INF1343, Winter 2012, Quiz 1

_____

write your name above

"Luke's Droid Repair" is a small droid repair shop based on Tatooine. Luke uses a database to keep track of the different information related to his shop's operations. One of the tables in the database, called "customer," stores information about the shop's customers. Here is the SQL statement that was used to create this table:

```
create table customer (
 customer_id integer,
 name varchar(200),
 species varchar(100), -- e.g., "Hutt," "Human" or "Kitonak"
 telephone integer(20),
 email varchar(100),
 last_order_date date, -- the last time the customer made an order
 last_order_amount decimal(10,2) -- the amount of last order in Tatooine dollars
);
```

a) Luke wants to do a marketing campaign targeting Hutts, one of Tatooine's species. This requires getting names, telephone numbers, and email addresses of all customers in his database who are Hutts and who have not made an order in 2012 or 2011. Luke wants this list sorted by the amount of each customer's last order, from the largest to the smallest. Write the necessary query below.

The answer:

```
select name, telephone, email
from customer
where
  species="Hutt"
  and last_order_date < "2011-01-01"
order by last_order_amount desc;
```

Notes:

- For both questions, queries containing syntactic mistakes were penalized heavier than those containing just semantic ones. (Having syntactic mistakes means one would not need to know the question to conclude that your query is wrong.) Syntactic mistakes are a sign that you are not spending enough time with the database. In general, for an A-range grade **both** answers had to be syntactically correct.
- For question one, multiple use of "where" was one of the more common syntactic mistakes. "Where" should only be used once (unless you are using nested queries, which we will cover later). So: "where X and Y", not "where X and where Y."
- The answer shown above assumes that we are still in 2012 (and that therefore we are not going to have any orders from 2013, 2014, etc). If we wanted to avoid this assumption we could use a somewhat more complicated condition:

  ```
  … and (last_order_data < "2011-01-01" or last_order_date > "2012-12-31")
  ```

- MySQL's date format is "YYYY-MM-DD." "2011-01-01" is not the same as "01-01-2011". If we wanted to avoid an assumption about the data format, we could use MySQL's "year()" function:

  ```
  … and year(last_order_data) != "2011" and year(last_order_date) != "2012"
  ```

- Comparing just with the year (e.g., "2011") could work, but you need to be careful. You *can* use "< "2011"" to catch dates before 2011, but you *cannot* use ">"2012"" to catch dates after 2012, because all 2012 dates satisfy this condition too! (If you do a dictionary sort, "2011-01-01" comes after "2012".)

- You cannot do a comparison of last_order_date using something like “> 2011-01-01”, because 2011-01-01 is 2009! (2011 minus 1, minus 1.)
- You cannot do a three way comparison, such as "2011-01-01" <= last_order_date < "2012-01-01". What this expression would do is evaluate the first part ("2011-01-01" <= last_order_date) and then compare the result (True or False) with "2012-01-01". The result is always going to be False.
- There is really no good reason to use “like” when working with dates.

b) Luke noticed that some of the records in the customer table do not specify customers’ species. This can be a problem on Tatooine. Luke wants to know how many records fail to specify species. Write the query below.

The answer:

```
select count(*)
from customer
where species is null;
```

Notes:

- You need to use “is null” here. Neither “=NULL”, nor “="NULL"” will give you the right result and both were considered syntactic mistakes. We covered this in class.
- The question needs to be answered by using “count”. Not using count is a major omission.
- You need to use count(*) here. If you use count(species), your query will always return 0. The reason is that count(x) counts the number of rows where the value of x is **not** null. So, if you specify “where species is null” and then do count(species), you are asking the database to first select those rows where the value of species is null and then to count how many of those rows have something other than null in the species field. That’s *always* 0. Some of you used count(customer_id) instead. This will give you a count of only those customers whose customer_id is set. This will usually give you the right answer in practice, however it imposes an additional condition that is not asked for in the question.