CSCI 4320
Spring 2023
Exam 2

Last Name: _____

First Name: _____

Write your name on this paper and write your answers on the answer sheets provided. You may refer to any SQL or Java code that you have created.

1. Suppose a database exists to store information about albums and artists and that it has been designed as follows:

   - artist=(artist_id, name, hometown)
   - genre=(genre_id, description)
   - album=(album_id, artist_id, title, year, genre_id)

   Show the SQL statements necessary to create these tables so that the following integrity rules will be enforced.

   (a) (3 pts) the artist's name is required, but the hometown is optional; genre description is required; title is required but year is not

   (b) (4 pts) every album must have an valid artist specified; if an artist is removed from the list then any albums associated with them should be removed as well

   (c) (2 pts) include an SQL statement (separate from the `CREATE TABLE` command) that will create an entry in the `genre` table with an id number of 1 and a description of "Unknown"

   (d) (3 pts) if a album's genre is not known, the genre id number should default to the value 1

   (e) (4 pts) if a genre is deleted from the list, any albums so assigned should revert to "Unknown"; if a genre's id number changes then then entries in the `album` table should be changed to match automatically

   (f) (4 pts) when a new album is being entered make sure that the year is not in the future. HINT: The expression `date_part('year',current_date)` will return the current year (in four digit format). If the year is from the future then deny entry.

   (g) (3 pts) specify a primary key for each table

2. (4 pts) What is the difference between a shared lock and an exclusive lock?

3. (3 pts) In the context of database systems, what is a transaction?

4. (6 pts) Draw the transaction state diagram presented in class.

5. (4 pts) Describe two characteristics we want transactions in a DBMS to exhibit. HINT: The acronym ACID represents four such characteristics.

6. Suppose the following tables represent skunk sightings by year for various locations. Yes ... that's right ... skunk sightings!

location

id	city	state
23	Abilene	TX
24	Anson	TX
25	Baird	TX
26	Clyde	TX
27	Buffalo Gap	TX
49	Hobbs	NM

sightings

id	year	amount
23	2016	26
23	2017	43
23	2018	33
24	2016	31
24	2017	null
24	2018	29
25	2017	null
25	2018	null

(a) (2 pts each) What would be displayed by each of the following queries:

i. `SELECT SUM(amount) FROM sightings WHERE id=24`

ii. `SELECT COUNT(amount) FROM sightings WHERE id=24`

(b) (3 pts) Write a SQL query to display city name and state name for cities whose average number of skunk sightings is fewer than 30 per year.

(c) (4 pts) Write a SQL statement to create a view named `missing_data` that will list the city and year for all cities that have no valid sighting data.

(d) (2 pts) What PostgreSQL command would be used to establish an "access exclusive" lock on a the `sightings` table.

(e) (2 pts) How would the lock in the previous question be released?

(f) (3 pts) Write a command that will completely remove the `location` table.

(g) (6 pts) Write a sequence of JDBC statements that, if executed, *would* be susceptible to an SQL injection attack. You can assume a connection to the database already exists and is stored in a variable named `con`. Declare any additional variables you need to construct this scenario.

(h) Suppose we have teams of researchers assigned to record skunk sightings they have observed in the month. Each city has up to 10 researchers who are independently recording their numbers each month. For a researcher to enter their updated values they need to read the current number of sightings for a location and year, then add the number of new sightings from the past month, then update the sightings to take on the new total value. The code to perform these actions in JDBC might start something like this:

```
pstmt= con.prepareStatement("SELECT amount FROM sightings WHERE id=? AND year=?");
pstmt.setInt(1, location_id);
pstmt.setInt(2, current_year);
result= pstmt.executeQuery();
if (result.next()) {
   amount= result.getInt(1);
}
```

```
else {
   amount= 0;
}
amount= amount + num_sighted_this_month;
```

i. (4 pts) Add JDBC commands to follow this sequence of steps that could be used to update the sightings table to hold the newly calculated number of sightings.

ii. (4 pts) In the scenario described above is there any concern that two concurrent updates to the sightings table might somehow conflict and result in an incorrect value? Explain your answer.

iii. (10 pts) Suppose for a moment that (if needed) the conditions were created where the sequence of code above (combined with your update commands) would in fact have the possibility of producing incoherent results in the face of concurrent transaction. Rewrite the block above so that it contains additional statements needed to give correct results regardless of whether the transactions were concurrent or not. Your protecting code should encompass the smallest section of code possible and still be effective.