrate
Carnivora	Mammalia
Casuariiformes	Aves
Ciconiiformes	Aves
Crocodilia	Reptilia
Cyclopoida	Hexanauplia
Decapoda	invertebrate
Galliformes	Aves

```
create table animal_kind
(
 animalk_id NUMBER primary key,
general_name varchar(100),
species varchar(100),
a_order varchar(100),
foreign key(a_order) references taxonomy(a_order),
status  varchar(10)
);

INSERT INTO animal_kind VALUES (105001, 'Bengal Tiger', 'Panthera tigris tigris', 'Carnivora', 'EN’);
 INSERT INTO animal_kind VALUES (105002, 'African Lion', 'Panthera leo leo', 'Carnivora', 'VU’);
 INSERT INTO animal_kind VALUES (105003, 'Chimpanzee', 'Pan troglodytes', 'primate', 'EN’);
INSERT INTO animal_kind VALUES (106001, 'King Cobra', 'Ophiophagus hannah', 'serpentes', 'VU’);
INSERT INTO animal_kind VALUES (102001, 'Openbill Stork', 'Anastomus oscitans', 'Ciconiiformes', 'LC’);
 INSERT INTO animal_kind VALUES (101001, 'Red Eye Tree Frog', 'Agalychnis callidryas', 'Anura', 'LC’);
INSERT INTO animal_kind VALUES (101002, 'Asiatic salamanders', 'Hynobius oyamai', 'Urodela', 'VU’);
```

INSERT INTO animal_kind VALUES (103001, 'Lined Seahorse', 'Hippocampus erectus', 'Syngnathiformes', 'VU');
INSERT INTO animal_kind VALUES (101003, 'Axolotl', 'Ambystoma mexicanum', 'Urodela', 'CR');
INSERT INTO animal_kind VALUES (104001, 'Crustaceans', 'Acanthocyclops hypogeus', 'Cyclopoida', 'VU');

ANIMALK_ID	GENERAL_NAME	SPECIES	A_ORDER	STATUS
105001	Bengal Tiger	Panthera tigris tigris	Carnivora	EN
105002	African Lion	Panthera leo leo	Carnivora	VU
102001	Openbill Stork	Anastomus oscitans	Ciconiiformes	LC
101001	Red Eye Tree Frog	Agalychnis callidryas	Anura	LC
104001	Crustaceans	Acanthocyclops hypogeus	Cyclopoida	VU

Download CSV

Biological terms used in animal kind table

Officially, threatened species are those listed as Critically Endangered (CR), Endangered (EN) or Vulnerable (VU).
Practically this means:

- **Critically Endangered** (CR): A species facing an *extremely* high risk of extinction in the wild.
- **Endangered** (EN): A species considered to be facing a *very high* risk of extinction in the wild.
- **Vulnerable** (VU): A species considered to be facing a *high* risk of extinction in the wild.

```
create table animal(
  animal_id number primary key,
animal_name varchar(100),
cage_no number,
 height number,
weight number,
age number,
gender varchar(10),
 origin varchar(100),
animalk_id NUMBER,
FOREIGN KEY (animalk_id) REFERENCES
animal_kind(animalk_id),
 zoo_id number,   FOREIGN KEY(zoo_id) REFERENCES
zoo(zoo_id)
);
```

```
INSERT INTO animal VALUES (30001, ‘King Cobra', 101, 23, 1, 2, 'M', 'North america', 105001, 10009);
INSERT INTO animal VALUES (30002, 'Monkey', 102, 31, 15, 9, 'M', 'africa', 105002, 10009);
INSERT INTO animal VALUES (30004, 'Alligator', 104, 60, 19, 5, 'M ', 'india', 102001, 10003);
INSERT INTO animal VALUES (30005, 'Elephant', 105, 188, 430, 21, 'F', 'North america', 101001, 10003);
INSERT INTO animal VALUES (30006, 'Hyena', 106, 265, 11, 8, 'M', 'india', 104001, 10004);
INSERT INTO animal VALUES (30007, 'Ostrich', 107, 90, 7, 6, 'F', 'UK', 105001, 10004);
INSERT INTO animal VALUES (30008, 'Hippopotamus', 108, 305, 11, 5, 'M', 'india', 102001, 10007);
INSERT INTO animal VALUES (30009, 'Zebra', 109, 168, 43, 11, 'M', 'Australia', 105002, 10003);
INSERT INTO animal VALUES (30010, 'PeaCock', 110, 22, 23, 2, 'F', 'North america', 104001, 10009);
INSERT INTO animal VALUES (30011, 'White Tiger', 111, 130, 245, 7, 'M', 'india', 101001, 10003);
```

