



Zoo Management System

Ivan Cebreros

Student ID: 918174600

Github ID: icebreros

Milestone/Version	Date
M2V1	Apr 17, 2022
M1V1	Mar 15, 2022

Table of Contents

Section I: Project Description	2
Section II: Use Cases	3
Use Case 1	3
Use Case 2	3
Use Case 3	4
Use Case 4	4
Use Case 5	5
Use Case 6	6
Section III: Database Requirements (Business Rules)	7
Section IV: Detailed List of Main Entities, Attributes and Keys	9
Section V: Entity Relationship Diagram (ERD)	11
Section VI: Testing Table	12
Milestone 2	13
Revised ERD	13
Section VII: Database Model/EER	14
Section XI: Testing Table	17

Section I: Project Description

The purpose of this project is to create a management system for a Zoo. A Zoo should be a place where animals are kept in sections, and visitors can enter the Zoo and spectate the animals. The Zoo can house many different types of animals, so it would be great to keep track of the animals with a database that can easily manage them. Without keeping track of this information, maintaining a zoo and all its inhabitants would be extremely difficult. For staff, it's also important that staff knows what pens are empty, and which pens house what type of animal. Visitors might want to know the location of where a specific type of animal is within the Zoo. All of this data (and more) will be managed within this project.

Section II: Use Cases

Use Case 1

Use Case Title	Difficulty finding animals
Actors	Sally (visitor)
Description	<p>Sally is so excited to be at the zoo today, but she always runs into issues finding the animals in the Zoo. Since pens and sections change, the layout of the Zoo does as well. This can be particularly frustrating for both the visitor and the staff working there, as there is no clear and consistent system that lets them easily keep track of where the animals are.</p> <p>The Zoo Management system is here to help by providing the information of the pens and animals, along with their location within the Zoo. With this system, visitors like Sally can use the digital directories and look up animals by name, which will show where they are located in the zoo as well. This way, both the visitor and staff can keep better track of the animals, and would allow the staff to make on the fly changes when necessary.</p>

Use Case 2

Use Case Title	Check/Maintain Pens
Actors	Josh (staff)
Description	<p>Josh is a staff member at the Zoo. Every night he has to check each pen, confirming the animals within the pen and making updates if the pen is empty. This proves to be difficult as Josh runs into many issues because there are many inconsistencies with the animals and pens.</p>

	<p>The Zoo Management system can provide Josh with the appropriate tools to properly monitor and make changes to each pen. With this, other staff can also provide real time updates when changes occur, therefore reducing Josh's workload because updates can happen frequently and are easy to monitor and manage.</p>
--	---

Use Case 3

Use Case Title	Changing Ticket Prices
Actors	Mark (Manager)
Description	<p>Mark is the manager for the Zoo. On certain days the Zoo offers a lower ticket price. However, this proves to be a difficult change as Mark has to go to each entrance where the Ticket Terminals are located, and manually change each ticket price on the wall. Since he is the only one that can change the ticket price, it forces him to stop what he's doing and go around the entire Zoo to make these changes.</p> <p>With the Zoo Management system, this change is easy and can be done on the fly. These changes are easily reflected onto the ZooTicket terminal, and this change is something that can be done on a device from anywhere. This proves to be much easier to manage, and allows not only himself but his staff to be more efficient, and solve their management issues more easily and quickly.</p>

Use Case 4

Use Case Title	Checking inventory at gift shop
Actors	Andrew (visitor/buyer), Sarah (staff)
Description	<p>Andrew is a frequent visitor of the Zoo, as it's one of his favorite places to go. However, every time he visits the gift shop,</p>

	<p>there is always a problem involving the stock of a current item. In this case, he was looking for a stuffed Gorilla. The main issue that forces Andrew to come another time is the time it takes and the difficulty of finding and confirming the availability of a specific item when it's not in the gift shop itself.</p> <p>The Zoo Management system takes care of these aspects, as inventory is managed and kept track of. Now, Andrew can simply ask a staff member if there are any Gorillas in the back, in which the employee can look up the item by name, and confirm not only its quantity but its location in the storage room to quickly and easily get Andrew his gift. This is useful to everyone as it allows the staff to be more efficient, allows customers an easier path of purchase, and reduces customer issues with long lines and long service times.</p>
--	--

