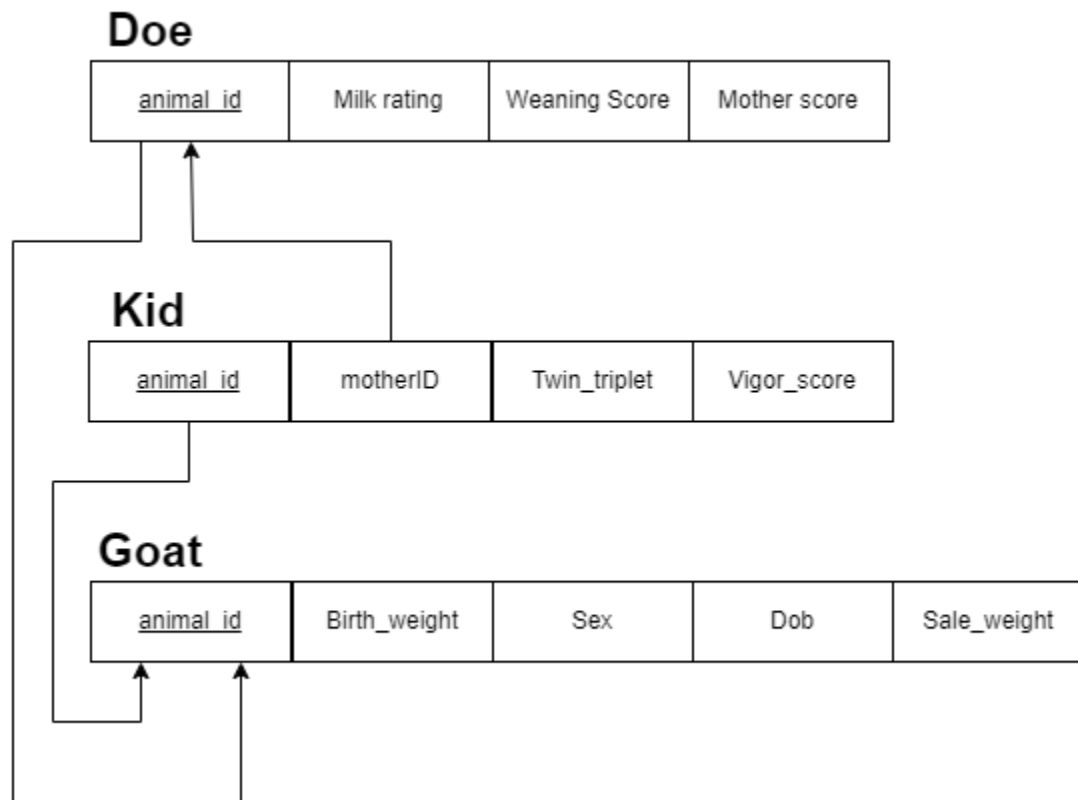


Patrick Schaeffer, Nick Valentino, Kayley Harnett



2. The tables are in BCNF, they only have atomic values making it 1NF, all attributes are dependent on the primary key making it 2NF, and there are no transitive dependencies so they are also 3NF. Because the primary key is the only candidate key, they are BCNF.

3. The views that we will need for the Doe Rating will have the information of each DOE: what their milk rating was, their weaning score, and their mother score. Relations will be from the children: their score from birth_weight, vigor_score, and if they were a twin.

The views that we will need for the Goat Astrology will contain the animal ID, DOB, and AVG(Sale_weight).

4.

Mother Score:

```
CREATE VIEW mother_score AS
SELECT Doe.animal_ID, Doe.Birth_weight, Milk_rating, Weaning_score,
Mother_score, Kid.Birth_weight, Vigor_score, Twin_triplet
FROM Doe NATURAL JOIN Goat, Kid NATURAL JOIN Goat
WHERE Doe.animal_id = Kid.motherID;
```

Goat astrology:

```
CREATE VIEW      goat_astrology
AS SELECT        Dob, AVG(Sale_weight)
FROM             GOAT
GROUP BY         Dob;
```

5. The mother_score view defined in #4 allows for the implementation of the “View Mother Scores” use case by listing the required information in a single virtual table. The goat_astrology view enables the “View Astrology” use case as it displays the average sale weight that corresponds with each day that a goat was born. Using this table, it will be easy to see which birth dates have higher sale weights and which birth dates have lower sale weights.