ANIMAL_ID	ANIMAL_NAME	CAGE_NO	HEIGHT	WEIGHT	AGE	GENDER	ORIGIN	ANIMALK_ID	ZOO_ID
30010	PeaCock	110	22	23	2	F	North america	104001	10009
30004	Alligator	104	60	19	5	M	india	102001	10003
30005	Elephant	105	188	430	21	F	North america	101001	10003
30006	Hyena	106	265	11	8	M	india	104001	10004
30007	Ostrich	107	90	7	6	F	UK	105001	10004
30008	Hippopotamus	108	305	11	5	M	india	102001	10007
30009	Zebra	109	168	43	11	M	Australia	105002	10003
30011	White Tiger	111	130	245	7	M	india	101001	10003
30001	Kinf Cobra	101	23	1	2	M	North america	105001	10009
30002	Monkey	102	31	15	9	M	africa	105002	10009

```
create table location(
city varchar(100) primary key,
 state varchar(100));

 INSERT INTO location VALUES ('Hyderabad', 'Telangana');
  INSERT INTO location VALUES ('Warangal', 'Telanagana');
 INSERT INTO location VALUES ('Visakhapatnam', 'Andharapradesh');
  INSERT INTO location VALUES ('Tirupathi', 'Andhara pradesh');
```

CITY	STATE
Hyderabad	Telangana
Warangal	Telanagana
Visakhapatnam	Andhara pradesh
Tirupathi	Andhara pradesh

```
create table pincode(   pinnum number primary key,
  street varchar(100),
city varchar(100),
 FOREIGN KEY (city) REFERENCES location(city));
```

```
INSERT INTO pincode VALUES (500002, 'Dilshuknagar', 'Hyderabad’);
INSERT INTO pincode VALUES (500016, 'Banjara Hills', 'Hyderabad’);
INSERT INTO pincode VALUES (500060, 'Vidyanagar', 'Hyderabad’);
 INSERT INTO pincode VALUES (500069, 'Miyapur', 'Hyderabad’);
INSERT INTO pincode VALUES (500125, 'Kukatpalli', 'Hyderabad’);
INSERT INTO pincode VALUES (600004, 'Kazipet', 'Warangal’);
 INSERT INTO pincode VALUES (600008, 'Subedari', 'Warangal’);
INSERT INTO pincode VALUES (600021, 'LB nagar', 'Warangal’);
INSERT INTO pincode VALUES (600035, 'Shambunipet', 'Warangal’);
INSERT INTO pincode VALUES (600055, 'Hanmakonda', 'Warangal');
```

