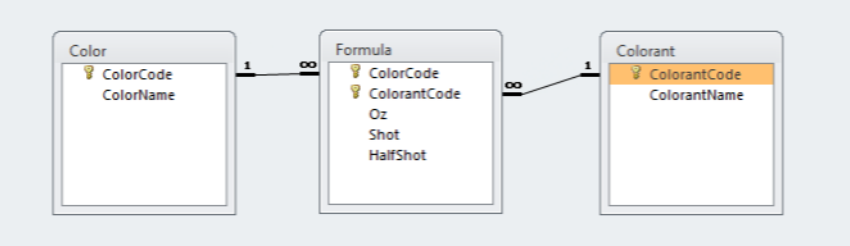
Lab 4- Advanced SQL (Chapter 8)

We have been working with paint formulas using the(oz– 48th – 96th) notation. Up to now, however, the database has been in 1NF. For lab 4, we will be using the database re-done in 3NF. Consider the following ER-Diagram:



1. From Blackboard, download ***PaintFormula3NF.sql*** Open this file in a text editor so you can see how the 3 tables were created, populated, and most importantly, how I constructed my SQL statement to list the formulas. Keep in mind, when you use more than 1 table, you want to NATURAL JOIN them. Just using a comma (,) to separate them generates a CROSS PRODUCT.
2. Now, FTP ***PaintFormula3NF.sql*** over to your turing account. Within your mysql engine, run *source* ***PaintFormula3NF.sql*** to create and populate the Paint Formula database tables.
3. Try running *source* ***PaintFormula3NF.sql*** again. You should expect errors because Constraints have been added to your Database. Let's fix this
    a. Using the appropriate SQL command (***SHOW CREATE TABLE Formula***), identify the name of the Constraints to the *Formula* table. There are 2 constraint names that are needed since there are 2 foreign keys.
    b. Delete these constraints. Recall, you must use the constraint name to do this (your answers in part a):
    ALTER TABLE Formula DROP FOREIGN KEY constraint_name;
    c. Now, to the PaintFormula3NF.sql table explicitly add the 2 Foreign Key Constraints to Formula after the 3 tables have been created. Go to www.w3schools.com and select ***SQL Foreign Key.*** Look at the example on how to simultaneously name the Constraint within the definition of the foreign key.
    d. At the very beginning of your file, BEFORE the DROP TABLE IF EXISTS… add the appropriate lines to
    ALTER TABLE Formula DROP FOREIGN KEY constraint_name;
4. Once PaintFormula3NF.sql has been modified, try running *source PaintFormula3NF.sql* a few times. You should no longer get an error.

5. For the remaining questions, write a script file, *yourLastName****Lab4.sql*** that will execute the SQL required to answer the following questions 5a-5f. Within your

script add comments denoting part a-f for each SQL query – all of these SQL queries will require aggregates. When all of the bugs are out of your SQL, create a script file *yourLastName**Lab4Output.txt*** that shows the execution of your SQL commands. Clean up the text file and submit your modified ***PaintFormulas3NF.sql***, *yourLastName**Lab4.sql***, and *yourLastName**Lab4Output.txt*** on Blackboard.

So you can double check your answers, I have provided the results of my queries at the end of the lab.

a. For each paint Color, display the names of the Colorants in the formula. Display the paint colors in ascending order. Notice that the column header for the Colorant names looks bad so use the alias ***Colorants*** for this field instead.

b. Count all the times a colorant appears in the Formulas. Display the Colorant names in ascending order. Use an alias to display ***Colorant_Count*** for the count field instead (spaces are not allowed in alias' so be sure to include the underscore).

c. Identify all the colorant names that contain a colorant with *Oxide* in the name.

d. Say I have a limited amount of the Magenta colorant and can only support those paint formulas that have shots and half shots (i.e., I cannot produce paints that have a nonzero value for the oz field). Write a script that lists the names of the paints in ascending order that are <u>not</u> affected by this limitation.

e. Identify the total number of ounces of each Colorant from all the paint formulas listed. Recall that the number in the Shot column represents a 48^{th} of an ounce and therefore should be divided by 48. Likewise, the number in the HalfShot column represents a 96^{th} of an ounce and should be divided by 96.

   Consider Navajo Red's brown oxide (I) colorant. The volume is calculated as 2 + 41/48 + 1/96, or 2.865 ounces. This example is only one colorant from one paint. For this query, you must find the sum, or total volume, of each colorant from all six paints. Use the SUM aggregate to accomplish this. Within SUM, use the operators as you do in Java (i.e., + - * /). Use the alias ***Total_Ounces*** for the SUM aggregate.

f. For each Colorant, list all of the names of the paints that use each colorant. For your aggregate use the alias ***Paints.*** Display in ascending order of the ColorantName

Color Names/Formulas	oz.	1/48	1/96
Tangier Island			
B	0	18	0
F	0	10	0
AXX	3	6	0
Barn Red			
B	0	20	0
F	0	40	0
V	0	12	0
X	0	20	0
KX	0	16	0
Navajo Red			
I	2	41	1
R	5	47	1
T	2	8	1
KX	0	47	0
Dusty Mauve			
B	0	19	1
T	0	30	0
V	1	8	0
Silver			
L	0	36	1
E	0	5	1
V	0	5	1
J Pastel			
L	0	16	1
T	0	34	0
V	1	38	0

Outputs for 5a – 5e

a).

ColorName	**Colorants**
BarnRed	**Organic Yellow,Black Oxide,Red Oxide,Titanium White,Magenta**
DustyMauve	**Black Oxide,Medium Yellow,Magenta**
JPastel	**Raw_Umber,Medium Yellow,Magenta**
NavajoRed	**Brown Oxide,Titanium White,Organic Red,Medium Yellow**
Silver	**Phalo Blue,Raw_Umber,Magenta**
TangierIsland	**Organic Yellow,Black Oxide,Red Oxide**

b).

ColorantName	Colorant_Count
Black Oxide	3
Brown Oxide	1
Magenta	4
Medium Yellow	3
Organic Red	1
Organic Yellow	2
Phalo Blue	1
Raw_Umber	2
Red Oxide	2
Titanium White	2

c).

ColorantName
Black Oxide
Yellow Oxide
Red Oxide
Brown Oxide

d).

ColorName
BarnRed
Silver

e).

ColorantName	Total_Ounces
Black Oxide	1.1979
Brown Oxide	2.8646
Magenta	3.3230
Medium Yellow	3.5104
Organic Red	5.9896
Organic Yellow	3.5417
Phalo Blue	0.1146
Raw_Umber	1.1041
Red Oxide	1.0416
Titanium White	1.3125

f).

ColorantName	Paints
Black Oxide	TangierIsland,DustyMauve,BarnRed
Brown Oxide	NavajoRed
Magenta	Silver,BarnRed,DustyMauve,JPastel
Medium Yellow	NavajoRed,DustyMauve,JPastel
Organic Red	NavajoRed
Organic Yellow	TangierIsland,BarnRed
Phalo Blue	Silver
Raw_Umber	Silver,JPastel
Red Oxide	BarnRed,TangierIsland
Titanium White	BarnRed,NavajoRed