Jesse Jordan, Gabriella Marinescu, Aishwarya Srivastava
CSE 3241, Phase 1
Professor Krishnasamy
2/15/17

---

1. Database Description

SQLand Theme Park wants to build a database to keep track of all its park operations and guests. There are many park locations nationwide. They would like to keep track of each park's location ID, and where they are located including the city, state, and zip code. Each park has 4 sections that have unique names: ToonLand, HorrorLand, WaterLand, SpaceLand. There is transportation available between all park sections. Each transportation is identified by the vehicle ID. It can be either a shuttle or a monorail.

The employees of the theme park are identified by their unique employee ID, and the database also stores their social security number, name, phone number and home address. There are many park departments at each theme park location. These departments are identified by their department ID. The database also stores the name of the department, a description of what each department entails, and a phone number for the department. The theme park has many attractions, each identified by an attraction ID. The first character of an attraction ID is always the letter 'A', the rest are all numeric digits. The database also stores the attraction name, and its description.

There are many food & beverage locations in each theme park. Each has a unique ID, a single vendor, a description, and type (it is either a quick service location or a sit down restaurant). In addition to restaurants, there are many merchandise stores, each with a unique store ID, and a description. The items sold in the stores are identified by a unique item ID, its price (which is the same across all merchandise locations), and a manufacturer.

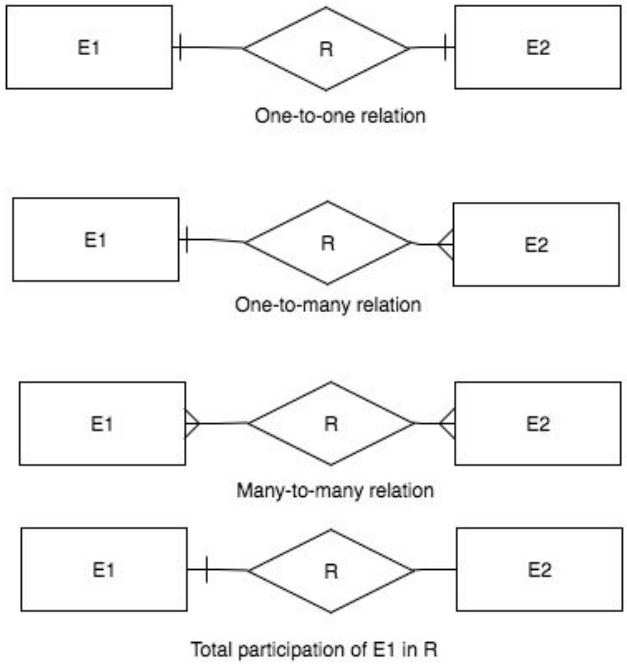
The theme park offers many character meet and greet experiences. Each experience is identified by an experience ID. The first character of a character experience ID is always the letter 'C', and the rest are numeric digits. The database also stores the start time and end time of the meet and greet, and a description of the event. There are also many live entertainment shows in the theme park, each identified by a unique show ID. It has a scheduled start time and end time, and has a type that is either a stage show or a parade.

There are various photo-booths around the theme park that are either located at an attraction or a character experience location. They are identified by a unique booth ID, and also lists the ID of the attraction or character experience they are photographing. There are many first-aid locations throughout the park each with a unique location ID.

The ticket information of each park guest is also stored in the database. Each ticket has a unique ID, the name of guest, their phone number, home address, age, and date of visit. The park also has a hotel that the database wants to store information on. Each room in the hotel is identified by the unique room number, and stores information of the guest including their name, ticket ID, the number of guests the room is booked for, the check-in date, the check-out date, and the guest's room type, which is either standard, double, or premium.

