



Auf dem Server *faulus.inf.unibe.ch* liegen wieder Postgresql-Datenbank “fs21_[Benutzername]_s08” bereit, falls Sie Teile Ihrer Abgaben ausprobieren wollen. Leider beherrscht Postgresql keine Subqueries innerhalb von CHECK-Constraints. Da diese aber für die letzte Aufgabe unumgänglich sind, ist die Benutzung der Datenbank nur zum Ausprobieren von Teilausdrücken geeignet und *nicht* Teil der Hausaufgabe.

Es wird empfohlen die Aufgaben in Reihenfolge abzuarbeiten. Die Aufgaben beziehen sich auf das umseitig abgebildete Krähenfuss-Diagramm.

Aufgabe 1
Beantworten Sie die folgenden Fragen bzgl. des Diagramms! Geben Sie jeweils eine *kurze* Begründung an!

- a) Wie viele Haustiere kann ein Halter haben?
- b) Kann ein Kamin gleichzeitig ein Laptop sein?
- c) Kann man das Lieblingsfutter einer Katze aus der Datenbank ablesen?
- d) Fehlen Verknuepfungsattribute? Sind diese unbedingt nötig? Wie können diese Verknuepfungen umgesetzt werden?
- e) Können mehrere Haustiere den gleichen Lieblingsplatz haben?
- f) Kann ein Lieblingsplatz bei mehren Personen sein?
- g) Kann man das Durchschnittsalter aller Hunde, die von einem bestimmten Aufpasser betreut werden, bestimmen?

Aufgabe 2
Bilden Sie die im Diagramm abgebildete Datenbank direkt auf Tabellen ab. Die Attribute des “Geburtsdatum” (Tag, Monat, Jahr) sollen dabei durch Integer-Werte dargestellt werden. Wählen Sie ansonsten geeignete Typen.

Aufgabe 3
Welche Tabellen-Abhängigkeiten/Integritätsbedingungen ergeben Sich durch

- a) die Vererbungen?
- b) die Beziehungen?
- c) (GebTag, GebMonat, GebJahr) als Geburtsdatum?

Gibt es dadurch entstehende Einfüge-Reihenfolgen oder notwendige Mehrfach-Einfügungen? Hinweis: Zur Vereinfachung dürfen Sie annehmen, dass jeder Monat 31 Tage hat.

Aufgabe 4
Geben Sie die erforderlichen SQL-Befehle an, um die Tabellen mit Konsistenz-Prüfungen (Constraints) in der Datenbank zu erzeugen. Beachten Sie dabei noch die folgenden Bedingungen:

- Wird ein Haustier aus der Tabelle entfernt, soll dessen Halter ebenfalls entfernt werden. Dieser soll jedoch Person bleiben.
- Beziehungen, die nach Entfernung von Entitäten veraltet sind, sollen immer entfernt werden.
- Freiwillige Aufpasser haben einen Stundenlohn von NULL.

