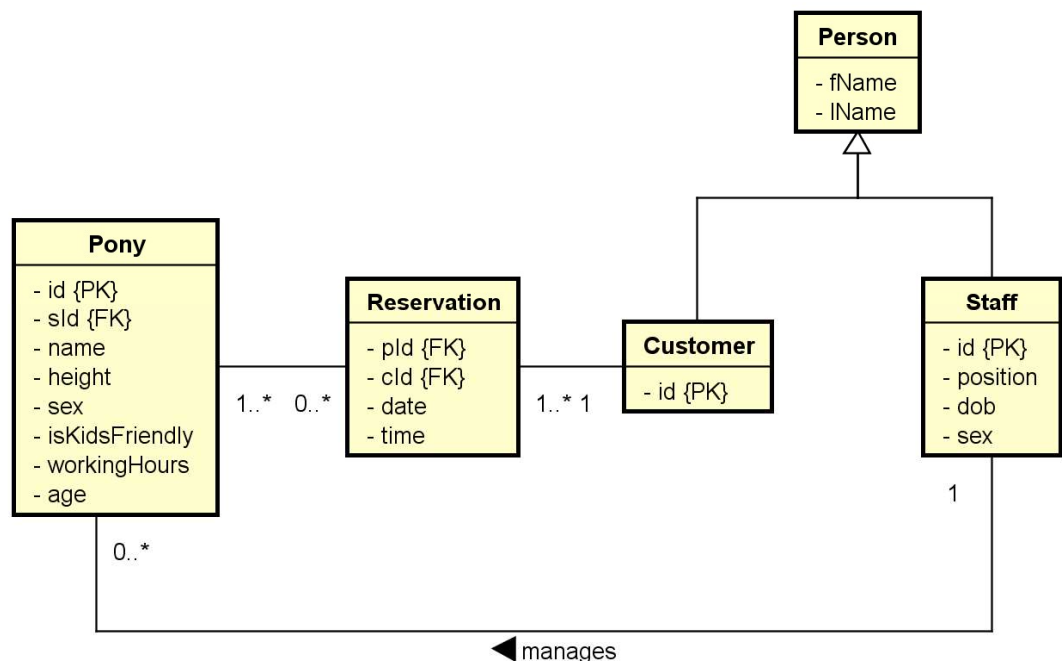


# Pony-riding

## 1. Business rules

- A pony-ride database needs to store information about ponies (id, name, height, sex, isKidsFriendly, workingHours, age, staffId), reservations(date, time, customersId, ponyId), customers(id, fname, lname) and staff(id, position, date of birth, sex)
- Each reservation has one customer and one or more ponies assigned to it; one customer can have many reservations; one pony can have up to 3 reservations per day. A staff with the right position has at least one pony assigned to him/her.



## 2.

### EER diagram

PONY	
NAME	DOMAIN
Id	INT
SId	INT
Name	VARCHAR(20)
Height	DECIMAL [5 [,2]]
Sex	CHAR(1) CHECK (VALUE IN ('F' OR 'M'))
IsKidsFriendly	BOOLEAN
WorkingHours	TIME
Age	INT

## 3. Logical database design

- Pony (id, sld, name, height, sex, isKidsFriendly, workingHours, age)  
PK: id  
FK: sld
- Person(fname, lname)
- Customer (id, fname, lname)  
PK: id
- Staff(id, fname, lname, position, dob, sex)  
PK: id
- Reservation(pld, cld, date, time)  
PK: pld, cld, date  
FK: pld, cld

#### 4. CREATE TABLE Pony (

```

Id INT PRIMARY KEY,
sld INT NOT NULL,
name VARCHAR(20),
height DECIMAL [5 [,2]],
sex CHAR(1) CHECK (VALUE IN ('F' OR 'M')),
isKidsFriendly BOOLEAN,
workingHours TIME,
age INT,
CONSTRAINT pony_sld FOREIGN KEY (sld) REFERENCES Staff(id)
);

```

#### CREATE TABLE PERSON (

```

fname VARCHAR(10),
lname VARCHAR(15)
);

```

#### CREATE TABLE Staff (

```

Id INT PRIMARY KEY,
Position VARCHAR(15),
DoB INT NOT NULL,
sex CHAR(1) CHECK (VALUE IN ('F' OR 'M')),
fname VARCHAR(10),
lname VARCHAR(15)
);

```

```
CREATE TABLE Customer (  
  Id INT PRIMARY KEY,  
  fname VARCHAR(10),  
  lname VARCHAR(15)  
);
```

```
CREATE TABLE Reservation (  
  pld INT NOT NULL,  
  cid INT NOT NULL,  
  date DATE,  
  time TIME,  
  CONSTRAINT reservation_pk PRIMARY KEY (pld, cid, date),  
  CONSTRAINT reservation_pld FOREIGN KEY (pld) REFERENCES Pony(id),  
  CONSTRAINT reservation_cld FOREIGN KEY (cid) REFERENCES Customer(id)  
);
```

```
INSERT INTO Pony VALUES(001, 001, 'PinkiePie', 121.22, 'F', true, 6, 2);  
INSERT INTO Pony VALUES(002, 003, 'RainbowDash', 90.99, 'F', true, 8, 4);  
INSERT INTO Pony VALUES(003, 003, 'Chealsie', 89.01, 'M', true, 4, 6);  
INSERT INTO Customers VALUES(001, 'Leonardo', 'da Vinci');  
INSERT INTO Customers VALUES(002, 'Mona', 'Lisa');  
INSERT INTO Customers VALUES(003, 'James', 'Bond');  
INSERT INTO Staff VALUES(001, 'Dominika', 'Kubicz', 'instructor', 29-04-1998, 'F');  
INSERT INTO Staff VALUES(002, 'Daniela', 'Koch', 'manager', 24-12-1998, 'F');  
INSERT INTO Staff VALUES(003, 'Jon', 'Snow', 'stable boy', 2-04-1992, 'M');  
INSERT INTO Reservation VALUES(001, 002, 16-04-2018, 14:00);  
INSERT INTO Reservation VALUES(002, 001, 16-04-2018, 14:00);  
INSERT INTO Reservation VALUES(001, 003, 16-04-2018, 18:00);
```

Show the names and heights of all kids friendly female ponies:

```
SELECT name, height
FROM Pony
WHERE sex='F' AND isKidsFriendly=true
ORDER BY name;
```

Show all workers that are taking care of ponies, order them descending by the number of ponies they take care of:

```
SELECT Staff.id, fName, lName, COUNT(Pony.id) as numberOfPonies
FROM Staff, Pony
GROUP BY Staff.id
HAVING numberOfPonies>0
ORDER BY numberOfPonies DESC;
```

Show all names of ponies that are managed by Dominika:

```
SELECT name
FROM Pony
WHERE sId IS
  (SELECT id
   FROM Staff
   WHERE name='Dominika');
```

Show all customers that have a reservation on 16-04-2018:

```
SELECT Customer.id, fName, lName
FROM Customer
INNER JOIN Reservation ON Customer.id=Reservation.cId
WHERE date=16-04-2018
ORDER BY lName;
```