

# Circus Database Project

Submitted for:

Course:	CSC 426: Principles of Database Systems
Section:	A
Instructor:	Dr. Pierre Akiki

Submitted by:

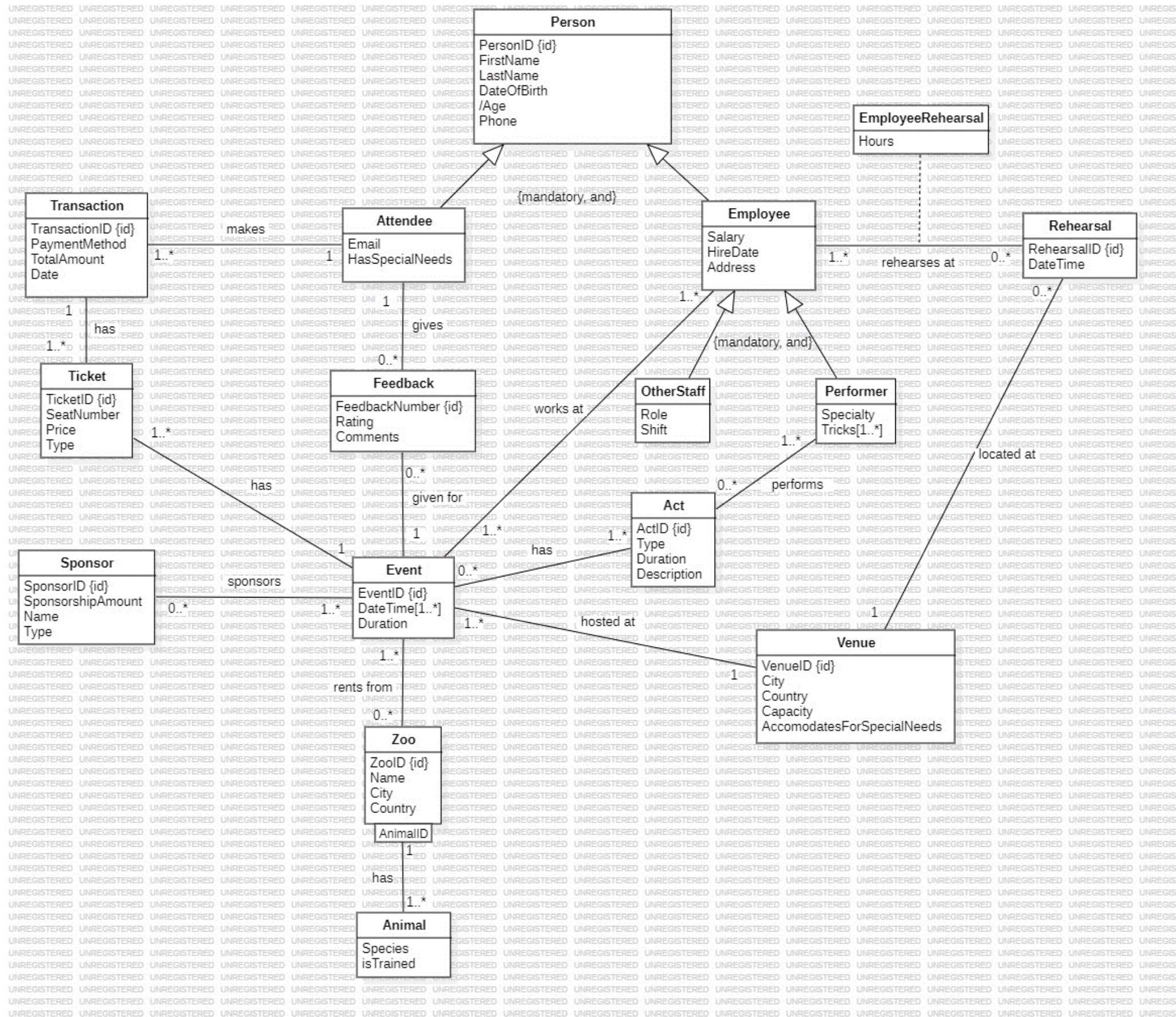
Student ID	Name	Major
20221450	Hovig Chavoushian	Computer Science
20221700	Serge Roussialian	Computer Science
20201330	Joey Bader	Computer Science

