CSCI 5408

Data Management, Warehousing, And Analytics

Assignment 1 - Problem 2

Building a Data Model for Nova Scotia on its Provincial Parks.

Prepared By

Bhavisha Oza (B00935827)

Problem-2: Building a Data Model for Nova Scotia on its Provincial Parks.

Step-1 & 2: List of unique entities & it's reason for Nova Scotia Provincial Parks

1. **Park:**
   All the provincial parks in Nova Scotia must be maintained by the park entity.
2. **Visitor:**
   The visitor entity records data on visitors, such as name, address, phone number, gender, etc.
3. **Event:**
   Information on the numerous activities held in parks is carried by the event entity.
4. **Facility:**
   Each facility related to several parks is listed in the facility entity.
5. **Hotel:**
   The hotel entity maintains a list of all the hotels and their connections to various parks.
6. **Department:**
   All the departments that may be present in several parks must be stored in the department entity.
7. **Staff:**
   Represents the staff members working department and the park. It includes attributes like name, position, contact information, and assigned tasks.
8. **Dependant:**
   Represents the family members of the staff working at the Bed & Breakfast. It includes attributes like family member name, relationship to the staff member, and contact information.
9. **Adventure_Sport:**
   The Adventure_Sport entity is in charge of storing the adventure sports connected to various parks.
10. **Amusement_Ride:**
    The list of numerous amusement rides present in various parks is maintained by the Amusement_Ride entity.

Step-3: Initial ERD using Crow's Foot Model

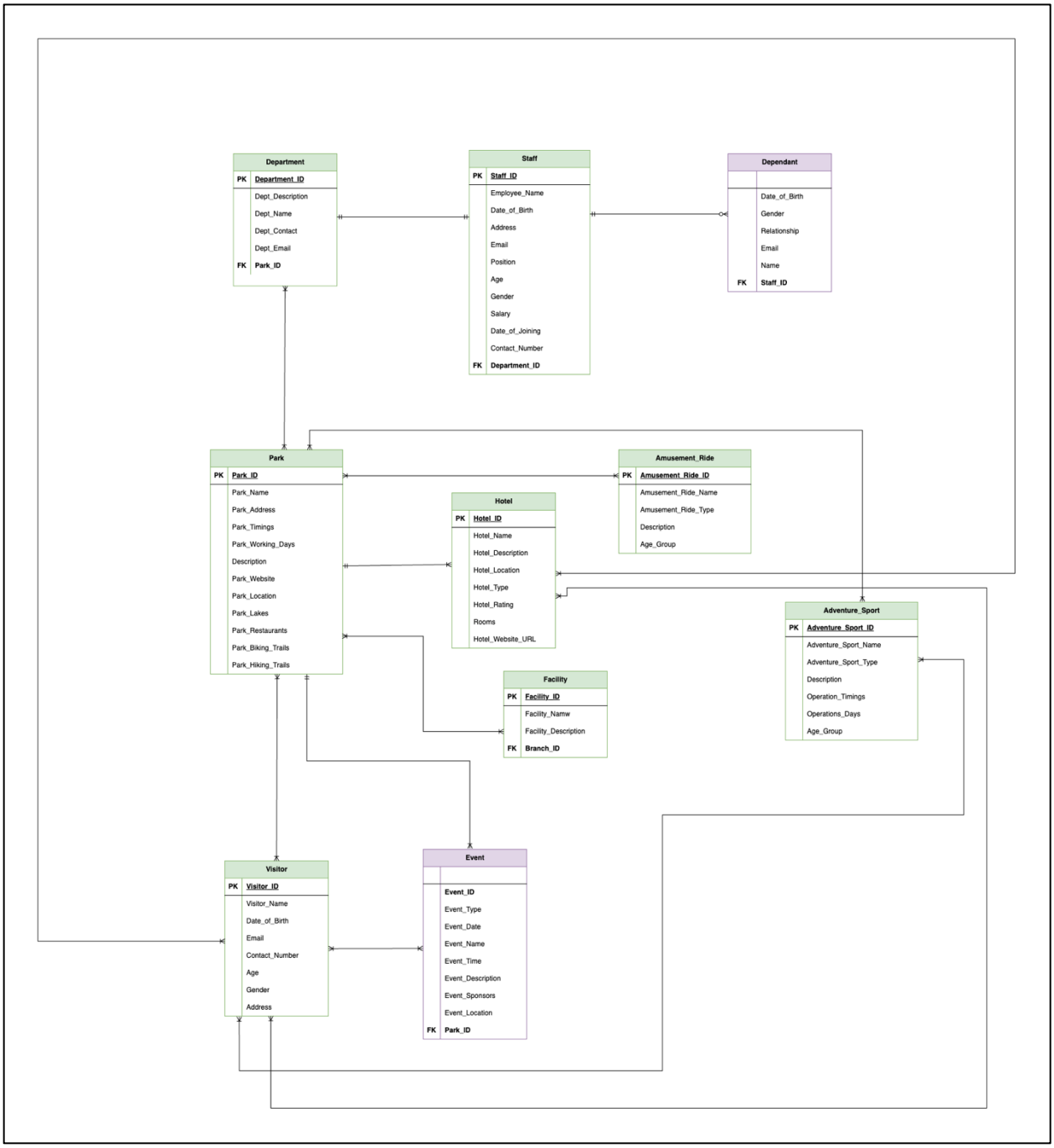

Figure 1: Initial Conceptual ER Diagram (Crow's Foot Model) of Nova Scotia Provincial Parks

Explanation: This Crow's Foot model ERD includes the strong entities and the weak entities. It represents the attributes for each entity, and cardinalities between them [2,4].

