DALHOUSIE UNIVERSITY

Data Management, Warehousing and Analytics

Jaskaran Singh

MACS

B00948857

js356337@dal.ca

Assignment 1

PROBLEM 2

Gitlab Repository link:

https://git.cs.dal.ca/singh16/csci5408_s23_b00948857_jaskaran_singh.git

Problem 2

Step 1 & 2: Entities

Following are the entities found after visiting https://parks.novascotia.ca/ related to the problem statement:

1. Park

- This entity will provide details of the parks in the NS region.
- It is a strong entity as it has park id as primary key (PK) for its unique identification.

2. Reservation

- It contains detail of the reservation done by a user in the park.
- It is a strong entity as every reservation has reservation id as primary key.

3. Visitor

- It contains detail of the visitor who is planning to visit the park.
- Every visitor will have id as PK, so it is a strong entity.

4. Payment

- It contains detail of the payment done by the visitor for reservation.
- It is a strong entity as it has payment id as primary key.

5. Staff

- It contains detail of the employees working for the park.
- It is a strong entity as every staff member has staff id for its unique identification.

6. Rules

- It contains detail of the rules & regulations for the parks.
- It is a weak entity as it depends on the park for its identification.

7. Facilities

- This entity has details of the park facilities & services.
- It is a strong entity as it can be uniquely identified on the basis of facility id.

8. Event

- This entity contains detail of the events happening in the parks.
- It is a weak entity as it depends on specific park for its identification.

9. Alerts

- It contains detail of the park's advisories, news and alerts.
- It is a weak entity as it depends on the park for its identification.

10. Vendor

- This entity contains list of vendors operating in the parks.
- It is a strong entity as every vendor will have vendor id for its identification.

11. Equipment

- It contains the list of equipments available for different facilities in the park.
- Since every equipment has equipment id as primary key, it is a strong entity.

12. Site

- This entity contains detail of the sites available for the reservation for camping in parks for the visitors.
- It is a weak entity as it depends on the park for its identification.

Relationships with cardinality:

- Park has **one-to-many** relationship with facility.
- Park has **one-to-many** relationship with site.
- Park has **one-to-many** relationship with reservation.
- Park has **one-to-many** relationship with alert.
- Park has **one-to-many** relationship with rule.
- Park has **one-to-many** relationship with event.
- Reservation has **many-to-one** relationship with visitor.
- Reservation has **many-to-many** relationship with site.
- Reservation has **one-to-one** relationship with payment.
- Reservation has **many-to-one** relationship with equipment.
- Reservation has **many-to-many** relationship with facility.
- Visitor has **one-to-many** relationship with payment.
- Site has **one-to-many** relationship with facility.
- Facility has **one-to-many** relationship with equipment.
- Vendor has **one-to-many** relationship with equipment.
- Staff has **one-to-many** relationship with facility.

Step 3: <u>Conceptual ERD using Crow's foot model</u>

- Following ERD has been prepared using https://draw.io/ .
- All entities and relationships between them have been mentioned with cardinality.