Use Case 5

Use Case Title	Incorrect location for animal
Actors	Trevor (visitor), staff
Description	<p>Trevor is a visitor at the Zoo. Although excited to see all the Zoo has to offer, his first mission involves seeing a Zebra. However, the main map shows outdated locations for animals, so Trevor takes a trip to the wrong location and the Zebra is nowhere to be found. Trevor is not only confused, but also very disappointed.</p> <p>With the Zoo Management System, Trevor can easily look up the location for the Zebra. And if the location is still incorrect, staff can use the management system to easily change the location of the Zebra to the correct one, and those changes will reflect across the entire system, for all digital directories. This allows staff to quickly fix mistakes and let those fixes reflect across the entire Zoo. And thankfully</p>

	because Trevor mentioned this issue, other visitors will now get the correct location of the Zebra, and not face the same confusion and disappointment Trevor faced.
--	--

Use Case 6

Use Case Title	Inefficient rewards program
Actors	Shilo (visitor), Zoo, ZooCard
Description	<p>Shilo is a long time visitor of the Zoo. The Zoo has a rewards program that allows visitors to submit a limited number of tickets from ZooStops for a return ticket, in which they can pool together and use to redeem rewards. She wishes the Zoo's rewards program was safer, easier to use, and more efficient. Shilo also has an issue with tickets not worthwhile for visitors to cash them at the end of the day. Not only carrying the tickets, but also the return ticket from their trip.</p> <p>The Zoo Management system is back again to provide visitors with their ZooCard. ZooCard allows users to simply tap their card on ZooStop and earn rewards on the spot. No waiting in lines or handling handfuls of tickets. This improves not only the efficiency and survival of the rewards system, but also results in less handling, cutting back on tickets, and providing a safer experience to the visitor.</p>

Section III: Database Requirements (Business Rules)

- 1. User**
 - 1.1. A user shall create only one account
- 2. Account**
 - 2.1. An account shall be created by only one user
 - 2.2. An account shall have at least one role
- 3. Admin**
 - 3.1. An admin is a user
 - 3.2. An admin is the only and only user that can edit roles
 - 3.3. An admin is the only and only user that can add new roles
 - 3.4. An admin is the only and only user that can delete roles
 - 3.5. An admin is the only and only user that can change ticket prices
- 4. Staff**
 - 4.1. A staff member can be an admin
 - 4.2. A staff member is the only user that can update animal locations
 - 4.3. A staff member is the only user that can update animal pens
 - 4.4. A staff member is the only user that can update locations
 - 4.5. A staff member is the only user that can update pens
 - 4.6. A staff member is the only user that can update items
- 5. Ticket**
 - 5.1. A ticket shall have a price
- 6. Role**
 - 6.1. A role shall be used by many accounts
 - 6.2. A role shall have a type
- 7. Animal**
 - 7.1. An animal shall have at least one pen
 - 7.2. An animal shall have at least one location
- 8. Pen**
 - 8.1. A pen can have many animals
 - 8.2. A pen can be empty
- 9. Location**
 - 9.1. A location shall have many pens
 - 9.2. A location shall have many animals
- 10. Item**
 - 10.1. An item shall have a name
 - 10.2. An item shall have a price
 - 10.3. An item shall have a quantity