PINNUM	STREET	CITY
500002	Dilshuknagar	Hyderabad
500016	Banjara Hills	Hyderabad
500060	Vidyanagar	Hyderabad
500069	Miyapur	Hyderabad
500125	Kukatpalli	Hyderabad
600004	Kazipet	Warangal
600008	Subedari	Warangal
600021	LB nagar	Warangal
600035	Shambunipet	Warangal
600055	Hanmakonda	Warangal

```
CREATE TABLE EMPLOYEE
(
  EMP_ID NUMBER PRIMARY KEY,
  EMP_FNAME VARCHAR(100),
  EMP_MNAME VARCHAR(100),
  EMP_LNAME VARCHAR(100),
  SALARY NUMBER,
  ZOO_ID NUMBER,
  FOREIGN KEY (ZOO_ID)REFERENCES ZOO(ZOO_ID),
  PINNUM NUMBER,
  FOREIGN KEY (PINNUM) REFERENCES PINCODE(PINNUM)
);
```

```
INSERT INTO EMPLOYEE VALUES (1001, 'RAGHU', 'PHANESH', 'SANITARY', 60000, 10004, 500001);
INSERT INTO EMPLOYEE VALUES (1002, 'SANKAR', 'KOLAPALI', 'SECURITY', 15000, 10009,500016);
INSERT INTO EMPLOYEE VALUES (1003, 'SAMVIDHA', 'JAARON', 'CAGEKEEPER', 15000, 10007, 600055);
INSERT INTO EMPLOYEE VALUES (1004, 'ROHITH', 'PINNAMRAJU', 'GATEKEEPER', 15000, 10004, 500125);
INSERT INTO EMPLOYEE VALUES (1005, 'NAVEEN', 'ALLU RATNA', 'CAGEKEEPER', 15000, 10003, 500125);
INSERT INTO EMPLOYEE VALUES (1006, 'VARUN', 'REDDY', 'CAGEKEEPER', 20000, 10007, 500125);
INSERT INTO EMPLOYEE VALUES (1007, 'RAJESH', 'AMARAGANI', 'CAGEKEEPER', 15000, 10009, 600021);
INSERT INTO EMPLOYEE VALUES (1008, 'WASEEM', 'AGARWAL', 'CAGEKEEPER', 15000, 10003, 500001);
INSERT INTO EMPLOYEE VALUES (1009, 'VINAY ', 'GUNDAPALLI', 'CAGEKEEPER', 20000, 10009, 600008);
INSERT INTO EMPLOYEE VALUES (1010, 'SHIVA REDDY', 'RAMALA', 'CAGEKEEPER', 20000, 10007, 600055);
```

EMP_ID	EMP_FNAME	EMP_MNAME	EMP_LNAME	SALARY	ZOO_ID	PINNUM
1002	Sankar	kolapali	security	15000	10009	500016
1003	Samvidha	jaaron	cagekeeper	15000	10007	600055
1004	Rohith	pinnamraju	gatekeeper	15000	10004	500125
1005	naveen	allu ratna	cagekeeper	15000	10003	500125
1006	varun	reddy	cagekeeper	20000	10007	500125
1007	rajesh	Amaragani	cagekeeper	15000	10009	600021
1009	vinay	Gundapalli	cagekeeper	20000	10009	600008
1010	shiva reddy	ramala	cagekeeper	20000	10007	600055

```
create table contact(
 emp_id number,
FOREIGN KEY (emp_id) REFERENCES employee(emp_id),   phone_no
number);


INSERT INTO contact VALUES (1001, 8741122565);
INSERT INTO contact VALUES (1002, 6179485234);
INSERT INTO contact VALUES (1003, 7849562134);
INSERT INTO contact VALUES (1004, 9844565225);
INSERT INTO contact VALUES (1005, 9848522338);
INSERT INTO contact VALUES (1006, 6320154879);
INSERT INTO contact VALUES (1007, 8484879111);
INSERT INTO contact VALUES (1008, 9787488845);
INSERT INTO contact VALUES (1009, 9784684135);
INSERT INTO contact VALUES (1010, 7454846513);
```

