SQL Tasks

Web: http://sol.gfxile.net/g3/

2. As your very first task, use SQL to calculate the number of seconds in one day (where day is 24 hours, hour is 60 minutes and a minute is 60 seconds).
3. Let's see if you can build a query now. Get x, y and z information from all the rows from the table 'stars'.
4. Okay. Time for you to try something out. Make a query which returns starid, x, y and z for all stars where x is greater than zero and starid is less than one hundred. Sort the results by the y-coordinate so that the smallest values come first.
5. Build a query where you calculate the sum of all y values divided by all x values.
6. Hilight five stars which have star id between 5000 and 15000, and have class 7. (Hint: don't try to do this with a single query at this point).
7. Hilight all the stars with starid between 10000 and 11000. (I know, this is not too difficult, but it looks neat).
8. Kill off all stars with starid lower than 10000. Do this inside a transaction, so that when I run the ROLLBACK command, we're back with the original galaxy.
9. Starting from the normal galaxy, update it so that you swap the x and z coordinates for stars which have star id between 10000 and 15000.
10. Hilight all stars with starid of at least 20000, which have planets with moons that have an orbit distance of at least 3000. Remember to remove any old hilights before starting.
11. Hilight the star (or stars) which has the planet with the highest orbit distance in the galaxy. Remember to clear the old hilights before beginning.
12. Generate a list of stars with star ids between 500 and 600 (but not including 500 and 600) with columns "starname", "startemp", "planetname", and "planettemp". The list should have all stars, with the unknown data filled out with NULL. These values are, as usual, fictional. Calculate the temperature for a star with ((class+7)*intensity)*1000000, and a planet's temperature is calculated from the star's temperature minus 50 times orbit distance.
13. Create a VIEW called "numbers" with the columns "three", "intensity" and "x", where "x" and "intensity" come from the stars table, "three" contains the number 3 on all rows. For additional fun, sort the whole thing by "x" - although I won't care.
14. Create a table named 'colors' with the columns 'color' and 'description'. Color is integer, description is text. Populate the table with color values from -3 to 10; each star class has its own color; fill the description with something (I won't care exactly what). Basic idea is that it will be possible to make a join between stars and colors where stars' class is compared to colors' color number.
15. Create a table called "quotes" with two columns: "id", which is primary key, and takes integers, and "quote" which contains non-null text strings, such as quote of the day (http://www.qotd.org/). Fill in a couple of rows so that I have something to query for.
16. First, create and populate a table using this command*. Rename the table to 'my_table', and add a column called 'moredata'. Add one whole new row and change the 'moredata' value of at least one existing row. (Yes, I'm aware you could do all that by changing the creation commands, but that is not the point of this exercise).
    *

```
CREATE TABLE alter_test (id INTEGER PRIMARY KEY, data TEXT NOT NULL);
INSERT INTO alter_test (data) VALUES ('Foo');
INSERT INTO alter_test (data) VALUES ('Bar');
INSERT INTO alter_test (data) VALUES ('Baz');
```

17. Hilight the star with the most orbitals (combined planets and moons). If multiple stars have the highest number of orbitals, highlight the one with the lowest star id.
18. Build a query which returns starids from planets.
    The starids should be selected so that for each starid (x) in the list:
    - there should exist a planet with a starid that's three times x
    but
    - there should not exist a planet with starid two times x.
    Only use starids from the planets table.
19. Create a trigger which, when a new star is created, clears the hilight table and inserts the new star id to the hilight table.
20. Use ALTER TABLE to rename the 'gateway' table to 'gateways'. (ALTER TABLE was covered in chapter 16).