10.4. An item shall have a location

11. ZooCard

11.1. A ZooCard shall have a balance

11.2. A ZooCard is a user

12. ZooStop

12.1. A ZooStop shall have a reward amount

12.2. A ZooStop can be accessed by an admin

13. ZooTicket

13.1. A ZooTicket has a display price

13.2. A ZooTicket creates a ticket

13.3. A ZooTicket can be accessed by an admin

Section IV: Detailed List of Main Entities, Attributes and Keys

- 1. User (Strong)**
 - a. user_id: key, numeric
 - b. account_id: key, numeric
 - c. full_name: composite, alphanumeric
 - d. email: alphanumeric
 - e. dob: multivalue, timestamp
- 2. Account (Strong)**
 - a. account_id: key, numeric
 - b. user_id: key, numeric
 - c. role: alphanumeric
- 3. Admin (Weak)**
 - a. admin_id: key, numeric
 - b. user_id: key, numeric
- 4. Staff (Weak)**
 - a. staff_id: key, numeric
 - b. user_id: key, numeric
- 5. Ticket (Strong)**
 - a. ticket_id: key, numeric
 - b. price: numeric
- 6. Role (Strong)**
 - a. role_id: key, numeric
 - b. type: alphanumeric
 - c. description: alphanumeric
- 7. Animal (Strong)**
 - a. animal_id: key, numeric
 - b. type: alphanumeric
 - c. pen_id: key, numeric
- 8. Pen (Strong)**
 - a. pen_id: key, numeric
 - b. is_empty: boolean
 - c. num_of_inhabitants: numeric
- 9. Location (Strong)**
 - a. location_id: key, numeric
 - b. pen_id: key, numeric

- c. map_location: alphanumeric

10. Item (Strong)

- a. item_id: key, numeric
- b. name: alphanumeric
- c. price: numeric
- d. quantity: numeric
- e. location: alphanumeric

11. ZooCard (Strong)

- a. zoocard_id: key, numeric
- b. User_id: key, numeric

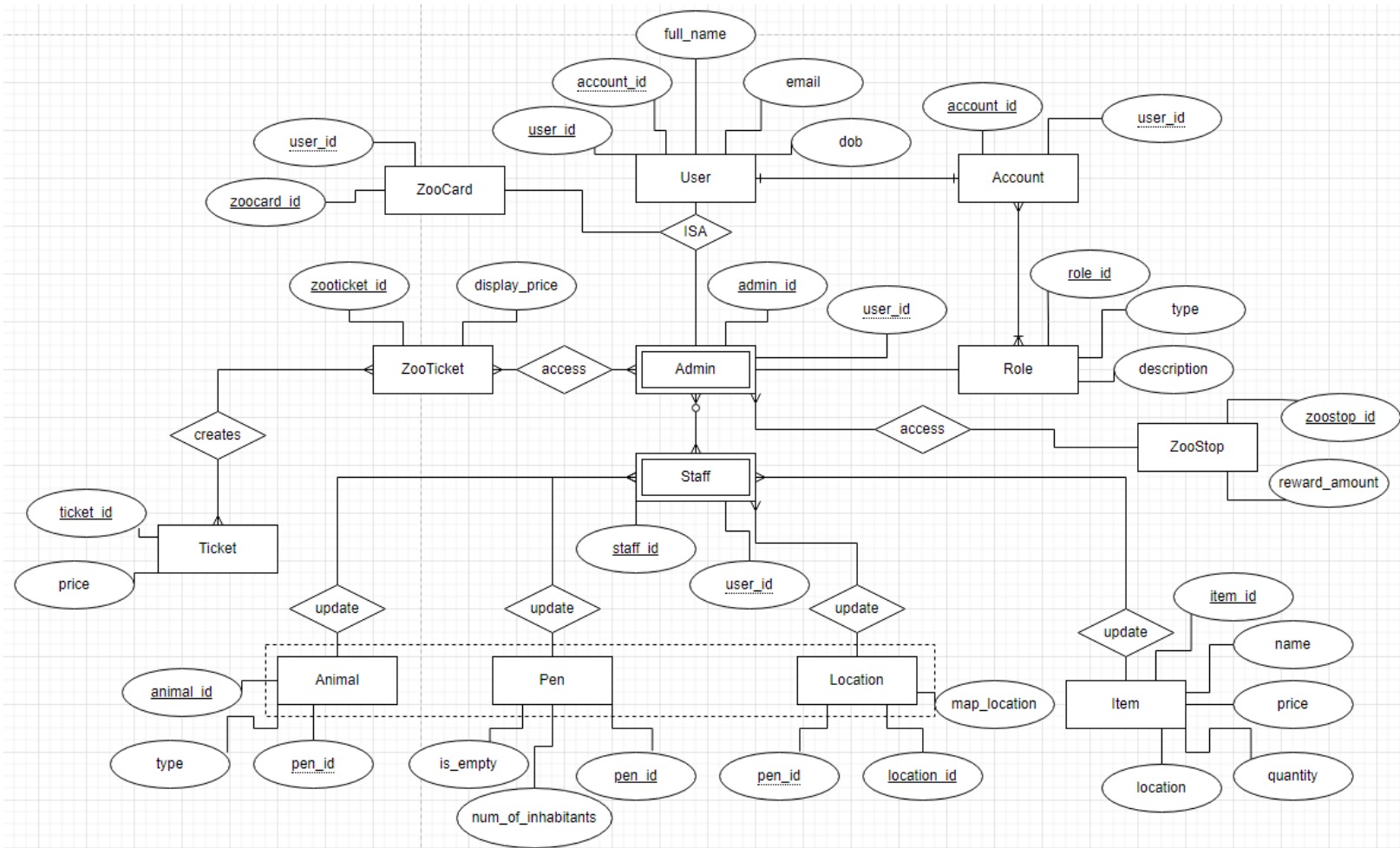
12. ZooStop (Strong)