EMP_ID	PHONE_NO
1002	6179485234
1003	7849562134
1004	9844565225
1005	9848522338
1006	6320154879
1007	8484879111
1009	9784684135
1010	7454846513

```
create table visitor(    v_id number PRIMARY KEY,
 phone_no number,
pinnum number,   foreign key (pinnum) references pincode(pinnum),
v_fname varchar(100),
v_lname varchar(100)
);

INSERT INTO visitor VALUES (1000002, 8247423616, 500001, 'Sandhya', 'Dhanavath’);
 INSERT INTO visitor VALUES (1000003, 9848522338, 500002, 'Shankar', 'Tejavath’);
 INSERT INTO visitor VALUES (1000004, 7532148967, 500060, 'Waseem', 'Syed’ );
 INSERT INTO visitor VALUES (1000005, 6459783120, 500125, 'Manoj', 'Boganadham’);
 INSERT INTO visitor VALUES (1000006, 8524615397, 500069, 'Infi', 'Chan’);
 INSERT INTO visitor VALUES (1000007, 9754125896, 600004, 'Bhushank', 'Kul’);
 INSERT INTO visitor VALUES (1000008, 8462157930, 600055, 'Abhiram', 'Nallama’);
INSERT INTO visitor VALUES (1000009, 6841759325, 600008, 'Ashish', 'Anand’);
 INSERT INTO visitor VALUES (1000010, 8945632178, 600154, 'Lakshita', 'Chowdary’);
 INSERT INTO visitor VALUES (1000011, 9685741232, 600035, 'Nayan', 'Jyothi’);
INSERT INTO visitor VALUES (1000012, 8675941236, 600021, 'Ranil', 'Bala’);
INSERT INTO visitor VALUES (1000013, 7849562134, 600154, 'Tanisha', 'Agarwal');
```

V_ID	PHONE_NO	PINNUM	V_FNAME	V_LNAME
1000003	9848522338	500002	Shankar	Tejavath
1000004	7532148967	500060	Waseem	Syed
1000005	6459783120	500125	Manoj	Boganadham
1000006	8524615397	500069	Infi	Chan
1000007	9754125896	600004	Bhushank	Kul
1000008	8462157930	600055	Abhiram	Nallama
1000009	6841759325	600008	Ashish	Anand
1000011	9685741232	600035	Nayan	Jyothi
1000012	8675941236	600021	Ranil	bala

```
create table purchase(
  purchase_id number primary key,    purchase_name
varchar(100));

insert into purchase values(102,'Credit Card');
insert into purchase values(103,'Cash');
insert into purchase values(104,'UPI');
insert into purchase values(105,'e-wallets');
```

PURCHASE_ID	PURCHASE_NAME
102	Credit Card
103	Cash
104	UPI
105	e-Wallets

```
create table ticket(
 ticket_id number primary key,
  ticket_date date,
 v_id number,
   FOREIGN KEY (v_id) REFERENCES visitor(v_id),
   purchase_id number,
  FOREIGN KEY (purchase_id) REFERENCES purchase(purchase_id),
 zoo_id number,    FOREIGN KEY (zoo_id) REFERENCES zoo(zoo_id));
```

```
INSERT INTO ticket VALUES (9034351, '15-08-2020', 1000002, 104, 10004);
INSERT INTO ticket VALUES (2110003, '15-02-2020', 1000003, 104, 10007);I
NSERT INTO ticket VALUES (6382682, '15-02-2020', 1000004, 103, 10003);I
NSERT INTO ticket VALUES (6824217, '14-02-2020', 1000005, 102, 10007);
INSERT INTO ticket VALUES (5193139, '15-02-2020', 1000006, 103, 10009);
INSERT INTO ticket VALUES (5542291, '14-02-2020', 1000007, 102, 10003);
INSERT INTO ticket VALUES (2580752, '14-02-2020', 1000008, 104, 10007);
INSERT INTO ticket VALUES (9154961, '15-02-2020', 1000009, 102, 10007);
INSERT INTO ticket VALUES (8391607, '14-02-2020', 1000010, 101, 10007);
INSERT INTO ticket VALUES (1329791, '14-02-2020', 1000011, 102, 10007);
```