## Project Description

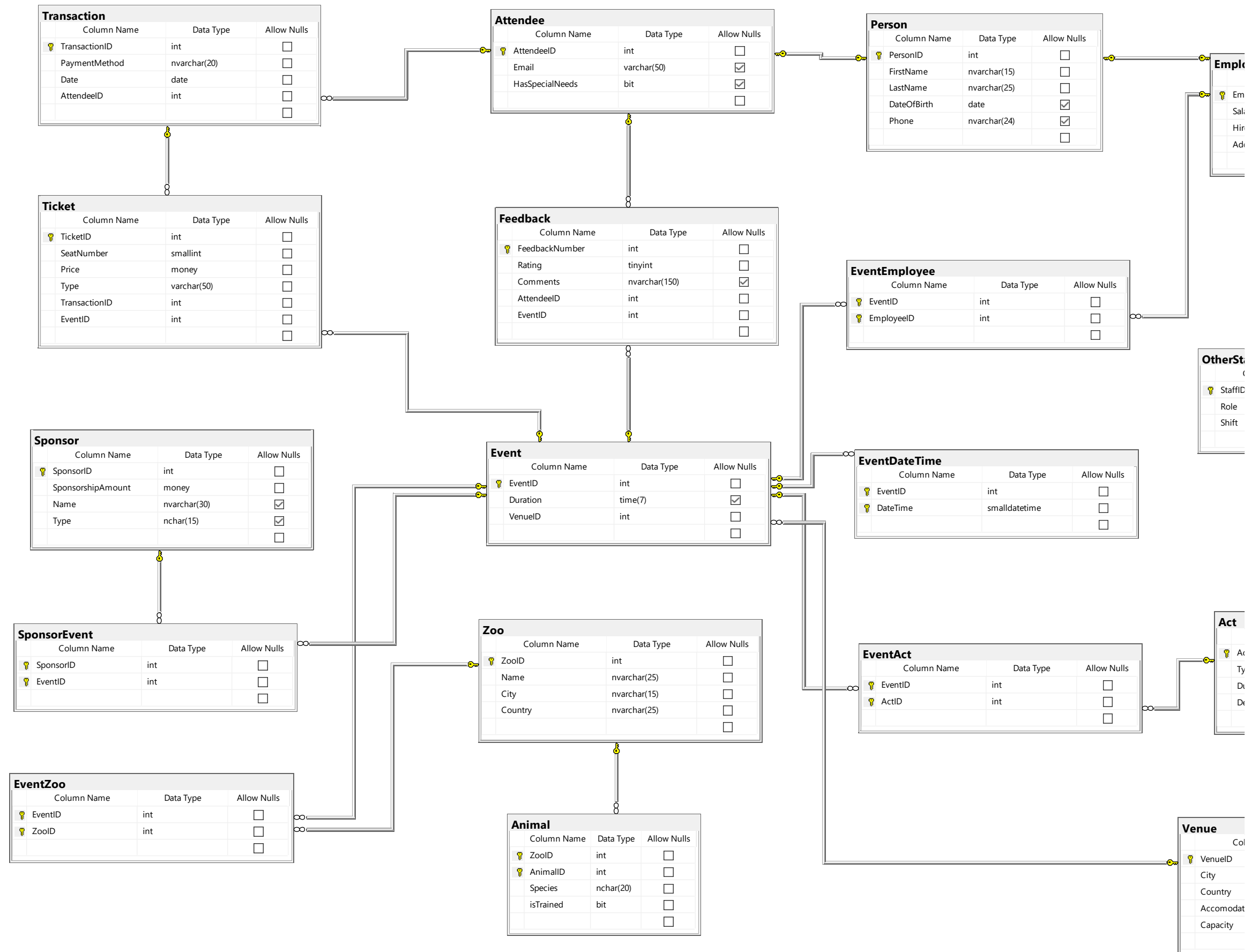
<b>Entity</b>	<b>Description</b> (what does the entity represent and how is it related to other entities) – 30 words max for each entity
<b>Person</b>	Base entity for the types of people we have (Attendees and Employees).
<b>Attendee</b>	Person attending the circus Event. Inherits from Person, makes a Transaction, and gives Feedback.
<b>Employee</b>	Members working in the circus, categorized as Performers and Other Staff. Inherits from Person, works at an Event, and rehearses during Rehearsals.
<b>Performer</b>	Circus artists with specialized skills. Inherits from Employee and performs in Acts.
<b>Other Staff</b>	Other employees such as: directors, lighting technicians, maintenance staff, etc. Inherits from Employee.
<b>Transaction</b>	Records of financial transaction from Attendees buying Tickets.
<b>Ticket</b>	Ticket information for Event admission.
<b>Event</b>	Circus show at a specific Venue. Involves different Acts and renting Animals from a Zoo.
<b>Sponsor</b>	An organization or an individual person sponsoring the Event.
<b>Feedback</b>	Feedback provided by Attendees after an Event.
<b>Zoo</b>	Location where the circus can rent Animals from.
<b>Animal</b>	Animals associated with a specific Zoo. Weak entity to Zoo.
<b>Venue</b>	Location where circus Events and Rehearsals are held.

