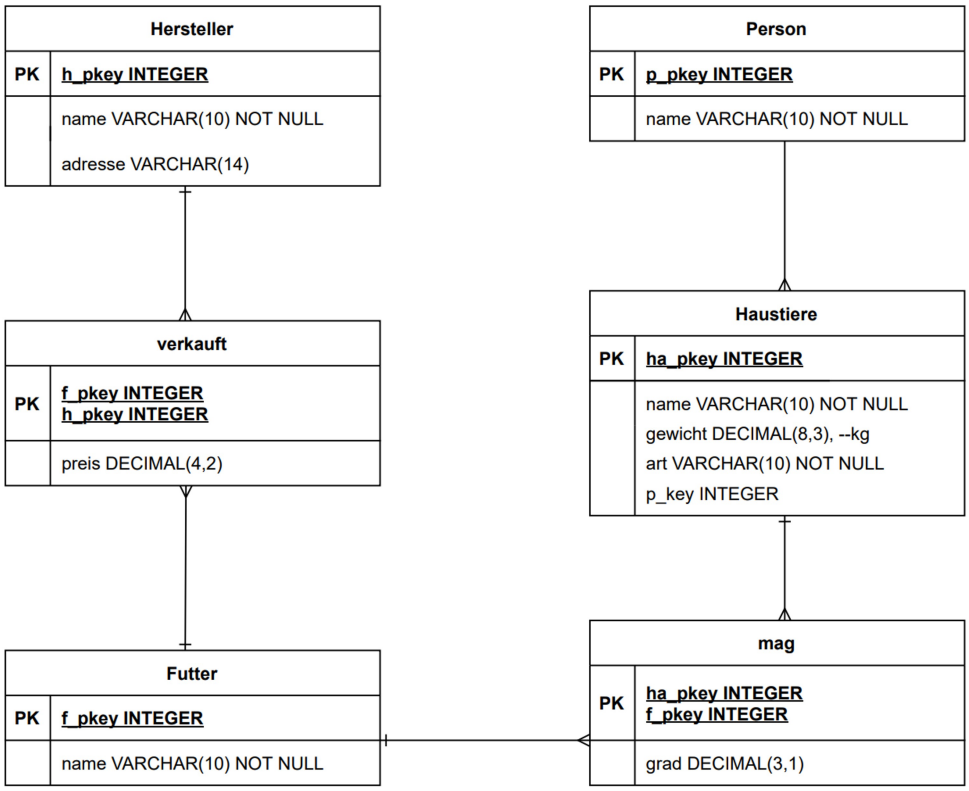
Serie 9

Donnerstag, 28. April 2022 16:18

Alan Andrey, 20-124-525
Sandro Beer, 19-535-006

Aufgabe 1

Laden Sie von Ilias die Setup-Datei `setup.sql` herunter und führen Sie das Script gegen Ihre Datenbank aus. Zeichnen Sie die Situation als **Krähenfuss-Diagramm**!



Aufgabe 2

Bauen Sie eine Sicht (View) *LieblingsFutter*, welche das beliebteste (im arithmetischen Mittel) Futter (Namen) je Tierart inklusive Hersteller und Preis angibt. Beim Hersteller soll aber gegebenenfalls nur der/die billigste/billigsten Hersteller berücksichtigt werden. Wenn es keinen Hersteller (mehr) gibt, soll das Futter dennoch in der View erscheinen.
Falls es mehrere gleichbewertete Futter gibt, sollen alle gleichbewerteten Futter mit aufgeführt sein. Falls es keine Daten gibt um eine Beliebtheit zu berechnen, soll das Futter ignoriert werden. Beispiel: Futter A (am billigsten von 'Miau' angeboten zum Preis von 13.40), B, C, D, E (am billigsten von 'Wuf' angeboten zum Preis von 15.40) mit mittleren Beliebtheiten bei Hunden 1.1, 1.3, 2.0, 2.0, 4 und bei Katzen 3.1, 1.4, 0.7, -(mangels Daten nicht ermittelbar), 0.7. Dann soll die Tabelle etwa so aussehen (Sortierung ist irrelevant):