2. Entity-relationship

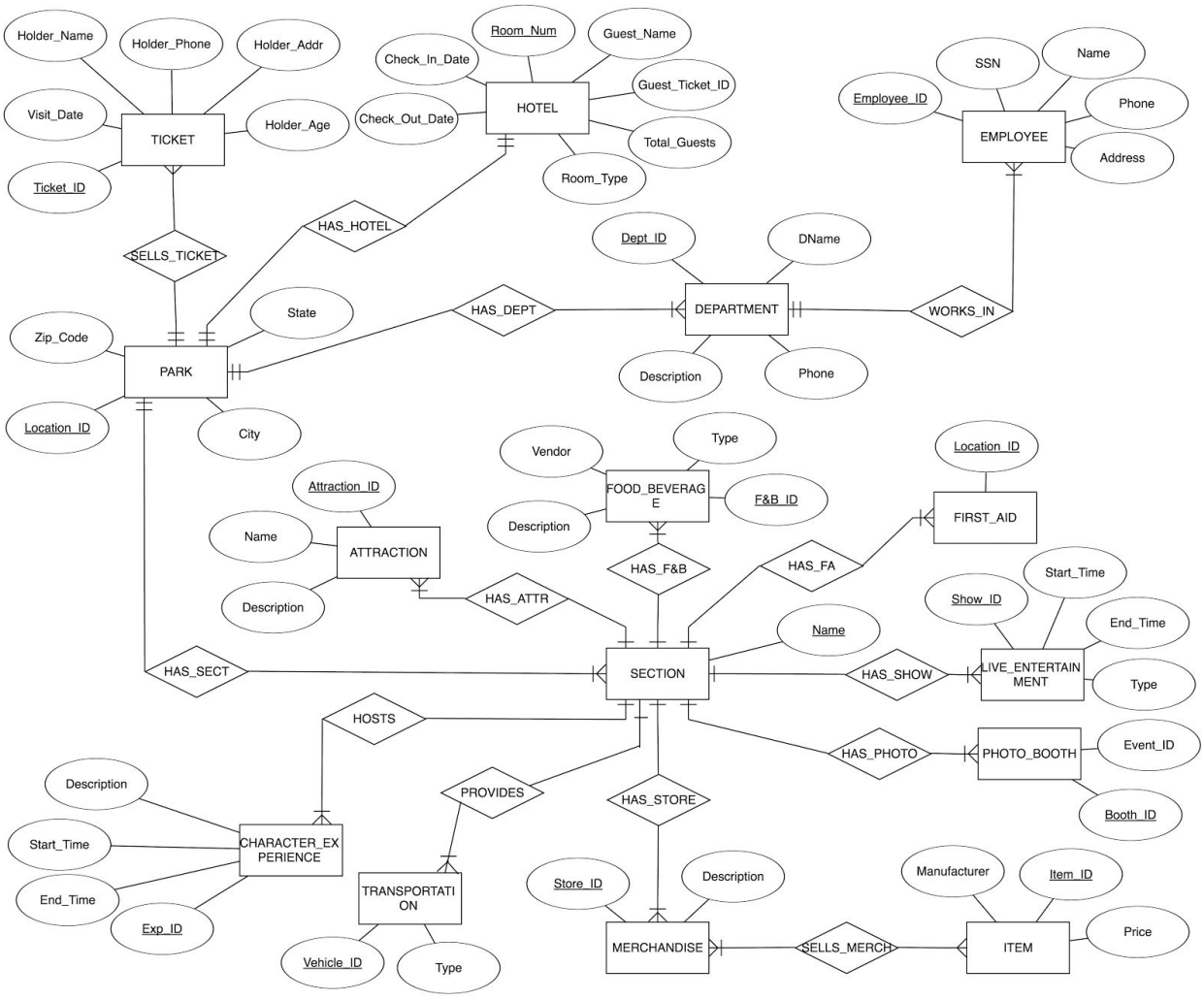
2.1. Key



2.2. Additional Constraints:

- Each park has exactly **4** sections
- Transportation's Type is either **Shuttle** or **Monorail**
- Each Food & Beverage has only **1** Vendor
- Food & Beverage's Type is either **Quick-Service** or **Sit-Down**
- Live Entertainment's Type is either **Show** or **Parade**
- Photo Booth's Event ID is either **Character Experience ID** or **Attraction ID**
- Hotel's Room Type is either **standard**, **double**, or **premium**
- An Attraction ID is preceded by the character '**A**' in the database, and a Character experience ID is preceded by the character '**C**' in the database.

2.3. ER Diagram



3. Relational Mapping

PARK (<u>Location_ID</u>, City, State, Zip_Code)

SECTION (<u>Name</u>, Park_ID)
Foreign Key: **Park_ID** references PARK

TRANSPORTATION (<u>Vehicle_ID</u>, Type, Park_Section)
Foreign Key: **Park_Section** references SECTION

EMPLOYEE (<u>Employee_ID</u>, SSN, Name, Phone, Address, Department_ID)
Foreign Key: **Department_ID** references DEPARTMENT

DEPARTMENT (Dept_ID, DName, Description, Phone, Park_ID)
Foreign Key: **Park_ID** references PARK

ATTRACTION (Attraction_ID, Name, Description, Park_Section)
Foreign Key: **Park_Section** references SECTION

FOOD_BEVERAGE (F&B_ID, Vendor, Description, Type, Park_Section)
Foreign Key: **Park_Section** references SECTION

MERCHANDISE (Store_ID, Description, Park_Section)
Foreign Key: **Park_Section** references SECTION

ITEM (Item_ID, Price, Manufacturer)

CHARACTER_EXPERIENCE (Exp_ID, Start_Time, End_Time, Description, Park_Section)
Foreign Key: **Park_Section** references SECTION

LIVE_ENTERTAINMENT (Show_ID, Start_Time, End_Time, Type, Park_Section)
Foreign Key: **Park_Section** references SECTION

PHOTO_BOOTH (Booth_ID, Event_ID, Park_Section)
Foreign Key: **Event_ID** references Attraction and Character_Experience and **Park_Section** references SECTION

FIRST_AID (Location_ID, Park_Section)
Foreign Key: **Park_Section** references SECTION

TICKET_INFO (Ticket_ID, Holder_Name, Holder_Phone, Holder_Addr, Holder_Age, Visit_Date, Park_ID)
Foreign Key: **Park_ID** references PARK

HOTEL (Room_Num, Guest_Name, Guest_Ticket_ID, Total_Guests, Check_In_Date, Check_Out_Date, Room_Type, Park_ID)
Foreign Key: **Park_ID** references PARK

SELLS_MERCH (Store_ID, Item_ID)
Foreign Key: **Store_ID** references MERCHANDISE and **Item_ID** references ITEM

Primary key types:

- LocationID char(5)
- DeptID char(10)
- empID char(9)
- Name varchar(10)
- AttractionID char(10)
- FB_ID char(10)
- ShowID char(10)
- expID char(10)
- FirstAid char(10)
- boothID char(10)
- storeID char(10)

Aish:

- Employee
- Attraction
- Character Experience
- Merchandise
- Hotel

Gabi:

- Department
- Food and Beverage
- First Aid
- Item
- Transport

Jesse:

- Section
- Live Entertainment
- Photo Booth
- Ticket
- Sells merch