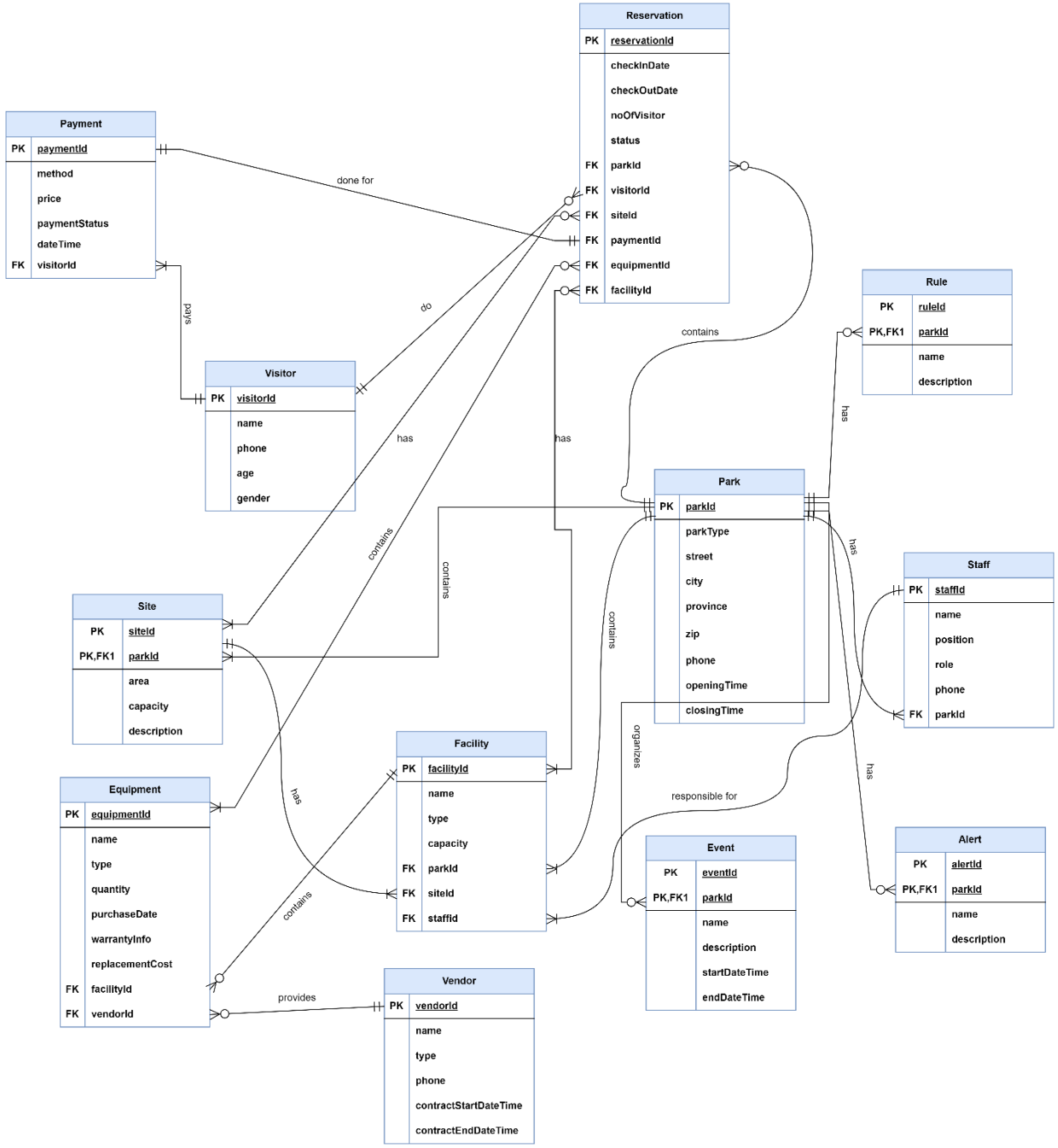

Figure 1: Conceptual ERD using Crow's foot model

Step 4: <u>Problems with above ERD and its solution:</u>

- Since every event can have some rules and regulations and alerts related to its scheduling. Therefore, it should have relationship with Rule and Alert entities. But here, Park has relationship with Rules and Alert and Event have relationship with Park. So, a **chasm trap** exists. It can be resolved by building new relationships b/w Event and Alert and also Event and Rule.