<b>Act</b>	Specific performances during circus Events performed by Performers.
<b>Rehearsal</b>	Practice sessions for Employees at a Venue.

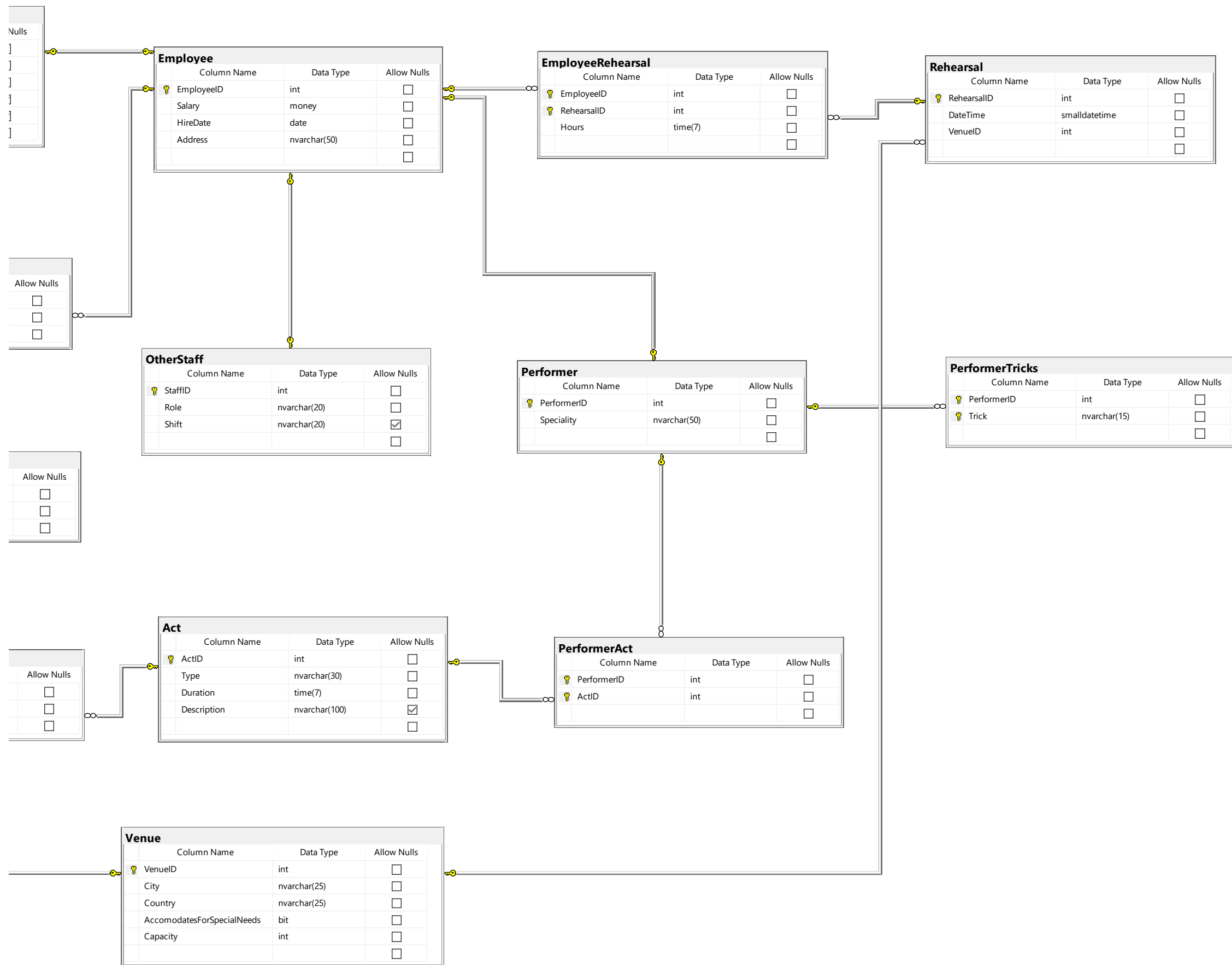
## Enhanced Entity Relationship (EER) Model



## Relational Model







Queries

Query 1: Retrieves the first names and the ticket price of all the attendees who attended an event that had a duration of 1 hour.