LieblingsFutter			
Tierart	**Futter**	**Hersteller**	**Preis**
Hund	E	Wuf	15.40
Katze	A	Miau	13.40

```
--Aufgabe 2
CREATE VIEW LieblingsFutter AS
    WITH Beliebtheit AS (
        SELECT art AS Tierart,  Futter.name AS Futter, AVG(grad) AS Bewertung
        FROM Haustiere INNER JOIN  mag USING (ha_pkey) INNER JOIN  FUTTER
        USING (f_pkey)
        GROUP BY Tierart, Futter),

        Hilfstable AS (SELECT * FROM Beliebtheit),

        Beliebteste AS (
        SELECT Tierart , Futter
        FROM Hilfstable
            WHERE Bewertung >= ALL(SELECT MAX(Bewertung)
                                   FROM Beliebtheit
                                   WHERE Hilfstable.Tierart = Beliebtheit.Tierart)),
        Hilfstable1 AS ( --Tabelle mit allen Futtersorten und dazugehörigen
        Herstellern
                              --Wir wissen nicht, wie wir Futter ohne Hersteller
                             auch berücksichtigen können
            SELECT Hersteller.name AS Hersteller, preis, Futter.name AS Futter
            FROM Futter INNER JOIN verkauft USING(f_pkey) INNER JOIN
            Hersteller USING(h_pkey)),

        Hilfstable2 AS (SELECT * FROM Hilfstable1),

        Billo_Anbieter AS (
        SELECT Hilfstable1.Futter AS Futter, Hilfstable1.Hersteller AS Hersteller,
        preis
        FROM Hilfstable1
        WHERE preis <= ALL(SELECT MIN(preis)
                           FROM Hilfstable2
                           WHERE Hilfstable1.Futter = Hilfstable2.Futter))

        --Füge nun die Hilfstabellen zusammen
        SELECT Tierart, Futter, Hersteller, Preis
        FROM Beliebteste INNER JOIN Billo_Anbieter USING (Futter);
```

Aufgabe 3

Finden Sie geeignete Daten um Ihre View zu testen und fügen Sie diese in die Tabellen Ihrer Datenbank ein!
Welche möglichen Zustände gibt es? Wie müssen sinnvolle Testdaten daher aussehen? Stellen Sie sicher, dass Ihre eingegebenen Testdaten alle gefundenen Fälle abdecken! Schreiben Sie Ihre diesbezüglichen Überlegungen auf.

Um alle möglichen Fälle abzudecken, müssen wir den Fall abdecken, in dem eine Tierart zwei Futter mag, jedoch das eine mehr als das andere. Des weiteren eine zweite Tierart, die nicht mehr hergestellt wird. Damit decken wir ab, dass diese Tierart überhaupt angezeigt wird und ob Null überhaupt erlaubt ist (das letzte der beiden, haben wir leider nicht hinbekommen).

Aufgabe 4

Finden Sie Datenbankabfragen für die folgenden Fragestellungen:

a) Welcher Hersteller (Name) liefert Futter namens ‘Stroh‘?

```
SELECT Hersteller.name
    FROM Hersteller INNER JOIN verkauft USING (h_pkey) INNER JOIN Futter USING (f_pkey)
    WHERE Futter.name = 'Stroh';
```

b) Gibt es Haustiere ohne Besitzer?

```
SELECT Haustiere.name
    FROM Haustiere
    WHERE ha_pkey IS NULL;
    --Falls Liste Leer ist existieren keine Tiere ohne Besitzer, ansonsten ja
```

c) Gibt es Haustiere, die alle Futter gleichermassen mögen?

```
WITH Haustiere_min_max AS(
    SELECT ha_Pkey,
    MIN(grad) AS hamin,
    CASE WHEN MAX(CASE WHEN grad IS NULL THEN -10000 ELSE 0 END) = 0
   THEN MAX(grad) --Wenn NULL soll wert zu etwas unerreichbarem gewechselt werden
  END AS hamax          --Konnten nicht überprüfen, ob alle Futter "gemocht" werden
    FROM Haustiere INNER JOIN mag USING(ha_pkey)

    GROUP BY ha_pkey)

    SELECT ha_pkey
        FROM Haustiere_min_max
        Where hamin = hamax;
        --Falls Liste Leer ist existieren keine Tiere ohne Besitzer, ansonsten ja
```

d) Welche Haustiere mögen die Futter, die sie probiert und bewertet haben, gleichermassen?

```
WITH Haustiere_min_max AS(
    SELECT ha_Pkey,
    MIN(grad) AS hamin,
    MAX(grad) AS hamax --Hier ignorieren MIN und MAX die NULL werte
    FROM Haustiere INNER JOIN mag USING(ha_pkey)

    GROUP BY ha_pkey)

SELECT ha_pkey
FROM Haustiere_min_max
Where hamin = hamax;
--Falls Liste Leer ist existieren keine Tiere ohne Besitzer, ansonsten ja
```