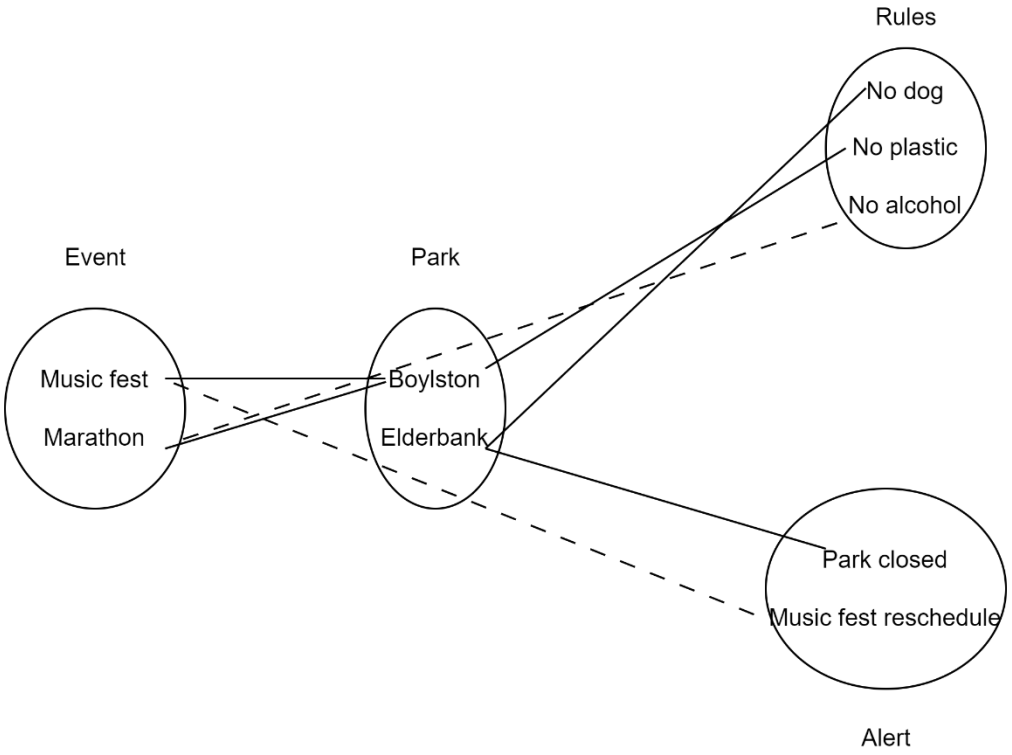

Figure 2: Chasm Trap shown b/w entities

- Also, there should be a relationship b/w vendor and park as parks can have multiple vendors working on contractual basis for facilities in the park and one vendor could provide service in multiple parks (M: N relationship).
- Additionally, there should be a relationship b/w reservation and event as visitor can have reservation for an event in the park (N:1 relationship).
- No fan trap or absence of capturing historical data is noted in the design. For e.g., for equipment, if price vary in future, attributes are already present in equipment table for capturing relevant data, i.e., replacement cost and purchase date.

After resolving these design issues, final conceptual model has been drawn on the next page:

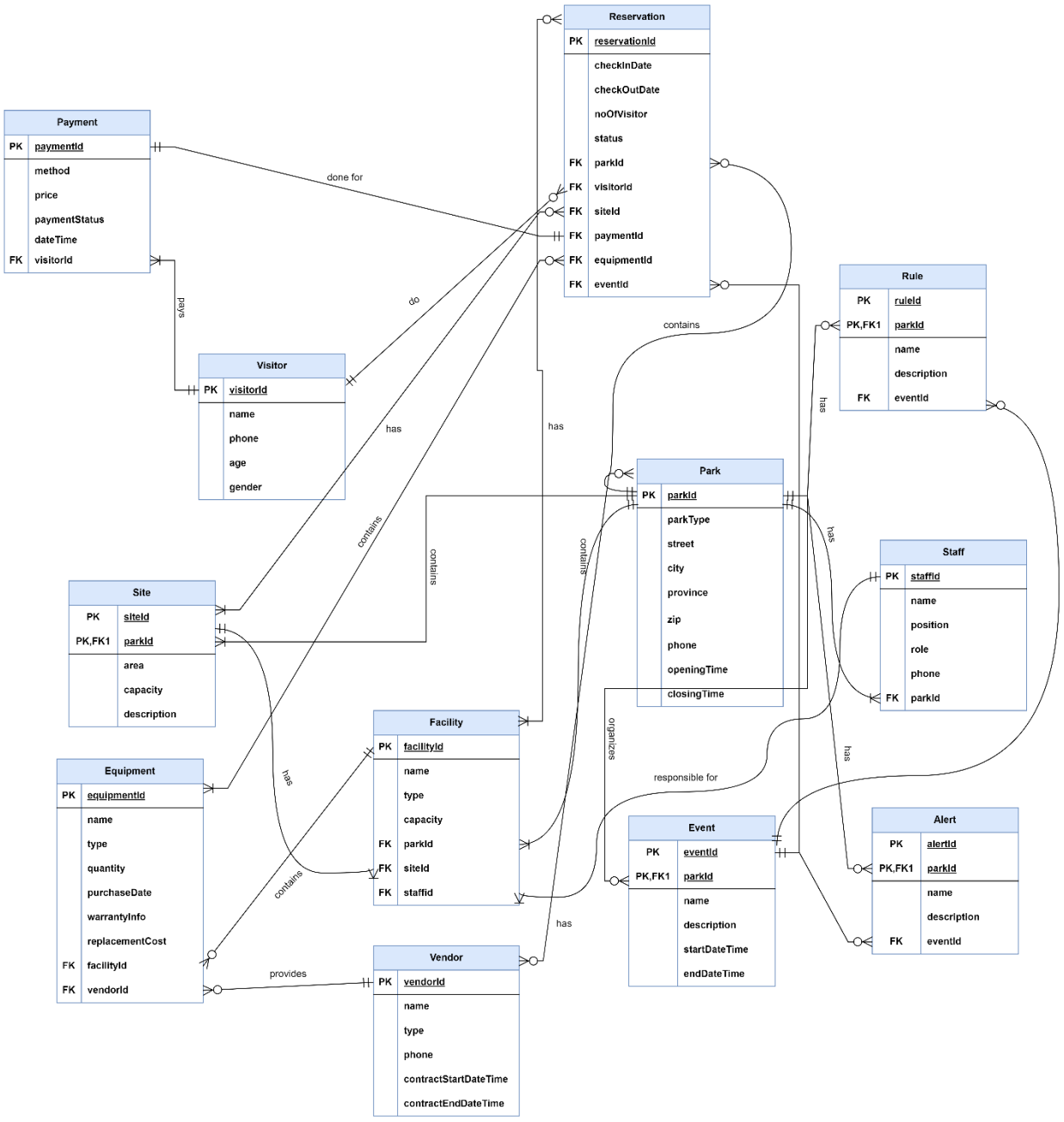

Figure 3: Crow's foot model after resolving design issues

Step 5: Normalization in logical model

- Since M: N relations exist in the above conceptual model in vendor: park entities and also for reservation: facility entities, they need to normalized to make them 1NF.

- Two new tables, Park_has_vendor and Reservation_has_facility have been introduced as shown below:

