

Problem- 2:

I visited the website <https://parks.novascotia.ca/> and found appropriate information on Nova Scotia Parks.

Mini-World: Parks in Nova Scotia

Identifying Entites: (Show strong and weak entity)

Entities	Reason						
1. Park <table><tr><td>Sub-Type entities of Park</td></tr><tr><td>1. Boat Launch</td></tr><tr><td>2. Beach</td></tr><tr><td>3. Hiking Trail</td></tr><tr><td>4. Camping</td></tr><tr><td>5. Day Use</td></tr></table>	Sub-Type entities of Park	1. Boat Launch	2. Beach	3. Hiking Trail	4. Camping	5. Day Use	Park is a strong entity because it has a key attribute. And it's a real-world object.
Sub-Type entities of Park							
1. Boat Launch							
2. Beach							
3. Hiking Trail							
4. Camping							
5. Day Use							
2. Visitor	Visitor is the strong entity because it does not depend on any other entity. And it's a real-world object.						
3. Event <table><tr><td>Sub-Type entities of Event</td></tr><tr><td>1. Public Event</td></tr><tr><td>2. Private Event</td></tr></table>	Sub-Type entities of Event	1. Public Event	2. Private Event	Event is a strong entity because it does not depend on any entity. It has its own primary key. It's a real-world object characteristics.			
Sub-Type entities of Event							
1. Public Event							
2. Private Event							
4. Facilities	Facilities is a strong entity as it does not depends upon any entity. It has its own primary key						
5. Park Advisories	Park advisories is the real-world object and is a strong entity because it does not depend on any entity. And it has its own primary key.						
6. Campsite	Campsite is a weak entity as it depends on Park. Also, it's a real-world object but does not have its own primary key.						
7. Reservations	Reservations is a weak entity as it depends on another entity. It does not has its own key attribute.						
8. Newsletter	Newsletter is a weak entity as it depends on visitor. It is a real-world object characteristics.						
9. Equipments	Equipments is a weak entity. It also is a real-world object that depends on the visitor entity						
10. Camp Program	Camp program is a weak entity that depends on the visiors that apply for camp program. Also it's a real- world Object.						
11. Vendor <table><tr><td>Sub-Type entities of Vendor</td></tr><tr><td>1. Food</td></tr><tr><td>2. Non-food</td></tr></table>	Sub-Type entities of Vendor	1. Food	2. Non-food	Vendor is a strong entity that does not depend on any other entity. It's a real-world object that has a key attribute.			
Sub-Type entities of Vendor							
1. Food							
2. Non-food							
12. Allowed_items	Allowed items is a weak entity as it depends on the park and visitor. Because the park permits just the allowed items that will be brought by the visitor/users.						
13. Job application	Job is a strong entity as it has its key attribute and it does not depend on any other entity.						

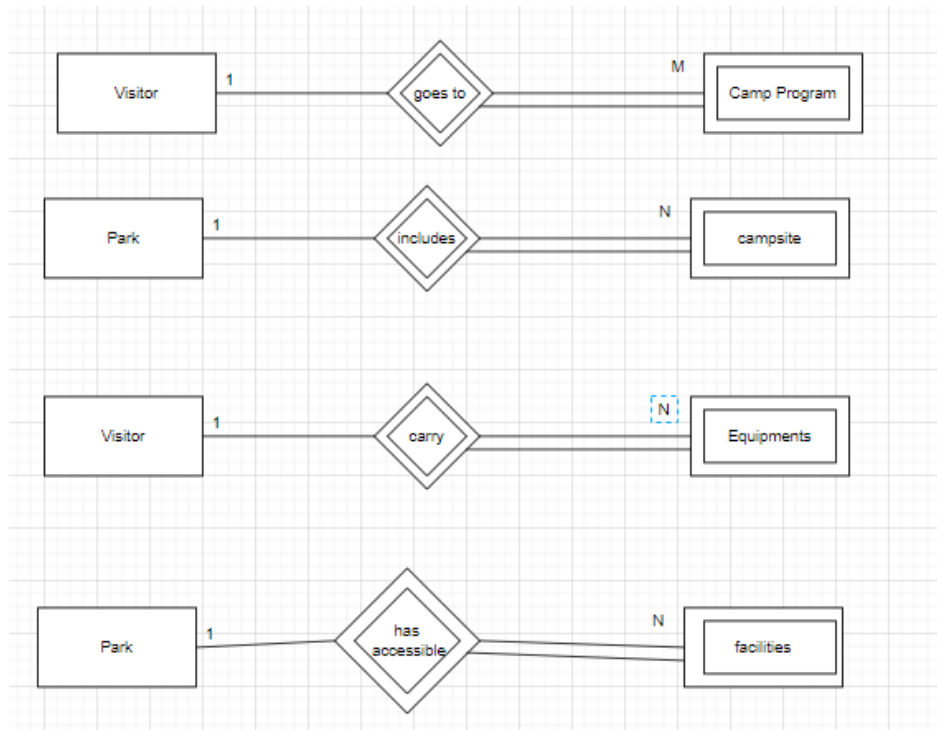
Identifying its attributes:

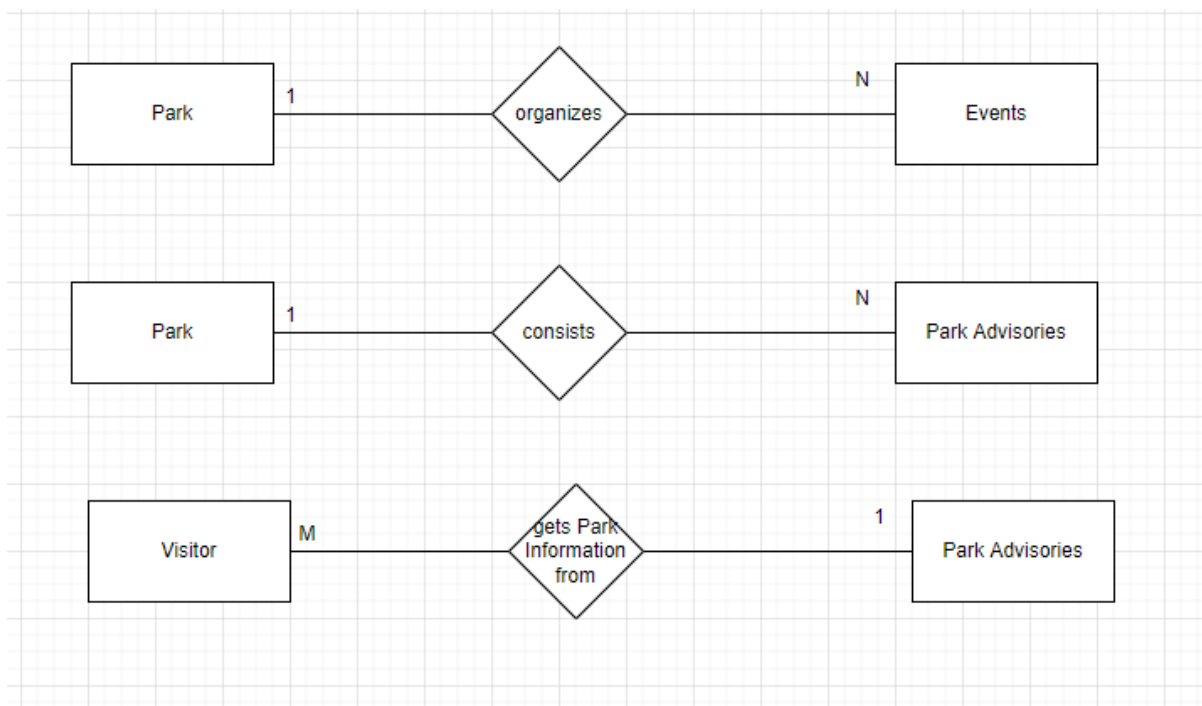
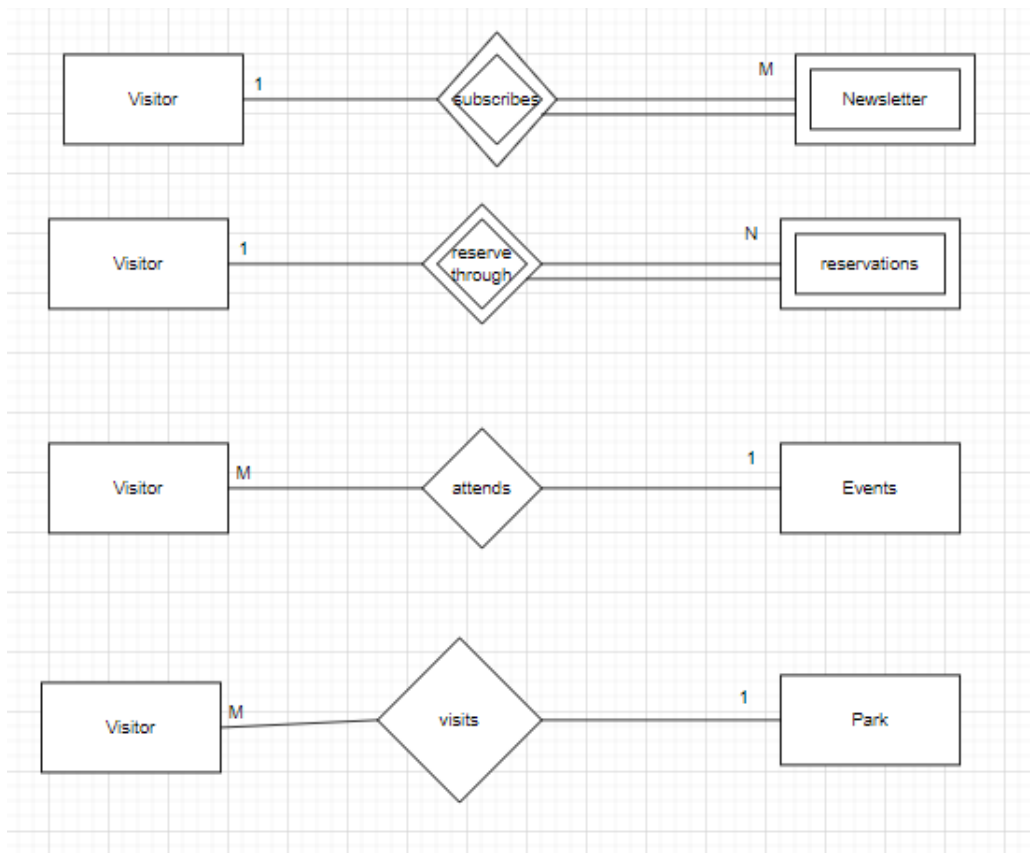
Entities	Attributes
1. Park	Park_id, check_in, check_out, availability, city, park_id, location
2. Visitor	V_id, v_age, v_phone, v_name
3. Event	E_id, E_timings, E_list, E_fees
4. Facilities	F_id, indoor, outdoor, f_name, no._of_facilities
5. Park Advisories	PA_id, PA_name, PA_type, PA_contact
6. Campsite	Capacit, duration, location, campsite_name, no. of days
7. Reservations	Reservation_fee, Reservation_id, Park_name, Arrival_date, Departure_date, Party_size
8. Newsletter	Newsletter_type, newsletter_name
9. Equipments	Equipment_type, amount_of equipments, Equipment_owner_name
10. Camp Program	Cp_name, duration, camp_fees
11. Vendor	vendor_id, vendor_emailid, vendor_address
12. Allowed_items	i_name, item_quantity, i_charge
13. Park_event_history	Record_name, record_num, record_id
14. Job Applications	Application_id, requirement, application_deadline