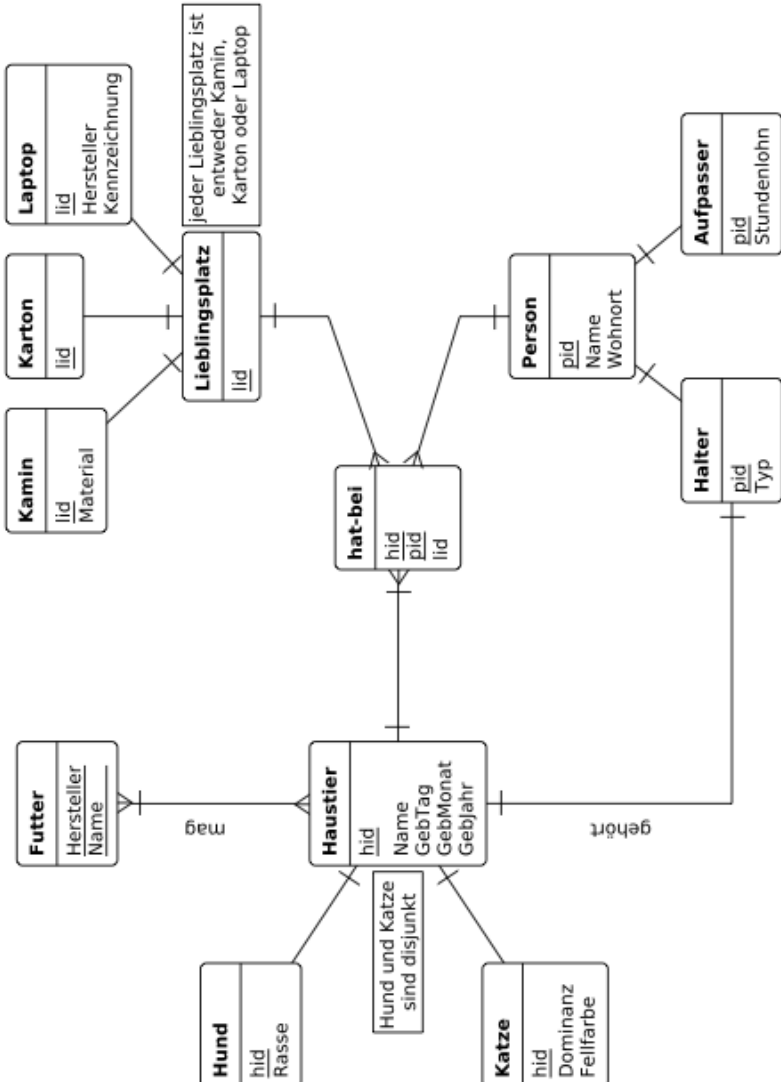
1a.) mehrere
B.) nein
C.) ja siehe mag tabelle
D.) ja zB. Die verknüpfung vom haustier mit seinem Lieblingsfutter ist nicht spezifisch gegeben und haustier personverknüpfung fehlt
E.) ja
F.) nein
G.) ja

2.) Siehe SQL
3.)

- Dominanzgrad von Katzen (0-100%) und Kennzeichnung von Laptops können unbestimmar sein und müssen nicht unbedingt in der Datenbank stehen. Der Rest muss eingetragen werden. (Hinweis: NULL-Werte!)

Geben Sie eine SQL-Datei (.sql) mit den Statements über Ilias ab und auch EIN PDF File mit den Antworten UND den SQL-Statements!
(Klartext-Datei in utf, mit Endung .sql, die Sie in psql direkt mittels \i laden könnten)

Hinweis: Es ist unumgänglich einige Bedingungen mittels “ALTER TABLE name ADD table.constraint” nachträglich einzuführen.
Hinweis 2: Wenn Sie testweise Daten in die Datenbank eingeben, werden normalerweise Konsistenz Checks am Ende jedes SQL-Befehls duchgeführt (auto commit). Sie können dies umgehen indem Sie eine Transaktion mit dem “BEGIN;” Kommando erzeugen und mit “COMMIT;” beenden. Die Constraints werden dann erst am nach dem “commit” geprüft.



2.) Haustier [hid : int , Name : Varchar , Tag : int , Monat : int , Jahr : int , pid : int]
Hund [hid : int , Rasse : Varchar]
Katze [hid : int , Dominanz : Varchar , Fellfarbe : Varchar]
Futler[Hersteller : Varchar , Name : Varchar]
mag [hid : int , Hersteller : Varchar , Name : Varchar]
Poson [pid : serial , name : varchar , Wohnort : Varchar]
Halter [pid : int , Typ : Varchar]
gehört [pid : int , hid : int]
Aufpasser [pid : int , Stundenlohn : int]
Lieblingsplate [lid : serial]
hat-bei [hid : int , pid : int , lid : int]
Kamin [lid : int , Material : Varchar]
Laptop [lid : int , Hersteller : Varchar , Kennzeichnung : Varchar]
Karton [lid : int]

3a.) ein Hund / Katze muss als Haustier eingetragen werden
Ein Aufpasser muss einer Person zugewiesen werden
Ein Karton/Kamin/Laptop muss ein Lieblingsplatz sein
Ein Halter muss eine als eine Person eingetragen sein
Siehe SQL datei für die Umsetzung.

3b.) Jedes Haustier hat mindestens ein Lieblingsfutter

3c.) es kann sein das wir ein geburtsjahr von einem Haustier kennen jedoch nicht der monat oder den tag somit ist es praktisch die drei attribute einzeln zu haben falls man dies jedoch anders realisieren möchte kann man bei allen dreien not null hinzufügen

3d.) man könnte einen ALTER TABLE Haustier ADD CHECK(tag <= 31);

4.) Siehe SQL