Name: _____

Please answer the following questions within the space provided on the following pages. Should you need more space, you can use scratch paper, but clearly label on the scratch paper what problem it corresponds to. While you are not required to explain your queries, comments may help me to understand what you were trying to do and thus increase the likelihood of partial credit should something go wrong. If you get entirely stuck somewhere, explain in words as much as possible what you would try.

This is a pen and paper exam, and thus computers and internet capable devices are prohibited. If you have any confusion about question intention or wording, please do not hesitate to ask!

Your work must be your own on this exam, and under no conditions should you discuss the exam or ask questions to anyone but myself. Failure to abide by these rules will be considered a breach of Willamette's Honor Code and will result in penalties as set forth by Willamette's academic honesty policy.

Please sign and date the below lines to indicate that you have read and understand these instructions and agree to abide by them. *Failure to abide by the rules will result in a 0 on the test.* Good luck!!

_____ _____

Signature Date

Question:	1	2	3	4	5	6	Total
Points:	6	8	8	12	9	0	43
Score:							

1. There exists a table called `correspondents` in your database which has contains the following information:

Column Name	Data Type	Description
`id`	`SERIAL`	Unique numeric identifier
`name`	`TEXT`	Name of correspondent
`age`	`SMALLINT`	Age of correspondent
`gender`	`VARCHAR(15)`	Male, female, or nonbinary
`met_thru`	`VARCHAR(20)`	Where the correspondent was originally met
`last_talk`	`DATE`	Date of last talk or correspondence
`notes`	`TEXT`	Notes from last correspondence

Given the queries below, choose the option that best summarizes the question the query is attempting to answer.

(2) (a)
```
SELECT name
FROM correspondents
WHERE gender = 'Female' and last_talk < '2018';
```

A. What female was most recently corresponded with since 2018?
B. What females were talked to within the last 2018 days?
C. What is the age of all the females that were corresponded with since 2018?
D. What females have not been talked to since at least 2018?

(2) (b)
```
SELECT avg(age)
FROM correspondents
WHERE notes ILIKE '%had baby%'
   OR notes ILIKE '%pregnant%';
```

A. How many correspondents have children?
B. How old are correspondent's children on average?
C. What is the average age of all correspondents who most recently were pregnant?
D. What females were recently pregnant?

(2) (c)
```
SELECT DISTINCT met_thru FROM correspondents
ORDER BY last_talk DESC LIMIT 5;
```

A. How many unique locations have correspondents been met at?
B. What locations were the 5 most recent correspondents originally met at?
C. What 5 unique locations were the most recent correspondents originally met at?
D. What unique locations were the 5 most recent correspondents originally met at?

(8) 2. A mysterious table (named `mysterious_table`) has the following query run on it:

```
SELECT
  min(red - cyan::INT) AS new_a,
  percentile_disc(0.5) WITHIN GROUP (ORDER BY cyan) AS new_b,
  max(2 * red + green) AS new_c
FROM mysterious_table
WHERE blue ILIKE '%odd'
  AND orange BETWEEN '1:00' AND '13:00';
```

and returns a table with the following form:

Column Name	Data Type
new_a	INTEGER
new_b	REAL
new_c	NUMERIC

Determine as much information as you can about the columns comprising `mysterious_table`, and explain how you arrived at your conclusions.

Solution: Since `new_a` is an integer, and `min` returns a value straight from the column, it requires that `red` and `cyan::INT` both be integers. So we know `red` must be an integer. `percentile_disc` also pulls directly from a column, and so the fact that `new_b` ends up a `REAL` value means that `cyan` must also be a `REAL` value. And this makes sense with why it had to be converted to an integer for `new_a`. Given that `new_c` is a `NUMERIC` type and we already know that `red` is an integer, this implies that `green` must also be a `NUMERIC` type, since otherwise `new_c` would be either an `INT` or `REAL`. We are filtering on `blue` with pattern matching, which really only makes sense for some sort of string, either `TEXT` or `VARCHAR`. And then `orange` looks to be being compared to two different times, which most likely would make it a `TIME` type. It would potentially also be `TEXT`, but it would be very clunky. So in the end we have:

red	INT
cyan	REAL
green	NUMERIC
blue	TEXT
orange	TIME

3. You have the below CSV file of names and corresponding addresses.

```
William Ellison,57 Elizabeth Dr.,Merrillville,IN,46410
Colten Spears,8457 Sycamore Ave.,Amsterdam,NY,12010
Kendra Aguilar,76 North Alton Lane,Tualatin,OR,97062
Natalia Church,7789 Ryan Dr.,Englewood,NJ,07631
```

(3) (a) Write out a command to create a table that will hold this information, including appropriate data types. In addition to the fields in the CSV, your table should include an `id` column with the `serial` data type to uniquely identify the individuals.