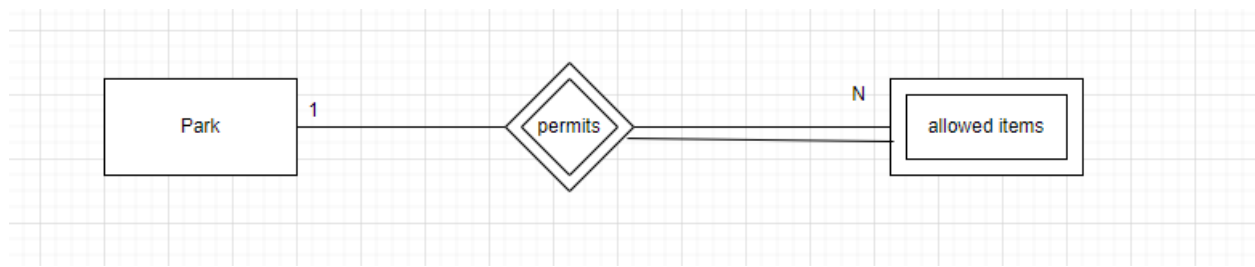
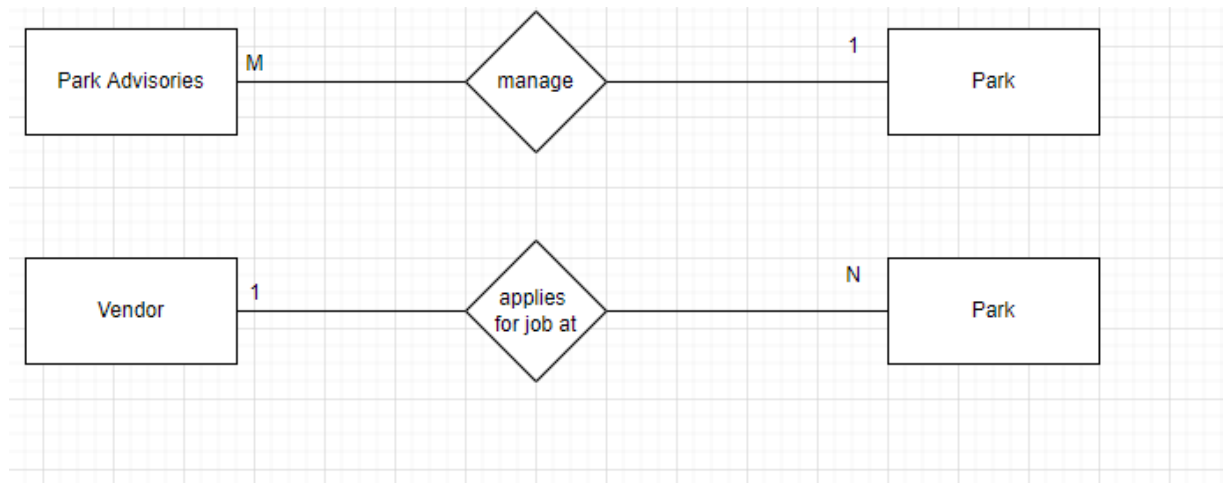
Sub-entities and their attributes:

Sub-entities	Attributes
1. Beach	B_name, Beach_id
2. Boat Launch	Bl_name, weather
3. Camping	area_size, Camp_id
4. Day use	Distance, num_of_trails, outdoor_opportunities, Du_id, Du_name
5. Hiking Trails	Wildlife, ht_id
6. Public Event	Reg_id, event_rules, capacity
7. Private Event	Pr_reg_id, event_contact, event_booking_fee
8. Food	Food_App_id, operational_time, menu, pricing, application_score
9. Non-food	Non_food_id, contract period

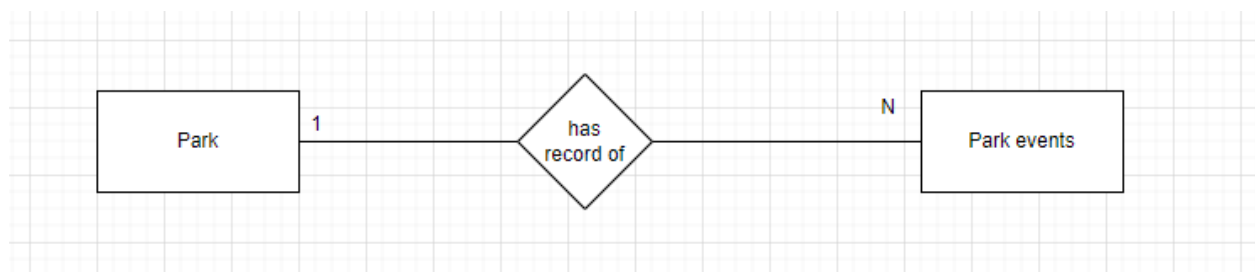
1. Identifying relationships and cardinalities among the entities:







Park events entity is used to store historical data of park.

