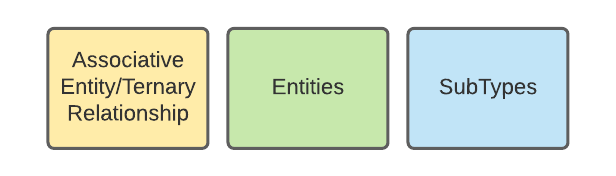
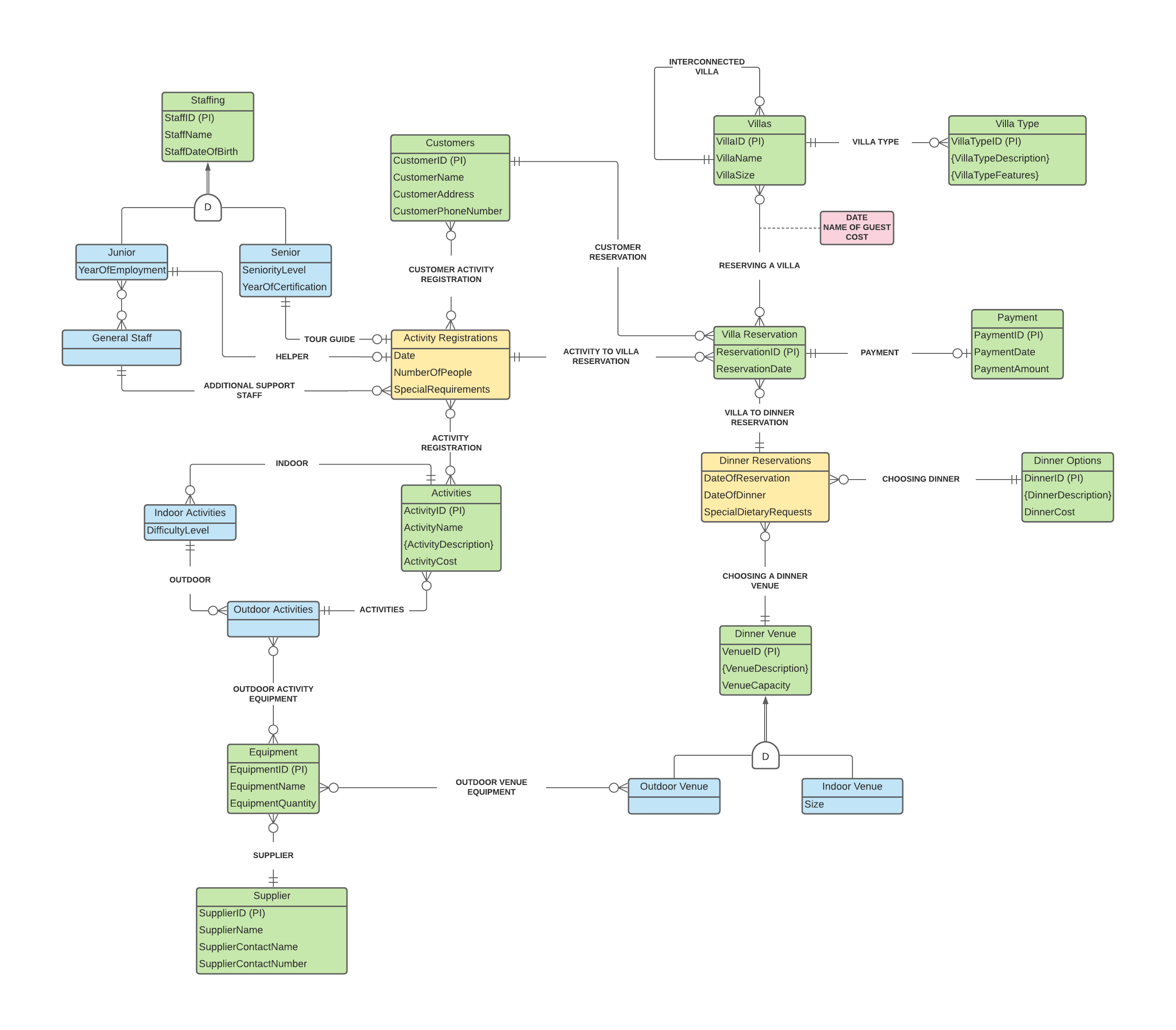
**COMP1350 2020 – ASSIGNMENT ONE**

|  |
| --- |
| **Student ID: 46410961**  **Student Name: AVA GARDINER**  **Tutor’s Name: Hijab Alavi**  **Class Number: COMP1350/S2/Day/Practical\_1/27** |