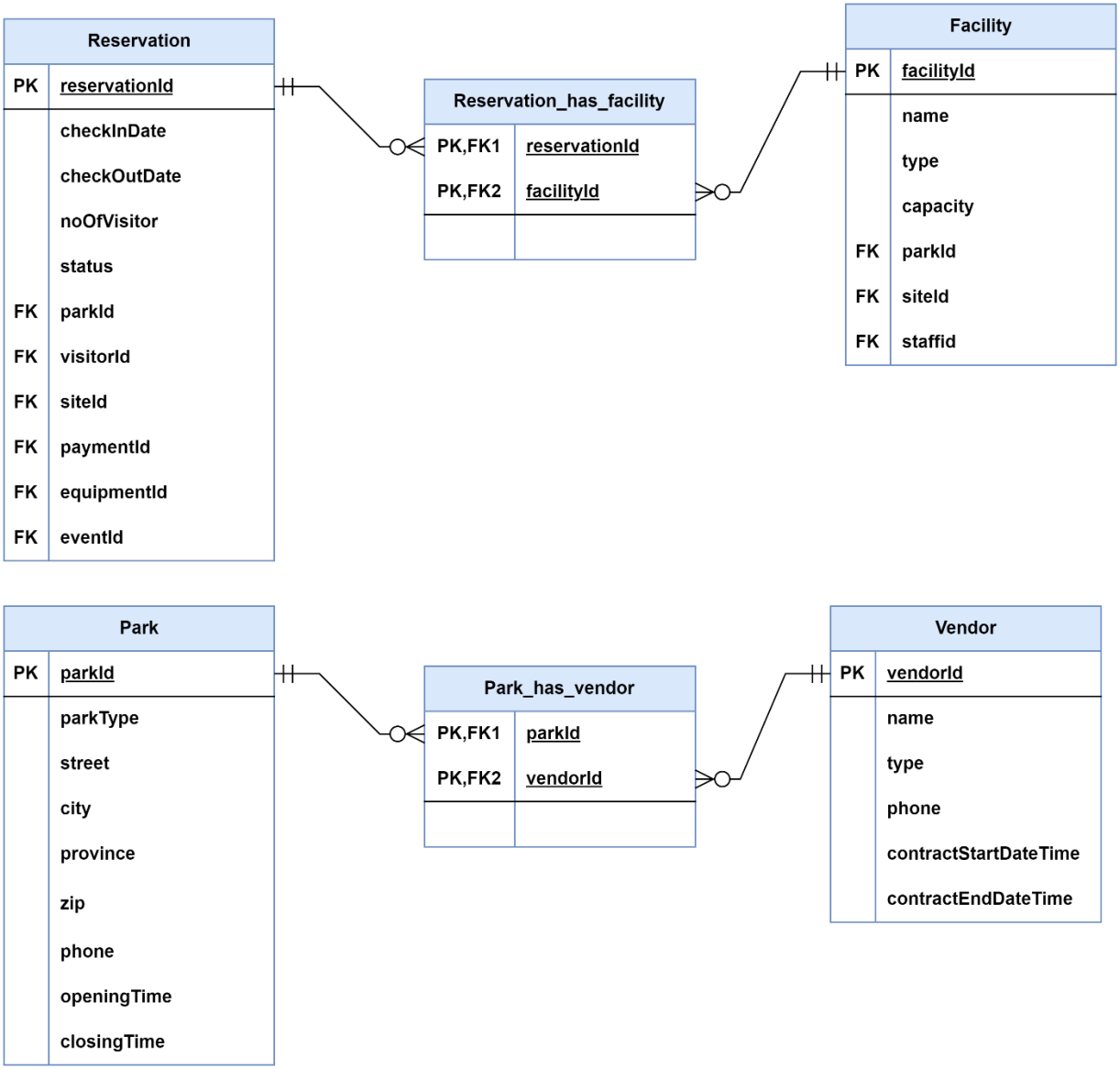

Figure 4: Normalization in logical model

Step 6 & 7: Physical Model

'ParkManager' database has been created and DDL queries have been attached in the 'park_manager_queries' file.