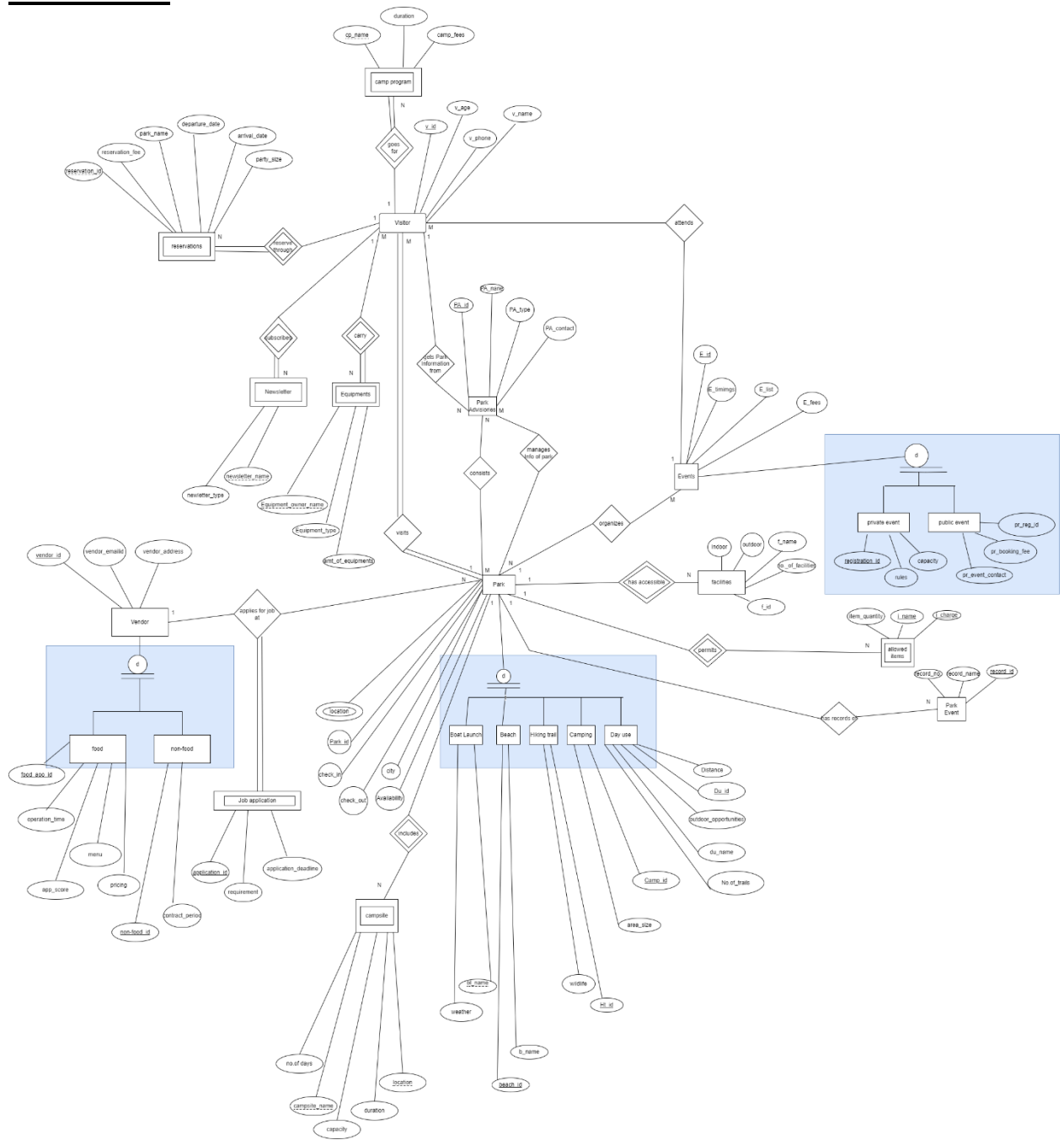


2. **Initial Chen Model:**

1. I established relationships between entities.
2. I added the attributes to its corresponding entities.
3. Added cardinalities
4. Disjoint total constraint is applied on the sub-entities of park. Where Park is the super-class.
5. Disjoint total constraint is applied on the sub-entities of park. Where Vendor is the super-class.
6. Disjoint partial constraint is applied on the sub-entities of park. Where Events is the super-class.

7. I added total participation corresponding to weak entities because a weak entity always has a total participation constraint with respect to its identifying relationship because it cannot be identified independently of its owner identity.
8. I added multi-valued attribute to the entity park i.e. location because there could be many location. Multi-valued attributes are represented as double value.
9. The job applications is a weak entity as it does not stay independently without vendor and park.