- a. zoostop_id: key, numeric
- b. reward_amount: numeric

13. ZooTicket (Strong)

- a. zooticket_id: key, numeric
- b. display_price: numeric

Section V: Entity Relationship Diagram (ERD)

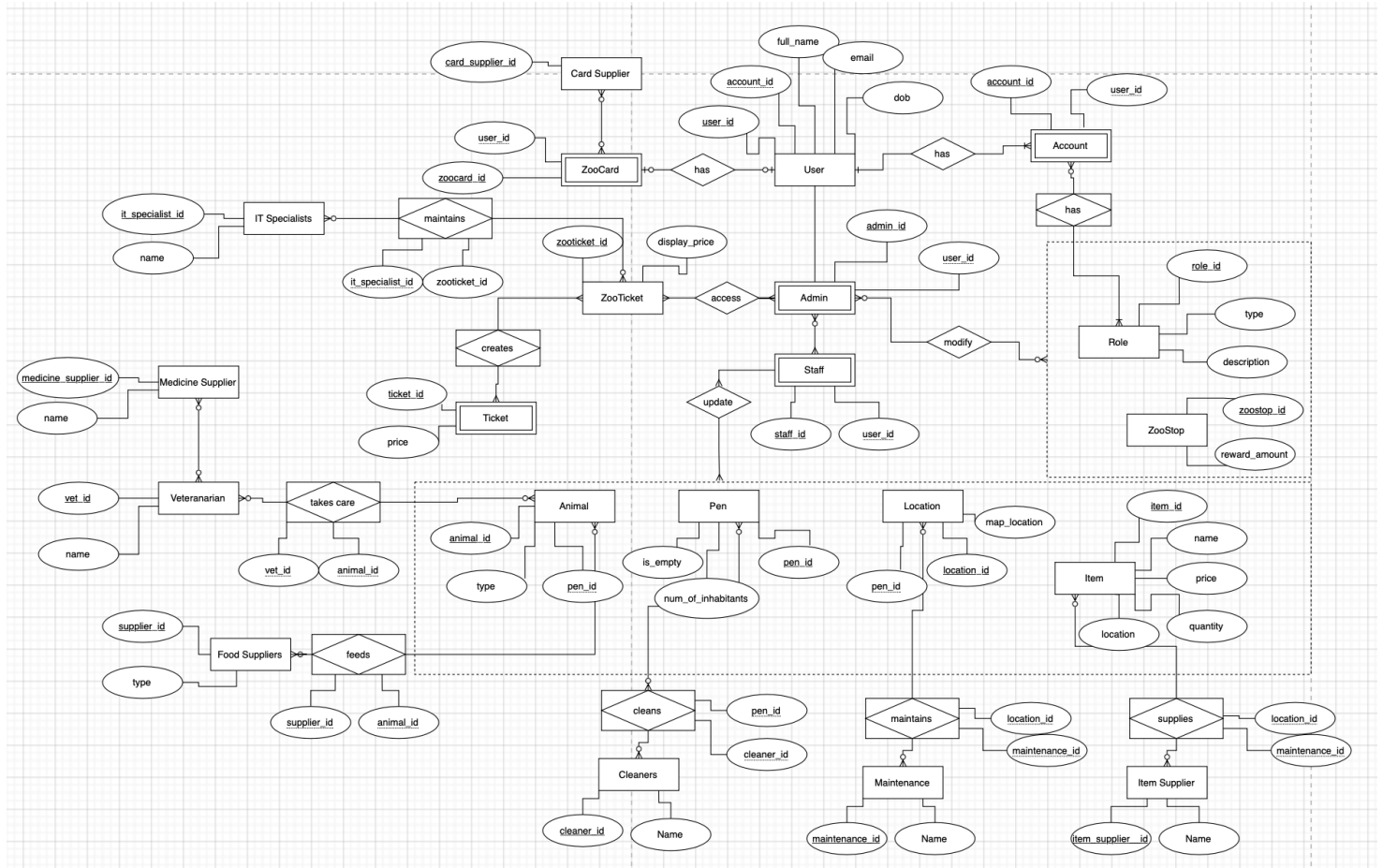


Section VI: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	User	create	Account	1-to-1	Fail	Many users can create an account with different email
2	Account	has	Role	1-to-M	Pass	None
3	Admin	ISA	User	1-to-1	Fail	Many user can be admin but admin must be a user
4	Staff	Can be	Admin	M-to-1	Pass	None
5	Staff	update	Animal	M-to-M	Pass	None
6	Staff	update	Pen	M-to-M	Pass	None
7	Staff	update	Item	M-to-M	Pass	None
8	Animal	has	Pen	1-to-1	Pass	None
9	Animal	has	Location	1-to-1	Pass	None
10	Pen	has	Animal	M-to-M	Fail	Pens can be empty
11	Location	has	Pen	1-to-M	Pass	None
12	Location	has	Animal	1-to-M	Fail	Pens can be empty within locations
13	ZooCard	ISA	User	1-to-1	Pass	None
14	ZooStop	Accessed by	Admin	1-to-M	Pass	None
15	ZooTicket	Accessed by	Admin	1-to-M	Pass	None