**Assumptions, if any:**

* **CUSTOMER -> VILLA RESERVATION**: **One to Many Relationship**, because multiple Customers can make a Villa Reservation.
* **VILLA RESERVATION -> VILLA**: **Many to Many Relationship**, because there can be multiple Villa Reservations for Multiple Villas.
* **VILLA -> VILLA TYPE**: **One to Many Relationship**, because there can be multiple Villas for each Villa Type, however each Villa can only have one Villa Type.
* **VILLA RESERVATION -> PAYMENT**: **One to One Relationship**, A single Payment with the cost of the Villa Reservations, the cost of all activity bookings, and the cost of dining from all dinner bookings can be made at the end for each Villa Reservation.
* **VILLA RESERVATION -> ACTIVITY REGISTRATION**: **One to Many Relationship**, because each Villa Reservation can store multiple Activity Registrations.
* **CUSTOMER –> ACTIVITY REGISTRATION**: **Many to Many Relationship**, because multiple Customers can register for multiple Activities.
* **ACTIVITY REGISTRATION -> ACTIVITIES**: **Many to Many Relationship**, because there can be multiple Activity Registrations for multiple Activities.
* **OUTDOOR ACTIVITIES-> EQUIPMENT**: **Many to Many Relationship**, because multiple types of Equipment can be used at multiple Outdoor Activities.
* **SUPPLIER -> EQUIPMENT**: **One to Many Relationship**, because one Supplier can supply multiple types of Equipment.
* **OUTDOOR VENUE -> EQUIPMENT**: **Many to Many Relationship**, because multiple Outdoor Venues can use multiple Equipment.
* **VILLA RESERVATION -> DINNER RESERVATION**: **One to Many Relationship**, because each Villa Reservation can hold multiple Dinner Reservations.
* **DINNER RESERVATION -> DINNER OPTION**: **One to Many Relationship**, because there are many Dinner Options to choose from, but Customers can only choose one Dinner Option when making a dinner reservation.
* **DINNER RESERVATION -> VENUE**: **One to Many Relationship**, because there are many Venues to choose from, but Customers can only choose one Venue when making a Dinner Reservation.
* **ACTIVITY RESERVATION -> SENIOR STAFF**: **One to One Relationship**, because every Activity Registration gets one tour guide, which is a Senior Staff.
* **ACTIVITY RESERVATION -> JUNIOR STAFF:** **One to One Relationship**, because every Activity Registration gets one helper, which is a Junior Staff.
* **ACTIVITY REGISTRATION -> GENERAL STAFF**: **One to Many**, because every Activity Registration can have multiple Support Staff.
* **VILLA -> VILLA**: **One to Many Relationship**, because there are many Villas, but some Villas could be interconnected to one other Villa.
* **ACTIVITIES -> INDOOR ACTIVITIES**: **One to Many Relationship**, because there can be multiple Indoor Activities to one Activity.
* **OUTDOOR ACTIVITIES -> ACTIVITIES**: **One to Many Relationship**, because there can be multiple Outdoor Activities to one Activity.
* **INDOOR ACTIVITIES -> OUTDOOR ACTIVITIES**: **One to Many Relationship**, because some Activities could be both Indoor and Outdoor.

**TASK 2: LOGICAL TRANSFORMATION**

**STEP ONE:** Strong Entities

* Customers (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber)
* Villa Reservation (**ReservationID (PK)**, ReservationDate)
* Payment (**PaymentID (PK)**, PaymentDate, PaymentAmount)
* Dinner Options (**DinnerID (PK)**, DinnerCost)
* Dinner Venue (**VenueID (PK),** VenueCapacity)
* Equipment (**EquipmentID (PK)**, EquipmentName, EquipmentQuantity)

**STEP TWO:** Weak Entities

* No weak entities.

**STEP THREE:** One to One Relationship

* Villa Reservation (**ReservationID (PK)**, ReservationDate, **PaymentID (FK)**)

**STEP FOUR:** One to Many relationship

* Customers (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber, **ReservationID (FK)**)

**STEP FIVE:** Many-Many relationship

* No Many to Many Relationship

**STEP 6:** Multi-valued attributes

* DinnerDescription (**DinnerID (PK, FK)**, DinnerName (PK))
* VenueDescription (**VenueID (PK, FK)**, VenueName (PK))

**STEP 7:** Associative Entity/Ternary Relationship

* Dinner Reservations (**ReservationID (PK, FK)**, **DinnerID (PK, FK)**, **VenueID (PK, FK),** DateOfReservation, DateOfDinner, SpecialDietaryRequests.)

**STEP 8a:** Works for total/partial; overlap/disjoint- inherits just PK

* Outdoor Venue (**OutdoorVenueID (PK)**)
* Indoor Venue {**IndoorVenueID (PK)**, Size)

**REPEAT STEP 2-7**

**STEP 2:** Weak Entities

* No Weak entities

**STEP 3:** One to One Relationship

* No One to 0ne Relationship

**STEP 4:** One to Many Relationship

* No One to Many Relationship

**STEP 5:** Many to Many Relationship

* Dinner Venue (**Venue ID (PK, FK)**, VenueCapacity, **EquipmentID (FK)**)

**STEP 6:** Multi-valued Attributes

* No multi-valued attributes

**STEP 7:** Associative Entity/Ternary Relationship

* No associative/ternary relationship

**FINAL TABLE LIST**

* Customers (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber, **ReservationID (FK)**
* Villa Reservation (**ReservationID (PK)**, ReservationDate, **PaymentID (FK)**)
* Payment (**PaymentID (PK)**, PaymentDate, PaymentAmount)
* Dinner Options (**DinnerID (PK)**, DinnerCost)
* Dinner Venue (**Venue ID (PK, FK)**, VenueCapacity, **EquipmentID (FK)**)
* Equipment (**EquipmentID (PK)**, EquipmentName, EquipmentQuantity)
* DinnerDescription (**DinnerID (PK, FK)**, **DinnerName (PK)**)
* VenueDescription (**VenueID (PK, FK)**, **VenueName (PK)**)
* DinnerReservation (**ReservationID (PK, FK)**, **DinnerID (PK, FK)**, **VenueID (PK, FK),** DateOfReservation, DateOfDinner, SpecialDietaryRequests.)
* Outdoor Venue (**OutdoorVenueID (PK)**)
* Indoor Venue **IndoorVenueID (PK)**, Size)

**APPLICATION OF 8b, 8c, 8d**

8b: Only applies for total

* Outdoor Venue (**OutdoorVenueID (PK)**, VenueCapacity)
* Indoor Venue (**IndoorVenueID (PK)**, VenueCapacity, Size)

8c: Only applies for Disjoint

* Venue (**VenueID (PK)**, VenueCapacity, Size, VenueType)

8d: Only applies for Overlap and the Constraint is Overlap

* N/A



**TASK 3: NOMALISATION**