- ERD for Nova Scotia Provincial Parks using Crow's Foot model has been drawn using https://www.draw.io [3].
- Primary key is highlighted with the word PK.
- Weak Entities are displayed with VIOLET color.
- Strong Entities are indicated with GREEN color.

Types of Attributes:

1. Prime Attributes:
    a. Park_ID from Park entity
    b. Staff_ID from Staff entity
    c. Department_ID from Department entity
    d. Hotel_ID from Hotel entity
    e. Facility_ID from Facility entity
    f. Visitor_ID from Visitor entity
    g. Adventure_Sports_ID from Adventure_Sports entity
    h. Amusement_Ride_ID from Adventure_Sports entity
2. Derived Attributes:
    a. Staff_Age from Staff entity
    b. Visitor_Age from Visitor entity
3. Multi-valued Attributes:
    a. Rooms from Hotel entity
    b. Park_Lakes from Park entity
    c. Park_Restaurants from Park entity
    d. Park_Biking_Trails from Park entity
    e. Event_Sponsors from Event entity
4. Partial Attributes:
    a. Dependent_Name from Dependent entity
    b. Event_ID from Event entity

Note: The rest of the attributes are all normal attributes.

All relationships:

1. The **Department** and **Staff** entities are in a one-to-many relationship. Although a department employs several staff members, each staff person only works in one department. A staff person only oversees one department at a time, and a department is overseen by one staff member.
2. The **Staff** and **Dependent** entities are in a one-to-many relationship. A staff person may have several dependents who belong to them, but each dependent is the property of only one staff member.
3. The **Visitor** and **Event** entities are in a many-to-many relationship. A visitor goes for multiple events and, an event is visited by multiple visitors.
4. The **Visitor** and **Hotel** entities are in a many-to-many relationship. A visitor books multiple hotels and, a hotel has multiple visitors check-in.

5. The **Visitor** and **Adventure_Sport** entities are in a many-to-many relationship. A visitor tries multiple adventure sports and, an adventure sport is tried by multiple visitors.
6. The **Visitor** and **Amusement_Ride** entities are in a many-to-many relationship. A visitor tries multiple amusement rides and, an amusement ride is tried by multiple visitors.
7. The **Park** and **Department** entities are in a many-to-many relationship.
8. The **Park** and the **Visitor** entities are in a many-to-many relationship. A park has multiple visitors and, a visitor visits multiple parks.
9. The **Park** and **Event** entities are in a one-to-many relationship. A park organises multiple events but, an event is organized by one park.
10. The **Park** and **Facility** entities are in a many-to-many relationship. A park has many facilities and, a facility is having in parks.
11. The **Park** and **Hotel** entity are in a one-to-many relationship. A park **includes** multiple hotels but, a hotel is only including by one park.
12. The **Park** and **Adventure_Sport** entities are in a many-to-many relationship. A park may provide several adventure sports, and a park may offer a particular adventure activity.
13. The **Park** and **Amusement_Ride** entities are in a many-to-many relationship. A park provides a variety of amusement rides, while a park provides a variety of rides.

Step-4: Design issue in initial Conceptual Model

In the **Figure 1** there were some design issues which were solved after making initial conceptual ERD model only [2].
The fan trap occurs in the relationship between the Staff entity and the Park entity. According to the cardinality given, a park contains several departments, and each department employs many people. The staff entity and the park entity, however, are not directly related. In this scenario, it becomes ambiguous, leading to a fan trap.
In the refined ERD, we have made the following changes to address the design issues:

Fan Trap:

- Create a one-to-many relationship between the Park entity and the Staff entity. A park can have multiple staff members, but a staff member can only belong to one park.

Refined Conceptual Crow's Foot model ERD:

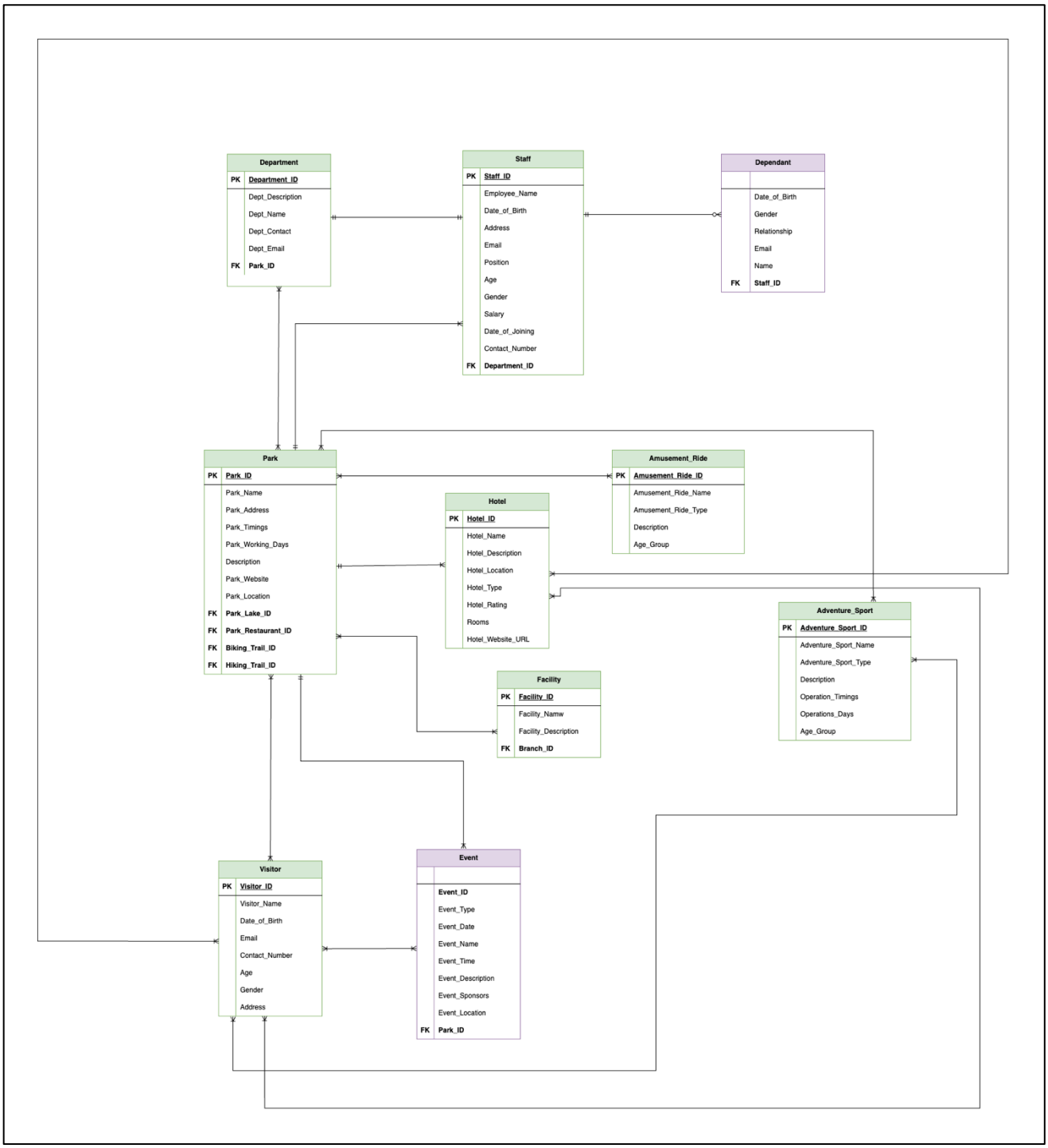

Figure 2: Refined Conceptual ERD (Crow's Foot Model) of Nova Scotia Provincial Parks

Step-5: Required Normalization

Many of the park entities have **multiple values**. Some of them may be characterised by a multi-valued attribute. Thus Lake, biking trail, and a restaurant are converted to **separate entities**. To normalize the multivalued attributes in the entities, need to perform the first normal form (1NF) by creating separate tables for these attributes. By separating the multivalued attributes into separate tables, we ensure that each attribute in a table contains atomic values. Here's the modified structure after applying 1NF. The following entities are related to the park entity on a one-to-many basis:
- Park_Restaurant
- Park_Biking_Trail
- Parrk_Hiking_Trail
- Park_Lake
- Event_Sponsors
- Hotel_Rooms

The attributes for the newly defined entities are as follows:

1. **Park_Restaurant:**
   - Restaurant_ID (Primary Key)
   - Restaurant_Name
   - Restaurant_Email
   - Restaurant_Contact_number
   - Restaurant_Address
   - Restaurant_Working_Hours
2. **Park_Biking_Trail:**
   - Biking_Trail_id (Primary Key)
   - Name
   - Description
   - Distance
   - Difficulty_level
   - Location
   - Age_group
3. **Park_Lake:**
   - Lake_id (Primary Key)
   - Name
   - Description
   - Lake_Type
   - Lake_Depth
   - Location
4. **Park_Hiking_Trail:**
   - Hiking_Trail_id (Primary Key)
   - Name
   - Description
   - Distance
   - Difficulty_level

- Location
- Age_group

5. **Event_Sponsors:**
   - Sponsor_Name (Partial Key)
   - Sponsor_Contact_Number
   - Sponsorship_Amount
   - Sponsorship_Date
6. **Hotel_Rooms:**
   - Room_ID (Partial Key)
   - Room_Type
   - Room_Capacity
   - Price
   - Availability

Figure 3 illustrates the final logical model which is drawn using draw.io [3]

Final Logical Crow's Foot Model ERD:

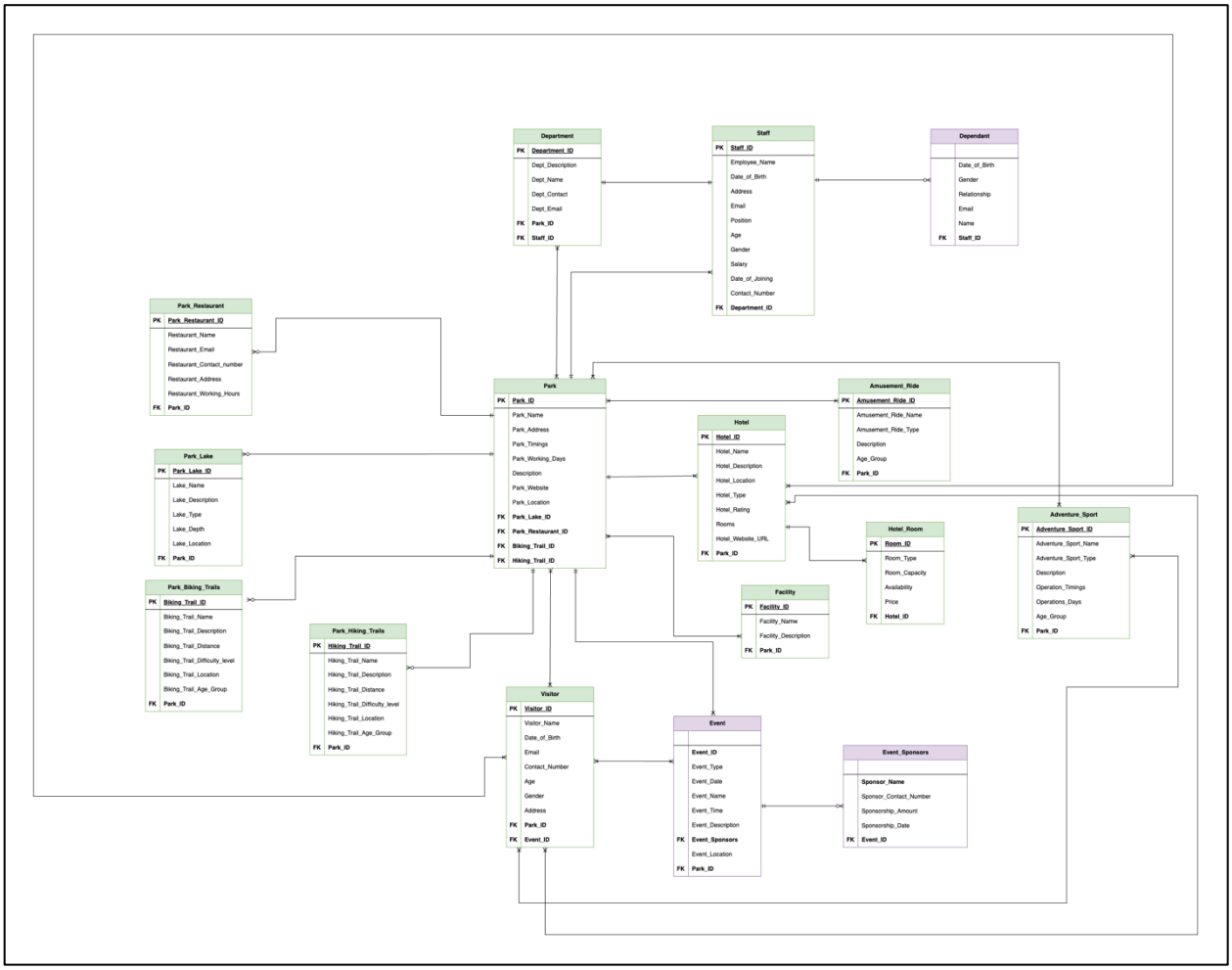

Figure 3: Logical ER Diagram of Nova Scotia Provincial Parks

Step-6: Database and empty tables creation

-- New Schema creation for database nova scotia provincial park

CREATE SCHEMA IF NOT EXISTS provincial_park;

USE provincial_park;

Created all the following tables:

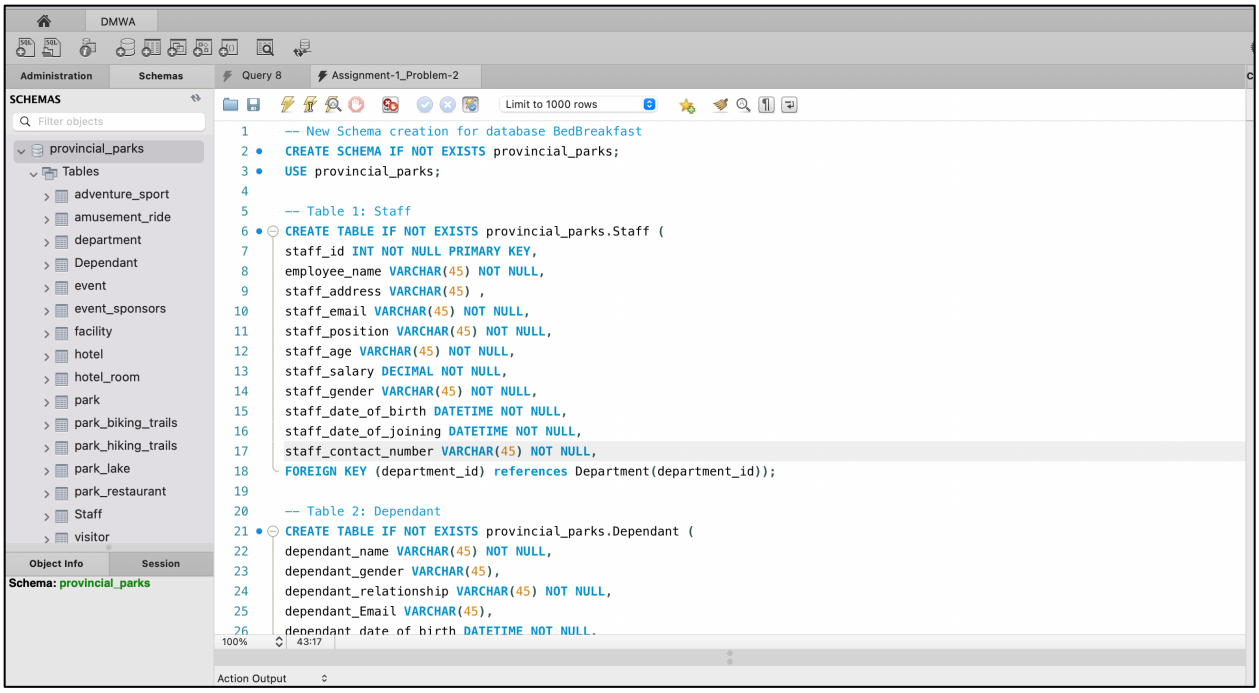

Figure 4: Physical Model of Nova Scotia Provincial Parks

Step-7: Export the SQL Dump

To export the SQL dump of MySQL Workbench database, followed below mentioned steps [5]:

1. Open MySQL Workbench and connect to your MySQL database server.
2. Select the database you want to export in the "SCHEMAS" section on the left-hand side.
3. Here the schema is provincial_parks
4. Go to the "Server" menu at the top and choose "Data Export".
5. In the "Data Export" window, select the "Export to Self-Contained File" option.
6. Choose the location where the SQL dump file needs to be saved.
7. Select "Dump Structure and Data" option to export both the database structure and data.
8. Click the "Start Export" button to begin the export process.
9. The dump is saved in the given folder.