3. Initial EERD:



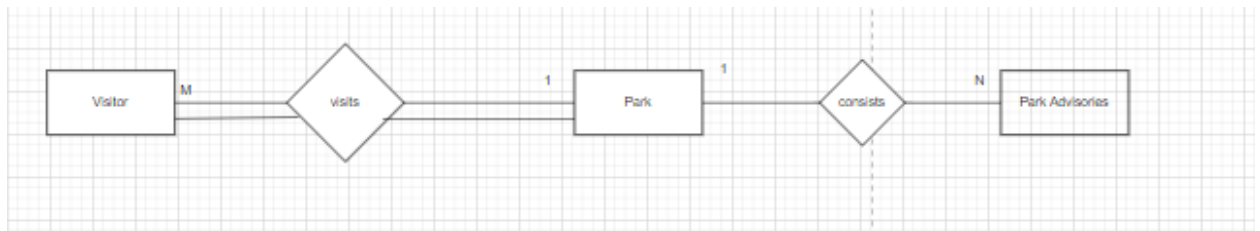
4. Errors, design issues, cardinalities I found after designing the initial EERD with their possible solutions:

1. Disjoint total constraint added in my initial erd to the superclass park will restrict the future incoming data/values of the park. It will not allow to add new type of Park because the person who will access the database will assume that the disjoint total constraint is present which means all parks are mentioned and no new parks will be added. So to avoid the inefficiency and scope of the future incoming values, overlap partial constraint should be applied to improve flexibility. "Overlap" because the characteristics in each park is same. Disjoint total constraint is represented as (d with double line)
Overlap partial constraint is represented as (0 with single line)
2. I also mentioned historical data in my initial EERD.
3. Similarly I changed disjoint total constraint to disjoint partial constraint of vendor and events.
4. After assigning the cardinalities in my initial ERD, I found errors in design issues and fan-trap.

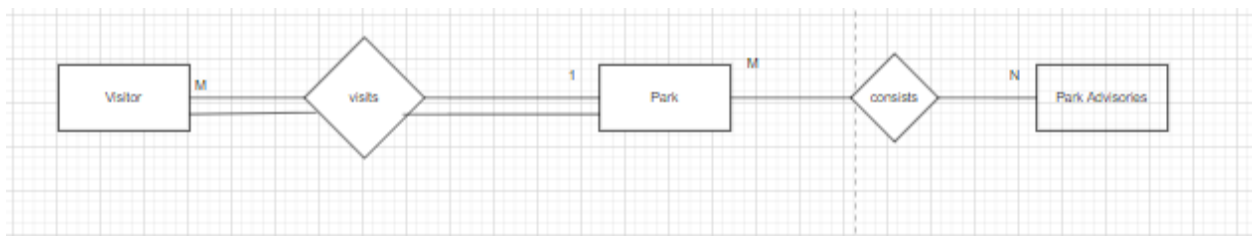
As there were many 1:M entities related to Park, it caused design traps like fan-traps. Fan-traps occur when there are 2 1:M relationship on one entity.

The most prominent ones are listed below: (Similarly did to other fan-traps too in the final ERD)

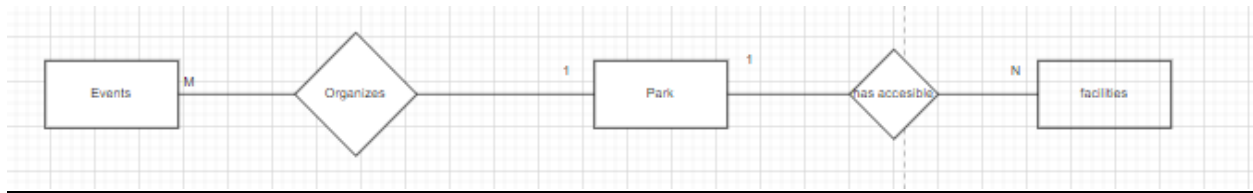
Fan-trap 1:



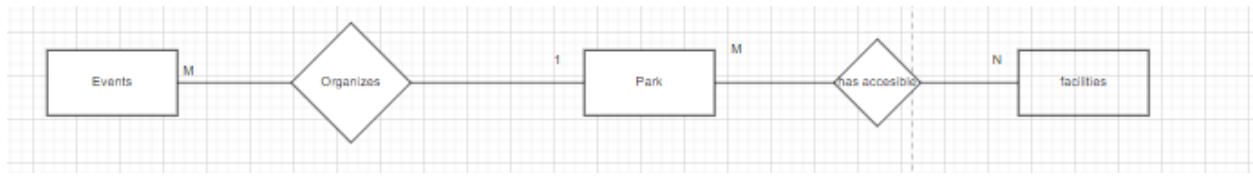
Solution:



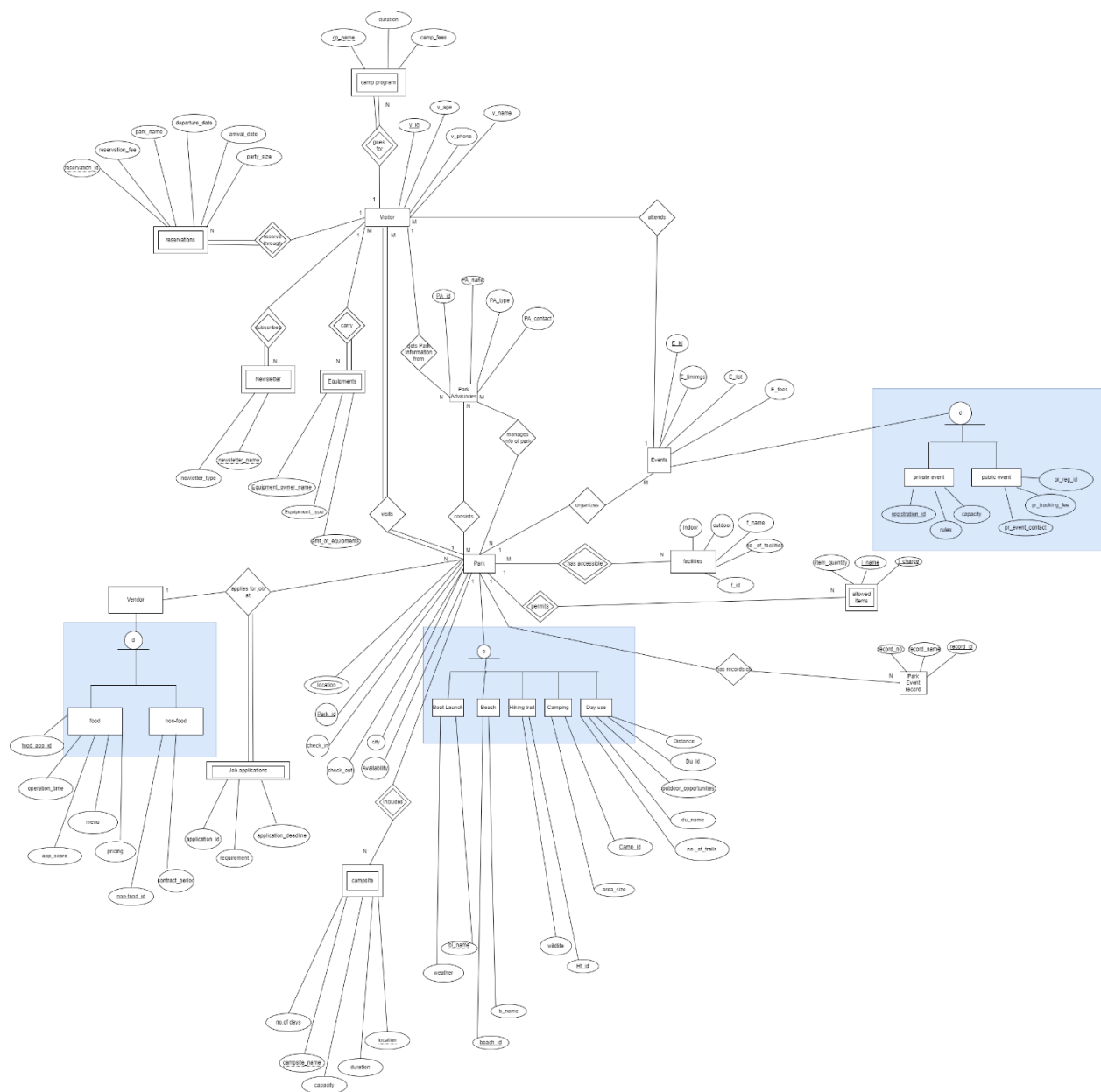
Fan-trap-2:



Solution:



5. Final EERD:



6. References:

For learning purpose:

<https://dal.brightspace.com/d2l/le/content/221749/viewContent/3010112/View>

<https://beginnersbook.com/2015/04/dbms-tutorial/>

<https://www.tutorialspoint.com/Difference-between-Strong-Entity-and-Weak-Entity#:~:text=A%20weak%20entity%20always%20has,to%20the%20same%20owner%20entity.>