TICKET_ID	TICKET_DATE	V_ID	PURCHASE_ID	ZOO_ID
2110003	15-02-2020	1000003	104	10007
6382682	15-02-2020	1000004	103	10003
6824217	14-02-2020	1000005	102	10007
5193139	15-02-2020	1000006	103	10009
5542291	14-02-2020	1000007	102	10003
2580752	14-02-2020	1000008	104	10007
9154961	15-02-2020	1000009	102	10007
1329791	14-02-2020	1000011	102	10007

```
create table looks_after(
  animal_id number,
FOREIGN kEY (animal_id) REFERENCES animal(animal_id),     emp_id number,
 FOREIGN KEY (emp_id) REFERENCES employee(emp_id));
```

```
INSERT INTO looks_after(animal_id, emp_id) VALUES (30010, 1002);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30004, 1003);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30005, 1004);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30006, 1005);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30007, 1006);
 INSERT INTO looks_after(animal_id, emp_id) VALUES (30008, 1007);
 INSERT INTO looks_after(animal_id, emp_id) VALUES (30009, 1000);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30011, 1010);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30001, 1009);
INSERT INTO looks_after(animal_id, emp_id) VALUES (30002, 1010);
```

ANIMAL_ID	EMP_ID
30004	1005
30009	1005
30010	1002
30004	1003
30005	1004
30006	1005
30007	1006
30008	1007
30011	1010
30001	1009
30002	1010

```
create table visits(
  ticket_id number PRIMARY KEY,
 FOREIGN KEY (ticket_id) REFERENCES ticket(ticket_id),     in_time TIMESTAMP,
   out_time TIMESTAMP);
```

```
INSERT INTO visits VALUES (1329791, to_timestamp('16/02/2020 10:53:10', 'dd/mm/yyyy
HH24:MI:SS'),to_timestamp( '16/02/2020 16:53:15','dd/mm/yyyy HH24:MI:SS'));INSERT INTO visits
VALUES (9154961, to_timestamp('2020-02-16 10:53:45','yyyy/mm/dd HH24:MI:SS'),
to_timestamp('2020-02-16 16:07:41','yyyy/mm/dd HH24:MI:SS'));INSERT INTO visits VALUES
(5542291, TO_TIMESTAMP('2020/02/16 10:45:55','yyyy/mm/dd HH24:MI:SS'),
TO_TIMESTAMP('2020/02/16 16:05:09','yyyy/mm/dd HH24:MI:SS'));INSERT INTO visits VALUES
(5193139, TO_TIMESTAMP('2020/02/16 10:57:30','yyyy/mm/dd HH24:MI:SS'),
TO_TIMESTAMP('2020/02/16 16:07:11','yyyy/mm/dd HH24:MI:SS'));INSERT INTO visits VALUES
(6824217, TO_TIMESTAMP('2020/02/16 10:59:37','yyyy/mm/dd HH24:MI:SS'),
TO_TIMESTAMP('2020/02/16 16:49:04','yyyy/mm/dd HH24:MI:SS'));INSERT INTO visits VALUES
(2110003, TO_TIMESTAMP('2020/02/16 10:35:55','yyyy/mm/dd HH24:MI:SS'),
TO_TIMESTAMP('2020/02/16 16:39:35','yyyy/mm/dd HH24:MI:SS'));INSERT INTO visits VALUES
(6382682, TO_TIMESTAMP('2020/02/16 10:19:33','yyyy/mm/dd HH24:MI:SS'),
TO_TIMESTAMP('2020/02/16 16:37:00','yyyy/mm/dd HH24:MI:SS'));
```

TICKET_ID	IN_TIME	OUT_TIME
6824217	16-FEB-20 10.59.37.000000 AM	16-FEB-20 04.49.04.000000 PM
1329791	16-FEB-20 10.53.10.000000 AM	16-FEB-20 04.53.15.000000 PM
9154961	16-FEB-20 10.53.45.000000 AM	16-FEB-20 04.07.41.000000 PM
5542291	16-FEB-20 10.45.55.000000 AM	16-FEB-20 04.05.09.000000 PM
5193139	16-FEB-20 10.57.30.000000 AM	16-FEB-20 04.07.11.000000 PM
2110003	16-FEB-20 10.35.55.000000 AM	16-FEB-20 04.39.35.000000 PM
6382682	16-FEB-20 10.19.33.000000 AM	16-FEB-20 04.37.00.000000 PM

Sources we used :

http://www.javatpoint.com/

http://www.w3schools.com/

http://www.wikipedia.org/ (for some biology related information)

Thank you