-- MySQL dump 10.13 Distrib 8.0.31, for macos12 (x86_64)

```
--

-- Host: 127.0.0.1    Database: provincial_parks

-- ------------------------------------------------------

-- Server version        8.0.31


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!50503 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
*/;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;


--

-- Table structure for table `adventure_sport`

--


DROP TABLE IF EXISTS `adventure_sport`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `adventure_sport` (
  `adventure_sport_id` int NOT NULL,
  `adventure_sport_name` varchar(45) NOT NULL,
  `adventure_sport_description` varchar(45) DEFAULT NULL,
  `adventure_sport_operation_timings` datetime DEFAULT NULL,
  `adventure_sport_operation_days` varchar(45) NOT NULL,
  `adventure_sport_age_group` int NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`adventure_sport_id`),
  KEY `fk1_park_id_idx` (`park_id`),
  CONSTRAINT `fk1_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `adventure_sport`
--

LOCK TABLES `adventure_sport` WRITE;
/*!40000 ALTER TABLE `adventure_sport` DISABLE KEYS */;
/*!40000 ALTER TABLE `adventure_sport` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `amusement_ride`
```

```
--


DROP TABLE IF EXISTS `amusement_ride`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `amusement_ride` (
  `amusement_ride_id` int NOT NULL,
  `amusement_ride_name` varchar(45) NOT NULL,
  `amusement_ride_type` varchar(45) NOT NULL,
  `amusement_ride_description` varchar(45) NOT NULL,
  `amusement_ride_age_group` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`amusement_ride_id`),
  KEY `fk2_park_id_idx` (`park_id`),
  CONSTRAINT `fk2_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `amusement_ride`
--


LOCK TABLES `amusement_ride` WRITE;
/*!40000 ALTER TABLE `amusement_ride` DISABLE KEYS */;
/*!40000 ALTER TABLE `amusement_ride` ENABLE KEYS */;
```

```
UNLOCK TABLES;


--
-- Table structure for table `department`
--


DROP TABLE IF EXISTS `department`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `department` (
  `department_id` int NOT NULL,
  `department_name` varchar(45) NOT NULL,
  `department_description` varchar(45) DEFAULT NULL,
  `department_contact` int DEFAULT NULL,
  `department_email` varchar(45) NOT NULL,
  `staff_id` int DEFAULT NULL,
  `park_id` int DEFAULT NULL,
  PRIMARY KEY (`department_id`),
  KEY `staff_id_idx` (`staff_id`),
  KEY `fk9_park_id_idx` (`park_id`),
  CONSTRAINT `fk9_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`),
  CONSTRAINT `staff_id` FOREIGN KEY (`staff_id`) REFERENCES `Staff` (`staff_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--

-- Dumping data for table `department`

--


LOCK TABLES `department` WRITE;

/*!40000 ALTER TABLE `department` DISABLE KEYS */;

/*!40000 ALTER TABLE `department` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `Dependant`

--


DROP TABLE IF EXISTS `Dependant`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `Dependant` (

  `dependant_name` varchar(45) NOT NULL,

  `dependant_gender` varchar(45) DEFAULT NULL,

  `dependant_relationship` varchar(45) NOT NULL,

  `dependant_Email` varchar(45) DEFAULT NULL,

  `dependant_date_of_birth` datetime NOT NULL,

  `staff_id` int NOT NULL,

  KEY `staff_id` (`staff_id`),

  CONSTRAINT `dependant_ibfk_1` FOREIGN KEY (`staff_id`) REFERENCES `Staff`
(`staff_id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `Dependant`
--


LOCK TABLES `Dependant` WRITE;
/*!40000 ALTER TABLE `Dependant` DISABLE KEYS */;
/*!40000 ALTER TABLE `Dependant` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `event`
--


DROP TABLE IF EXISTS `event`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `event` (
  `event_id` int NOT NULL,
  `event_type` varchar(45) NOT NULL,
  `event_date` datetime NOT NULL,
  `event_name` varchar(45) NOT NULL,
  `event_time` datetime NOT NULL,
```

```
  `event_description` varchar(45) DEFAULT NULL,
  `event_sponsors` varchar(45) NOT NULL,
  `event_location` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  `visitor_id` int NOT NULL,
  KEY `fk3_park_id_idx` (`park_id`),
  KEY `visitor_id_idx` (`visitor_id`),
  CONSTRAINT `fk3_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`),
  CONSTRAINT `visitor_id` FOREIGN KEY (`visitor_id`) REFERENCES `visitor`
(`visitor_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `event`
--


LOCK TABLES `event` WRITE;
/*!40000 ALTER TABLE `event` DISABLE KEYS */;
/*!40000 ALTER TABLE `event` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `event_sponsors`
--
```

```
DROP TABLE IF EXISTS `event_sponsors`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `event_sponsors` (
  `sponsors_name` int NOT NULL,
  `sponsor_contact_number` varchar(45) NOT NULL,
  `sponsorship_amount` decimal(30,0) DEFAULT NULL,
  `sponsorship_date` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `event_sponsors`
--

LOCK TABLES `event_sponsors` WRITE;
/*!40000 ALTER TABLE `event_sponsors` DISABLE KEYS */;
/*!40000 ALTER TABLE `event_sponsors` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `facility`
--

DROP TABLE IF EXISTS `facility`;
```

```
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `facility` (
  `facility_id` int NOT NULL,
  `facility_name` varchar(45) NOT NULL,
  `facility_description` varchar(45) DEFAULT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`facility_id`),
  KEY `park_id_idx` (`park_id`),
  CONSTRAINT `fk_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `facility`
--

LOCK TABLES `facility` WRITE;
/*!40000 ALTER TABLE `facility` DISABLE KEYS */;
/*!40000 ALTER TABLE `facility` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `hotel`
--
```

```
DROP TABLE IF EXISTS `hotel`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `hotel` (
  `hotel_id` int NOT NULL,
  `hotel_name` varchar(45) DEFAULT NULL,
  `hotel_description` varchar(45) DEFAULT NULL,
  `hotel_location` varchar(45) DEFAULT NULL,
  `hotel_type` varchar(45) DEFAULT NULL,
  `hotel_rating` int NOT NULL,
  `room_id` int NOT NULL,
  `hotel_website` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`hotel_id`),
  KEY `park_id_idx` (`park_id`),
  CONSTRAINT `park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `hotel`
--

LOCK TABLES `hotel` WRITE;
```

```
/*!40000 ALTER TABLE `hotel` DISABLE KEYS */;
/*!40000 ALTER TABLE `hotel` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `hotel_room`
--

DROP TABLE IF EXISTS `hotel_room`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `hotel_room` (
  `room_id` int NOT NULL,
  `room_type` varchar(45) NOT NULL,
  `room_capacity` varchar(45) NOT NULL,
  `room_availability` varchar(45) NOT NULL,
  `room_price` varchar(45) NOT NULL,
  `hotel_id` int NOT NULL,
  PRIMARY KEY (`room_id`),
  KEY `hotel_id_idx` (`hotel_id`),
  CONSTRAINT `hotel_id` FOREIGN KEY (`hotel_id`) REFERENCES `hotel` (`hotel_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