Milestone 2

Revised ERD



Section VII: Database Model/EER

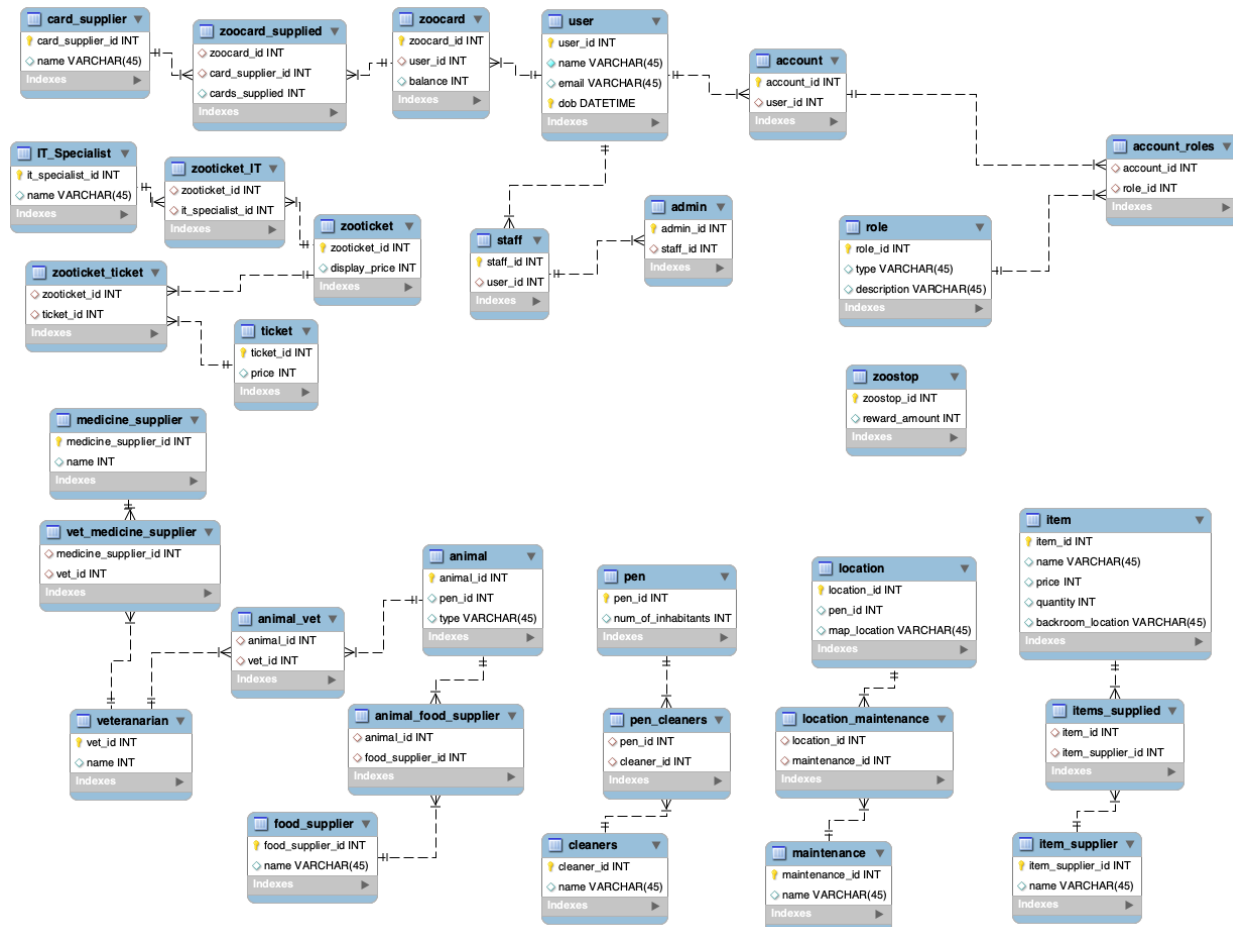


Table	FK	ON DELETE	ON UPDATE	COMMENT
account	user_id	ON CASCADE	ON CASCADE	If a user is deleted, then the account from that user must be deleted as well
staff	user_id	ON CASCADE	ON CASCADE	If a user is deleted, then the staff relation to user must be deleted as well
admin	staff_id	ON CASCADE	ON CASCADE	If a staff is deleted, then the admin associated with that staff no longer exists

zoocard	user_id	ON CASCADE	ON CASCADE	If a user is deleted, then the zoocard associated with that user must be deleted
zooticket_IT	zooticket_id	ON CASCADE	ON CASCADE	If a zooticket is deleted, then the associating IT Specialist must be removed from that specific zooticket
zooticket_IT	it_specialist_id	ON CASCADE	ON CASCADE	If a IT Specialist is deleted, then the associating zooticket must be removed from that specific specialist
account_roles	account_id	ON CASCADE	ON CASCADE	If an account is deleted, then they must be removed from association to their respective role
account_roles	role_id	ON CASCADE	ON CASCADE	If a role is deleted, then all accounts with that role must be removed as well
animal_food_supplier	animal_id	ON CASCADE	ON CASCADE	If an animal is deleted, then it no longer is assigned a food supplier
animal_food_supplier	food_supplier_id	ON CASCADE	ON CASCADE	If a food supplier is deleted, then the associated animal no longer has a supplier
pen_cleaners	pen_id	ON CASCADE	ON CASCADE	If a pen is deleted, then the associating cleaner no longer has to clean their assigned pen
pen_cleaners	cleaner_id	ON CASCADE	ON CASCADE	If a cleaner is deleted, then the associated pen no longer has a cleaner
location_maintenance	location_id	ON CASCADE	ON CASCADE	If a location is deleted, then the associating maintenance no longer maintains that location
location_maintenance	maintenance_id	ON CASCADE	ON CASCADE	If a maintenance is deleted, then associating locations are no longer maintained
items_supplied	item_id	ON CASCADE	ON CASCADE	If an item is deleted, then it is no longer being supplied by the respective supplier