/\* Retrieve the first names + the ticket price of all the attendees who attended an event that had a duration of 1 hour\*/

SELECT Person.FirstName, Ticket.Price

FROM Person Inner Join Attendee ON Person.PersonID = Attendee.AttendeeID

Inner Join [Transaction] ON Attendee.AttendeeID = [Transaction].AttendeeID

Inner Join Ticket ON [Transaction].TransactionID = Ticket.TransactionID

Inner Join [Event] ON Ticket.EventID = [Event].EventID

WHERE [Event].Duration = '1:00:00'

150 %

ResultsMessages

	FirstName	Price
1	Alice	20.00
2	Bob	30.00
3	David	20.00
4	David	20.00
5	David	20.00
6	David	20.00
7	Emily	20.00
8	Grace	20.00
9	Henry	45.00
10	Isabella	45.00
11	Kelly	20.00
12	Leo	20.00
13	Mia	20.00
14	Noah	20.00
15	Olivia	20.00
16	Paul	20.00
17	Quinn	20.00
18	Joey	45.00
19	Taylor	45.00

Query 2: Retrieves the EmployeeID, name, age, and phone number of all employees who had rehearsals in venue 20 and have a phone number and date of birth.

/\* Get the EmployeeID, Name and age of all employees that have a phone number and a date of birth that have done rehearsals in venue 20 \*/

Select Employee.EmployeeID, Name = ISNULL(Person.FirstName, '') + ' ' + ISNULL(Person.LastName, ''),Person.Phone, Age = DATEDIFF(year, Person.DateOfBirth, GETDATE())

FROM Person Inner Join Employee ON Person.PersonID = Employee.EmployeeID

Inner Join EmployeeRehearsal ON Employee.EmployeeID = EmployeeRehearsal.EmployeeID

Inner Join Rehearsal ON EmployeeRehearsal.RehearsalID = Rehearsal.RehearsalID

WHERE Rehearsal.VenueID = 20 and Person.DateOfBirth is not null and Person.Phone is not null

50 %

ResultsMessages

	EmployeeID	Name	Phone	Age
1	19	Sam Turner	24385274	13
2	20	Taylor Scott	90325838	35
3	22	Wendy Rodriguez	87424285	21

Query 3: Lists all events along with their sponsor details regardless of whether or not they have a sponsor

/\* Show all events along with their sponsor details regardless of whether or not they have a sponsor \*/

SELECT Event.\*, Sponsor.\*

FROM [Event] Left Join SponsorEvent ON [Event].EventID = SponsorEvent.EventID

Left Join Sponsor ON SponsorEvent.SponsorID = Sponsor.SponsorID

50 %

ResultsMessages

	EventID	Duration	VenueID	SponsorID	SponsorshipAmount	Name	Type
1	1	01:00:00.0000000	20	1	1000.00	MegaBox	Organization
2	2	01:00:00.0000000	21	1	1000.00	MegaBox	Organization
3	3	01:00:00.0000000	21	2	500.00	Dwayne Johnson	Individual
4	4	01:30:00.0000000	10	2	500.00	Dwayne Johnson	Individual
5	5	01:30:00.0000000	10	2	500.00	Dwayne Johnson	Individual
6	6	01:00:00.0000000	30	NULL	NULL	NULL	NULL
7	7	01:00:00.0000000	31	NULL	NULL	NULL	NULL
8	8	01:00:00.0000000	40	3	1500.00	NULL	Anonymous
9	9	01:00:00.0000000	50	1	1000.00	MegaBox	Organization
10	10	01:00:00.0000000	60	4	200.00	NotMicrosoft	Organization
11	11	01:00:00.0000000	61	4	200.00	NotMicrosoft	Organization
12	12	01:00:00.0000000	70	NULL	NULL	NULL	NULL
13	13	01:30:00.0000000	80	1	1000.00	MegaBox	Organization
14	13	01:30:00.0000000	80	2	500.00	Dwayne Johnson	Individual
15	13	01:30:00.0000000	80	3	1500.00	NULL	Anonymous

Query 4: Retrieves the name, city, and country of all Zoos that have Tigers or Lions

```
/* Retrieve the names, city, and country of all Zoos that have Tigers or Lions */  
  
SELECT distinct Zoo.[Name], Zoo.City, Zoo.Country  
FROM Zoo Inner Join Animal ON Zoo.ZooID = Animal.ZooID  
WHERE Animal.Species In ('Tiger', 'Lion')
```