```
-- Dumping data for table `hotel_room`
--

LOCK TABLES `hotel_room` WRITE;
/*!40000 ALTER TABLE `hotel_room` DISABLE KEYS */;
/*!40000 ALTER TABLE `hotel_room` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `park`
--

DROP TABLE IF EXISTS `park`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `park` (
  `park_id` int NOT NULL,
  `park_name` varchar(45) NOT NULL,
  `park_address` varchar(45) NOT NULL,
  `park_timings` datetime NOT NULL,
  `park_working_days` varchar(45) NOT NULL,
  `park_description` varchar(45) DEFAULT NULL,
  `park_website` varchar(45) DEFAULT NULL,
  `park_location` varchar(45) NOT NULL,
  `park_restaurant_id` int NOT NULL,
```

```
  `park_lake_id` int NOT NULL,

  `park_hiking_trail_id` int NOT NULL,

  `park_biking_trail_id` int NOT NULL,

  PRIMARY KEY (`park_id`),

  KEY `park_restaurant_id_idx` (`park_restaurant_id`),

  KEY `park_lake_id_idx` (`park_lake_id`),

  KEY `park_hiking_trail_id_idx`
(`park_biking_trail_id`,`park_hiking_trail_id`,`park_lake_id`,`park_restaurant_id`),

  KEY `park_hiking_trail_id_idx1` (`park_hiking_trail_id`),

  CONSTRAINT `park_biking_trail_id` FOREIGN KEY (`park_biking_trail_id`)
REFERENCES `park_biking_trails` (`park_biking_trail_id`),

  CONSTRAINT `park_hiking_trail_id` FOREIGN KEY (`park_hiking_trail_id`)
REFERENCES `park_hiking_trails` (`park_hiking_trail_id`),

  CONSTRAINT `park_lake_id` FOREIGN KEY (`park_lake_id`) REFERENCES `park_lake`
(`park_lake_id`),

  CONSTRAINT `park_restaurant_id` FOREIGN KEY (`park_restaurant_id`) REFERENCES
`park_restaurant` (`park_restaurant_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;



--

-- Dumping data for table `park`

--



LOCK TABLES `park` WRITE;

/*!40000 ALTER TABLE `park` DISABLE KEYS */;

/*!40000 ALTER TABLE `park` ENABLE KEYS */;
```

```
UNLOCK TABLES;


--
-- Table structure for table `park_biking_trails`
--


DROP TABLE IF EXISTS `park_biking_trails`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `park_biking_trails` (
  `park_biking_trail_id` int NOT NULL,
  `biking_trail_name` varchar(45) DEFAULT NULL,
  `biking_trail_description` varchar(45) DEFAULT NULL,
  `biking_trail_distance` varchar(45) NOT NULL,
  `biking_trail_difficulty_level` varchar(45) NOT NULL,
  `biking_trail_location` varchar(45) NOT NULL,
  `biking_trail_age_group` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`park_biking_trail_id`),
  KEY `fk7_park_id_idx` (`park_id`),
  CONSTRAINT `fk7_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
```

```
-- Dumping data for table `park_biking_trails`
--

LOCK TABLES `park_biking_trails` WRITE;
/*!40000 ALTER TABLE `park_biking_trails` DISABLE KEYS */;
/*!40000 ALTER TABLE `park_biking_trails` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `park_hiking_trails`
--

DROP TABLE IF EXISTS `park_hiking_trails`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `park_hiking_trails` (
  `park_hiking_trail_id` int NOT NULL,
  `hiking_trail_name` varchar(45) NOT NULL,
  `hiking_trail_description` varchar(45) DEFAULT NULL,
  `hiking_trail_distance` varchar(45) DEFAULT NULL,
  `hiking_trail_difficulty_level` varchar(45) NOT NULL,
  `hiking_trail_location` varchar(45) NOT NULL,
  `hiking_trail_age_group` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`park_hiking_trail_id`),
```

```
  KEY `fk8_park_id_idx` (`park_id`),
  CONSTRAINT `fk8_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `park_hiking_trails`
--