items_supplied	item_supplier_id	ON CASCADE	ON CASCADE	If a supplier is deleted, then the respective item they were supplying must also be removed
animal_vet	animal_id	ON CASCADE	ON CASCADE	If a animal is deleted, then the associating vet no longer is assigned to take care of the animal
animal_vet	vet_id	ON CASCADE	ON CASCADE	If a vet is deleted, then the associating animal no longer has a caretaker
vet_medicine_supplier	vet_id	ON CASCADE	ON CASCADE	If a vet is deleted, then it no longer requires a supplier
vet_medicine_supplier	medicine_supplier_id	ON CASCADE	ON CASCADE	If a supplier is deleted, then the associating vet no longer has medicine being supplied
zooticket_ticket	zooticket_id	ON CASCADE	ON CASCADE	If a zooticket is deleted, then respective tickets from that zooticket must be deleted
zooticket_ticket	ticket_id	ON CASCADE	ON CASCADE	If a ticket is deleted, then it no longer remains for the respective zooticket
zoocard_supplied	zoocard_id	ON CASCADE	ON CASCADE	If a zoocard is deleted, then the supplier to the zoocard is also removed
zoocard_supplied	card_supplier_id	ON CASCADE	ON CASCADE	If a card supplier is deleted, then the respective zoocards supplied must also be removed

Section XI: Testing Table

Entity	SQL Query	Pass/ Fail	Error Description	Possible Solution
user	Delete	Pass	None	None
user	Update	Pass	None	None
account	Delete	Pass	None	None
account	Update	Pass	None	None
role	Delete	Pass	None	None
role	Update	Pass	None	None
zoostop	Delete	Pass	None	None
zoostop	Update	Pass	None	None
staff	Delete	Pass	None	None
staff	Update	Pass	None	None
admin	Delete	Pass	None	None
admin	Update	Pass	None	None
IT_Specialist	Delete	Pass	None	None
IT_Specialist	Update	Pass	None	None
zoocard	Delete	Pass	None	None
zoocard	Update	Pass	None	None
card_supplier	Delete	Pass	None	None
card_supplier	Update	Pass	None	None
zooticket	Delete	Pass	None	None
zooticket	Update	Pass	None	None
medicine_supplie	Delete	Pass	None	None

r				
medicine_supplier	Update	Pass	None	None
veterinarian	Delete	Pass	None	None
veterinarian	Update	Pass	None	None
food_supplier	Delete	Pass	None	None
food_supplier	Update	Pass	None	None
animal	Delete	Pass	None	None
animal	Update	Pass	None	None
pen	Delete	Pass	None	None
pen	Update	Pass	None	None
location	Delete	Pass	None	None
location	Update	Pass	None	None
cleaners	Delete	Pass	None	None
cleaners	Update	Pass	None	None
maintenance	Delete	Pass	None	None
maintenance	Update	Pass	None	None
item_supplier	Delete	Pass	None	None
item_supplier	Update	Pass	None	None
item	Delete	Pass	None	None
item	Update	Pass	None	None
zooticket_IT	Delete	Pass	None	None
zooticket_IT	Update	Pass	None	None
account_roles	Delete	Pass	None	None
account_roles	Update	Pass	None	None
animal_food_supplier	Delete	Pass	None	None

animal_food_supplier	Update	Pass	None	None
pen_cleaners	Delete	Pass	None	None
location_maintenance	Update	Pass	None	None
items_supplied	Delete	Pass	None	None
items_supplied	Update	Pass	None	None
animal_vet	Delete	Pass	None	None
animal_vet	Update	Pass	None	None
vet_medicine_supplier	Delete	Pass	None	None
vet_medicine_supplier	Update	Pass	None	None
zooticket_ticket	Delete	Pass	None	None
zooticket_ticket	Update	Pass	None	None
zoocard_supplied	Delete	Pass	None	None
zoocard_supplied	Update	Pass	None	None