	Name	City	Country
1	Animal City	Beirut	Lebanon
2	Chapultepec Zoo	Mexico City	Mexico
3	San Diego Zoo	San Diego	United States
4	Taronga Zoo	Sydney	Australia

Query 5: Retrieves the 10 lowest ratings with their comments (among all events) along with the phone number and email (regardless if null or not) of the attendee who gave that rating.

```
/* Retrieve the 10 lowest ratings with their comments (among all events) along with (if possible) the phone number and email of the attendee who gave that rating */  
  
SELECT TOP 10 Rating, Comments, Attendee.AttendeeID, Phone, Email, EventID  
FROM Person Inner Join Attendee ON Person.PersonID = Attendee.AttendeeID  
Inner Join Feedback ON Attendee.AttendeeID = Feedback.AttendeeID  
ORDER BY Feedback.Rating asc
```

	Rating	Comments	AttendeeID	Phone	Email	EventID
1	1	Snooze fest tbb	13	81947562	Thompson@gmail.com	7
2	2	Boringg, if that's what it takes to be a performer t...	13	81947562	Thompson@gmail.com	12
3	2	Why was there barely any food??	3	NULL	NULL	2
4	2	NULL	6	NULL	NULL	4
5	2	Security threw me out when I started singing. Ho...	3	NULL	NULL	11
6	3	NULL	11	NULL	NULL	12
7	3	NULL	9	90562473	Isabella@hotmail.com	8
8	3	NULL	7	32568914	NULL	5
9	3	Road to get to the venue was SUPER confusing	8	NULL	NULL	6
10	3	Great Show, expected it to last longer though.	1	73821546	NULL	1

Query 6: Fetches how many seats are left in the venue that’s hosting the event with EventID 4 and whether or not it accomodates for special needs.

```
/* Find how many seats are left in the venue thats hosting the event with eventID 4 and whether it accomodates for special needs or not*/  
  
SELECT [Event].EventID, (Venue.Capacity - COUNT(*)) as SeatsAvailable, Venue.AccommodatesForSpecialNeeds  
FROM Person Inner Join Attendee ON Person.PersonID = Attendee.AttendeeID  
Inner Join [Transaction] ON Attendee.AttendeeID = [Transaction].AttendeeID  
Inner Join Ticket ON [Transaction].TransactionID = Ticket.TransactionID  
Inner Join [Event] ON Ticket.EventID = [Event].EventID  
Inner Join Venue ON [Event].VenueID = Venue.VenueID  
WHERE [Event].EventID = 4  
GROUP BY [Event].EventID, Venue.Capacity, AccomodatesForSpecialNeeds
```

	EventID	SeatsAvailable	AccommodatesForSpecialNeeds
1	4	144	1



**Query 7: Retrieves the AttendeeID along with their first names, the number of tickets they bought in their transaction, and the total price of said transaction.**

```
/* Retrieve the AttendeeID along with their first names, the number of tickets they bought in their transaction and the total price of said transaction. */  
  
SELECT A.AttendeeID,P.FirstName, COUNT(*) as NumberOfTicketsBought, SUM(Ticket.Price) as TotalPrice  
FROM Person P Inner Join Attendee A ON P.PersonID = A.AttendeeID  
Inner Join [Transaction] T ON T.AttendeeID = A.AttendeeID  
Inner Join Ticket ON Ticket.TransactionID = T.TransactionID  
GROUP BY A.AttendeeID, P.FirstName
```

50 %

ResultsMessages

	AttendeeID	FirstName	NumberOfTicketsBought	TotalPrice
1	1	Alice	1	20.00
2	2	Bob	1	30.00
3	3	Carla	1	25.00
4	4	David	4	80.00
5	5	Emily	1	20.00
6	6	Frank	3	60.00
7	7	Grace	1	20.00
8	8	Henry	1	45.00
9	9	Isabella	1	45.00
10	10	Jack	1	20.00
11	11	Kelly	4	80.00
12	12	Leo	1	20.00
13	13	Mia	1	20.00
14	14	Noah	1	20.00
15	15	Olivia	1	20.00
16	16	Paul	1	20.00
17	17	Quinn	1	20.00
18	20	Taylor	1	45.00
19	22	Wendy	1	20.00

**Query 8: Retrieves the count of each rating recieved among all events**

```
/* Retrieve the count of each rating recieved among all events */  
  
SELECT Rating, COUNT(*) as [Count]  
FROM Feedback  
GROUP BY Rating
```

50 %

ResultsMessages

	Rating	Count
1	1	1
2	2	4
3	3	7
4	4	7
5	5	3