LOCK TABLES `park_hiking_trails` WRITE;
/*!40000 ALTER TABLE `park_hiking_trails` DISABLE KEYS */;
/*!40000 ALTER TABLE `park_hiking_trails` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `park_lake`
--


DROP TABLE IF EXISTS `park_lake`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `park_lake` (
  `park_lake_id` int NOT NULL,
  `lake_name` varchar(45) DEFAULT NULL,
  `lake_description` varchar(45) DEFAULT NULL,
```

```
  `lake_type` varchar(45) NOT NULL,
  `lake_depth` varchar(45) NOT NULL,
  `lake_location` varchar(45) NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`park_lake_id`),
  KEY `fk6_park_id_idx` (`park_id`),
  CONSTRAINT `fk6_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `park_lake`
--


LOCK TABLES `park_lake` WRITE;
/*!40000 ALTER TABLE `park_lake` DISABLE KEYS */;
/*!40000 ALTER TABLE `park_lake` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `park_restaurant`
--


DROP TABLE IF EXISTS `park_restaurant`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
```

```
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `park_restaurant` (
  `park_restaurant_id` int NOT NULL,
  `restaurant_name` varchar(45) NOT NULL,
  `restaurant_email` varchar(45) NOT NULL,
  `restaurant_contact_number` varchar(45) NOT NULL,
  `restaurant_address` varchar(45) DEFAULT NULL,
  `restaurant_working_hours` datetime NOT NULL,
  `park_id` int NOT NULL,
  PRIMARY KEY (`park_restaurant_id`),
  KEY `fk4_park_id_idx` (`park_id`),
  CONSTRAINT `fk5_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `park_restaurant`
--

LOCK TABLES `park_restaurant` WRITE;
/*!40000 ALTER TABLE `park_restaurant` DISABLE KEYS */;
/*!40000 ALTER TABLE `park_restaurant` ENABLE KEYS */;
UNLOCK TABLES;

--
```

```
-- Table structure for table `Staff`

--


DROP TABLE IF EXISTS `Staff`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `Staff` (

  `staff_id` int NOT NULL,

  `employee_name` varchar(45) NOT NULL,

  `staff_address` varchar(45) DEFAULT NULL,

  `staff_email` varchar(45) NOT NULL,

  `staff_position` varchar(45) NOT NULL,

  `staff_age` varchar(45) NOT NULL,

  `staff_salary` decimal(10,0) NOT NULL,

  `staff_gender` varchar(45) NOT NULL,

  `staff_date_of_birth` datetime NOT NULL,

  `staff_date_of_joining` datetime NOT NULL,

  `staff_contact_number` varchar(45) NOT NULL,

  `department_id` int NOT NULL,

  PRIMARY KEY (`staff_id`),

  KEY `departmeny_id_idx` (`department_id`),

  CONSTRAINT `departmeny_id` FOREIGN KEY (`department_id`) REFERENCES
`department` (`department_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `Staff`
--

LOCK TABLES `Staff` WRITE;
/*!40000 ALTER TABLE `Staff` DISABLE KEYS */;
/*!40000 ALTER TABLE `Staff` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `visitor`
--

DROP TABLE IF EXISTS `visitor`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `visitor` (
  `visitor_id` int NOT NULL,
  `visitor_name` varchar(45) NOT NULL,
  `visitor_DOB` datetime DEFAULT NULL,
  `visitor_email` varchar(45) NOT NULL,
  `visitor_contact_number` varchar(45) DEFAULT NULL,
  `visitor_age` int NOT NULL,
  `visitor_gender` varchar(45) DEFAULT NULL,
  `visitor_address` varchar(45) DEFAULT NULL,
```

```
  `park_id` int NOT NULL,
  `event_id` int NOT NULL,
  PRIMARY KEY (`visitor_id`),
  KEY `fk4_park_id_idx` (`park_id`),
  KEY `fk2_event_id_idx` (`event_id`),
  CONSTRAINT `fk4_park_id` FOREIGN KEY (`park_id`) REFERENCES `park` (`park_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `visitor`
--

LOCK TABLES `visitor` WRITE;
/*!40000 ALTER TABLE `visitor` DISABLE KEYS */;
/*!40000 ALTER TABLE `visitor` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2023-06-05 2:07:52

Step-8: ERD by performing reverse engineering:

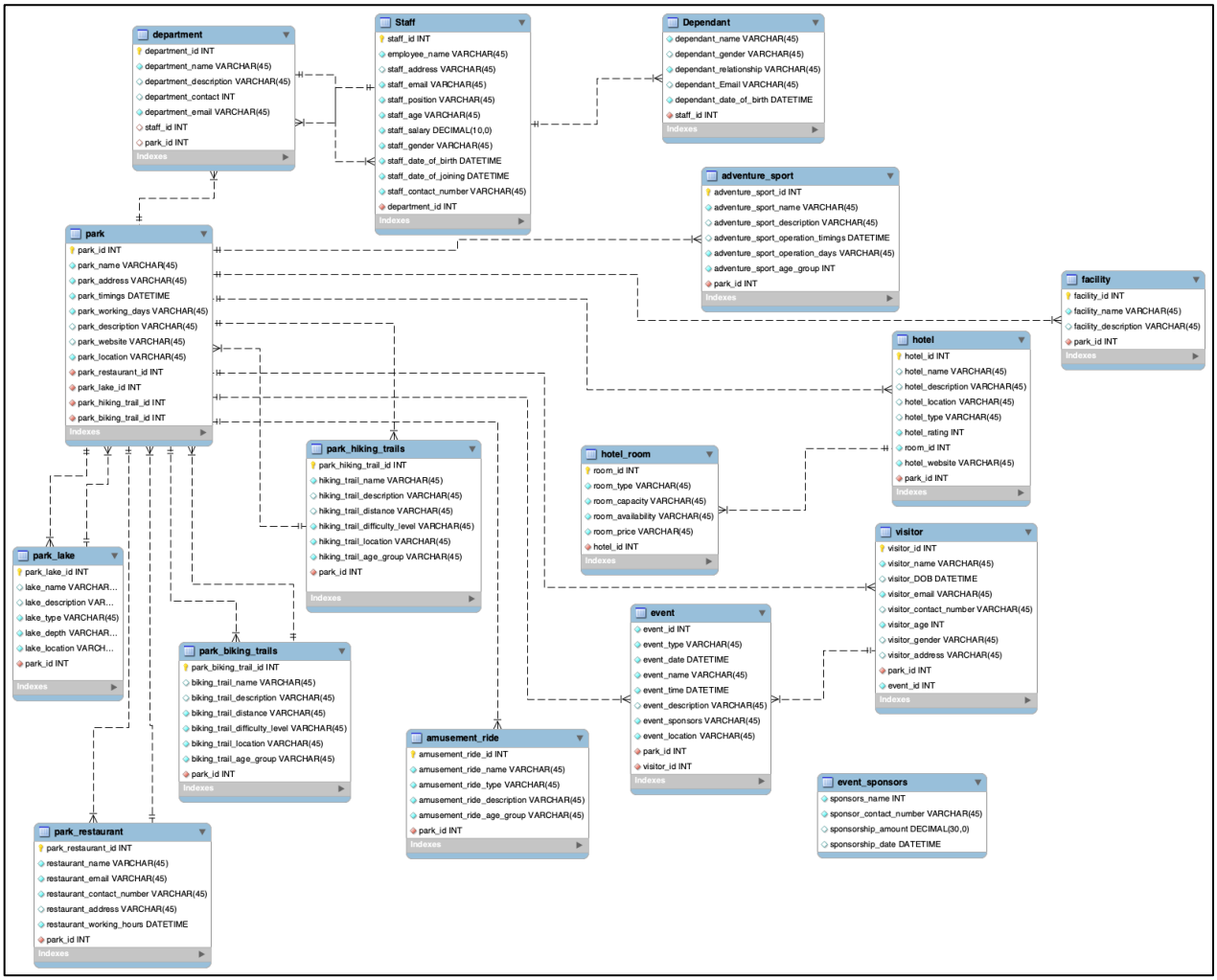

Figure 5: ER Diagram of Nova Scotia Provincial Parks with Reverse Engineering

Similarities found between the ERD generated by you (manual), and ERD generated by MySQL Workbench (automatic)

Similarities	Explanation
Entity Representation	Entities are displayed in boxes in both ERDs (ex. Crow's Foot model shown in Figure 3 and the diagram shown in Figure 5)
Standard Notation	Both ERDs use the same standard notation for entities, attributes, relationships, and cardinality.
Highlighting of primary keys	The primary keys of entities are highlighted in both ERDs, serving as unique identifiers.
Cardinality representation	Cardinality is expressed in both ERDs to indicate the number of instances of one entity that can be related to another entity.
Inclusion of foreign keys	Foreign keys are included in both ERDs to establish relationships between entities and maintain data integrity
Relationships displayed	Relationships between entities are illustrated in both ERDs, indicating one-to-one, one-to-many, or many-to-many connections.

Differences found between the ERD generated by you (manual), and ERD generated by MySQL Workbench (automatic).

Differences	Explanation
Cardinality representation	In the manually created ERD, cardinality can be represented with values such as 0, indicating that a relationship can have zero instances. However, the ERD generated by MySQL Workbench may not provide specific cardinality notations for 0 or many. It may use a more general notation for one-to-one, one-to-many, or many-to-many relationships without indicating the possibility of zero instances.
Human interpretation and decision-making	When creating an ERD manually, we rely on our understanding of the database schema and the business requirements. This allows for human interpretation and decision-making

	regarding the structure, relationships, and attributes represented in the diagram. The automatic ERD generated by MySQL Workbench relies on the provided schema metadata and may not incorporate the same level of contextual knowledge and decision-making.
Flexibility in capturing desired level of detail	The manual ERD allows for more flexibility in capturing the desired level of detail in the diagram. We can choose to include or exclude certain attributes, and provide additional annotations or descriptions as needed. The automatic ERD generated by MySQL Workbench may include a standard set of details and may not capture all the refinement of the database schema unless configured specifically.
Revision and iteration	With a manually created ERD, it is easier to revise and iterate on the design based on feedback or changing requirements. We can adjust, add annotations, and refine the diagram as needed. The ERD generated by MySQL Workbench may require regenerating the diagram if changes to the schema are made.

References:

[1] "MySQL Community Downloads," *MySQL* [Online]. Available: https://dev.mysql.com/downloads/workbench/ [Accessed: May 10, 2023].

[2] "Lecture 2_May 4, 2023," *Brightspace Dalhousie University* [Online]. Available: https://dal.brightspace.com/d2l/le/content/271677/viewContent/3617696/View [Accessed: May 27, 2023].

[3] "Flowchart Maker & Online Diagram Software," *Draw.io* [Online]. Available: https://app.diagrams.net/ [Accessed: May 20, 2023].

[4] "Lab-2," *Brightspace Dalhousie University* [Online]. Available: https://dal.brightspace.com/d2l/le/content/271677/viewContent/3628976/View [Accessed: May 19, 2023].

[5] "About the MySQL for Sitehost data backup and restoration policy," *University Information Technology Services* [Online]. Available: https://kb.iu.edu/d/apnn [Accessed: May 30, 2023].