Step 8: <u>Observations after reverse engineering</u>

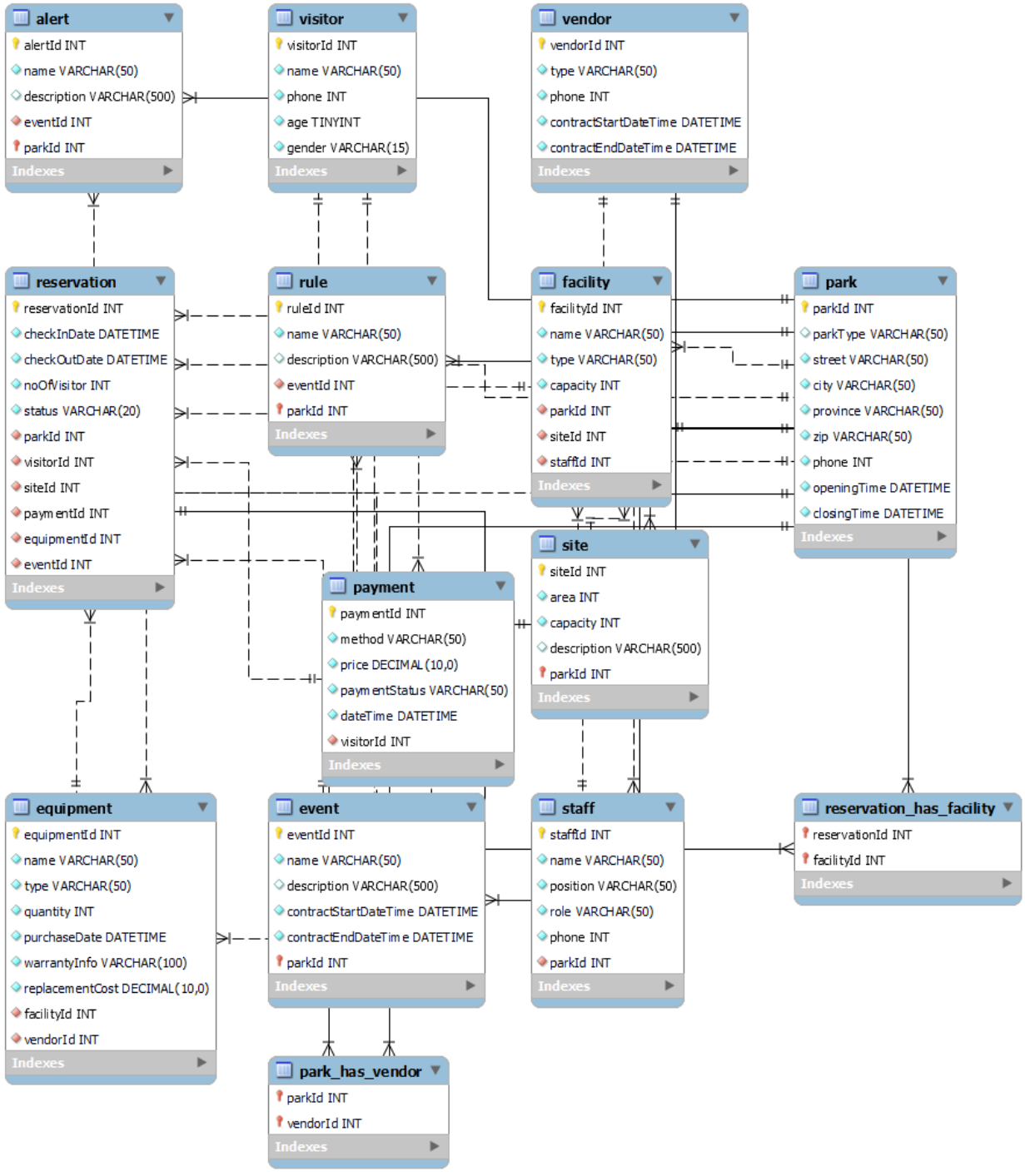

Figure 5: ERD after reverse engineering

- **Similarities found b/w manual ERD and ERD generated by MySQL Workbench:**

    1. No. of entities are same in both of the designs.
    2. Cardinality of the relationship's b/w entities/tables is also same in both of the design.
    3. Primary keys, foreign keys, weak entities, etc. are same in both of the designs.

- **Differences found b/w manual ERD and ERD generated by MySQL Workbench:**

S.No	Manual ERD	ERD by MySQL Workbench
1.	Attributes data type not mentioned	Attributes data type mentioned
2.	Relationship b/w weak and strong entity (Identifying relationship) and strong entities shown using same normal lines.	For weak to strong entity, normal lines are used, and for strong-to-strong entity, dashed lines are used.
3.	Attribute constraints like not null or unique has not been shown in ERD.	All attributes' constraints are shown separately using unique icons.

References:

- **Draw.io, ERD Tool**
  Link: https://www.draw.io

- **ERD with draw.io**
  Link: https://drawio-app.com/blog/entity-relationship-diagrams-with-draw-io/

- **Chen and Crow's Foot notation, Gleek**
  Link: https://www.gleek.io/er-diagrams