-- Noé Stemmer 17 106 956

-- Nina Baumgartner

-- DB Serie 10 Aufgabe 2

CREATE VIEW magOverview AS

SELECT DISTINCT Haustiere.art, Futter.name AS Futter,

(SELECT SUM (mag.grad) FROM mag)

WHERE Haustiere.ha\_pkey=mag.ha\_pkey

AND mag.f\_pkey=Futter.f\_pkey)

AS sum

FROM mag,Haustiere,Futter

WHERE Futter.f\_pkey = mag.f\_pkey

AND Haustiere.ha\_pkey = mag.ha\_pkey;

/\*We get a Relation with three columns "Haustiere.art", "Futter","mag.grad".

he mag.grad is the added value of degree to which the animals of a kind like a

certain food. \*/

CREATE VIEW arithmeticAverage AS

SELECT DISTINCT Haustiere.art, Futter.name AS Futter,

(SELECT AVG(sum)FROM magOverview

WHERE magOverview.art = Haustiere.art

AND magOverview.Futter = Futter.name)

AS average

FROM magOverview, Haustiere, Futter,

ORDER BY average DESC;

/\*We get a Relation with three columns "Haustiere.art", "Futter" and "average".

Average is the average degree to which to which the animals of a kind like a

certain food. Average is ordered by the highest ones first. \*/

CREATE VIEW ranking AS

SELECT arithmeticAverage.Futter, arithmeticAverage.art,

count(arithmeticAverage.average) as rank

FROM arithmeticAverage, arithmeticAverage AS ariAv

WHERE arithmeticAverage.art = ariAv.art

AND arithmeticAverage.average >= ariAv.average

GROUP BY arithmeticAverage.Futter, arithmeticAverage.art;

/\* Reltation that ranks the ordered averages. \*/

CREATE VIEW billigsterHersteller AS

Hersteller := SORT (name, h\_pkey)

verkauft := SORT (preis, h\_pkey)

i:= 1

j:= 1

WHILE (i <= #Hersteller AND <= #verkauft)

IF ( Hersteller [i][h\_pkey] = verkauft [j][h\_pkey]) THEN j = jj

WHILE ( Hersteller[i][h\_pkey] = T[j][h\_pkey] AND j <= #verkauft )

OUTPUT (Hersteller[i],verkauft[j])

j++

j = jj

i++

ELSE IF ( Hersteller[i][h\_pkey] > T[j][h\_pkey] ) THEN

j++

ELSE

i++ ;

-- DB Serie 10 Aufgabe 3

INSERT INTO Person(p\_pkey, name)

VALUES(1, 'Alex')

;

INSERT INTO Haustiere(ha\_pkey, name, gewicht, art, p\_pkey)

VALUES(1, 'Büsi', 00005.000 , 'Vierfarb', 1)

;

INSERT INTO Haustiere

VALUES(2, 'T-Rex', 00010.000 , 'Sennenhund', 1)

;

INSERT INTO Haustiere

VALUES(3, 'Kevin', 05000.000 , 'Weishorn', 1)

;

INSERT INTO Futter(f\_pkey, name)

VALUES(1, 'Katzenraum')

;

INSERT INTO Futter(f\_pkey, name)

VALUES(2, 'Hundetraum')

;

INSERT INTO Futter(f\_pkey, name)

VALUES(3, 'Rinotraum')

;

INSERT INTO Futter(f\_pkey, name)

VALUES(4, 'Trockenfu')

;

INSERT INTO Futter(f\_pkey, name)

VALUES(5, 'Vegie')

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(1, 1, 09.9)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(1, 2, 05.1)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(1, 3, 02.2)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(1, 4, 06.6)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(1, 5, 00.1)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(2, 1, 05.2)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(2, 2, 08.5)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(2, 3, 05.0)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(2, 4, 06.6)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(2, 5, 00.1)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(3, 1, 05.0)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(3, 2, 07.1)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(3, 3, 09.9)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(3, 4, 03.6)

;

INSERT INTO mag(ha\_pkey, f\_pkey, grad)

VALUES(3, 5, 00.1)

;

-- DB Serie 10 Aufgabe 4

-- a

select hersteller.name as Name from hersteller

inner join verkauft using(h\_pkey)

inner join futter using(f\_pkey)

where futter.name ='Stroh';

-- b

with counter as (

select count(\*) as anzahl

from haustiere

where p\_pkey IS NOT NULL)

select case when 0 = anzahl then true

else false end

from counter;

--c

with counter as (

select name

from haustiere

natural join mag

group by name

having min(grad) = max(grad) and count(mag.ha\_pkey) > 1

)

select case when count(name) > 0 then true

else false end

from counter

--d

count(mag.ha\_pkey) > 1

select name

from haustiere

natural join mag

group by name

having min(grad) = max(grad) and count(mag.ha\_pkey) > 1