Solution:

```
CREATE TABLE addresses (
  id SERIAL,
  name TEXT,
  address TEXT,
  city TEXT,
  state CHAR(2),
  zip CHAR(5)
);
```

(3) (b) Write out a command to import the data from the CSV file into your above created table. You can assume the CSV file is located at `/data/addresses.csv`.

Solution:

```
COPY addresses (name, address, city, state, zip)
FROM '/data/addresses.csv'
WITH (FORMAT CSV);
```

(2) (c) After importing the data, you realize that your table is still missing the information for Peter Hood, who lives at 73 East Wrangler Street, New Kensington, PA 15068. Write a command to add this information to the end of your table.

Solution:

```
INSERT INTO addresses (name, address, city, state, zip)
VALUES ('Peter Hood', '73 East Wrangler Street',
        'New Kensington', 'PA', '15068');
```

4. You have a table named `amazing` in your database that looks like below.

c1 *DATE*	c2 *TEXT*	c3 *INT*	c4 *DOUBLE PRECISION*
2022-06-29	Curling	24	9.5
2022-11-15	Tennis	NULL	5
2022-06-12	Baseball	2	-0.5
2022-12-27	Ultimate	0	2.1
2022-04-15	Surfing	NULL	4.5
2022-10-03	Cheerleading	10	-1

Use it to determine the output of the below queries, including column names and type.

(3) (a)
```
SELECT c2, c3 + c4 AS "add"
FROM amazing
WHERE c2 ILIKE '%e%i%'
ORDER BY c2;
```

Solution:

c2 *TEXT*	**add** *DOUBLE PRECISION*
Cheerleading	9
Tennis	NULL

(3) (b)
```
SELECT COUNT(*) % COUNT(c3) AS rem
FROM amazing
WHERE c4 > 0;
```

Solution:

rem *INT*
0

(3) (c)

```
SELECT c2
FROM amazing
WHERE c1::TEXT ILIKE '%-0_-%'
ORDER BY c4;
```

Solution:

c2 *TEXT*
Baseball
Surfing
Curling

(3) (d)

```
SELECT sum(c4)
FROM amazing
WHERE c3 IS NULL OR c4 < 0;
```

Solution:

sum *DOUBLE PRECISION*
8

5. Wordle has taken the world by storm, so suppose you had a table in your database keeping track of various player's performance. The table (named `wordle`) has the following columns:

Name	Type	Description
`puzzle_id`	`INT`	The unique puzzle id number
`release_date`	`DATE`	The day the puzzle was publically available
`solution`	`CHAR(5)`	The 5 letter solution for that puzzle
`player_name`	`TEXT`	The name of the player
`guesses`	`SMALLINT`	The number of guesses until solving the puzzle. `Null` if they failed to solve the puzzle in less than 7 guesses.

Each time any player attempts the daily puzzle, another row is added to the table. So, for example, a **few rows** of the table might look something like:

`puzzle_id`	`release_date`	`solution`	`player_name`	`guesses`
⋮				
222	2022-01-27	mount	Frank	5
222	2022-01-27	mount	Joe	4
223	2022-01-28	perky	Frank	`NULL`
224	2022-01-29	could	Jill	6
⋮				

Using this table, construct queries that would answer the following questions.

(3) (a) When the player named "Bobby" attempts a puzzle, he succeeds in solving it what percentage of the time?

Solution:

```
SELECT COUNT(guesses) / COUNT(*)::REAL * 100
FROM wordle
WHERE player_name = 'Bobby';
```

(3) (b) What is the most likely number of guesses it takes any player to complete the puzzle if the letter "a" is the second letter?

Solution:

```
SELECT mode() WITHIN GROUP (ORDER BY guesses)
FROM wordle
WHERE solution ILIKE '_a%';
```

(3) (c) How many of the daily puzzles have been solved in one guess? (Note that this is different from asking the number of players that have solved the puzzle in one guess.)

Solution:

```
SELECT COUNT(DISTINCT puzzle_id)
FROM wordle
WHERE guesses = 1;
```

(2 (bonus)) 6. What strengths or benefits do relational databases have in comparison to a standard spreadsheet method of storing information?

Solution: Probably the biggest are that relationships can be maintained between data without needing to copy or duplicate the data in various tables, as well as enforcing a schema to help keep data standardized and clean.