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! You got this!

_____ _____

Signature Date

Question:	1	2	3	4	5	6	Total
Points:	12	10	12	18	18	0	70
Score:							

1. Below you are given the queries that created three tables and a view of what is currently in those tables.

tab1

```
CREATE TABLE tab1 (
  A TEXT,
  B INT CHECK (B > 0),
  C NUMERIC(4,2) UNIQUE,
  PRIMARY KEY (A, B)
);
```

A	B	C
Henry	2	4.23
Jake	1	10.52
Katy	3	-4.83
Wyatt	4	83.10

tab2

```
CREATE TABLE tab2 (
  D DATE PRIMARY KEY,
  E NUMERIC(4,2)
    REFERENCES tab1 (C)
);
```

D	E
2022-02-14	10.52
2022-01-30	83.10
2022-04-24	4.23

tab3

```
CREATE TABLE tab3 (
  F INT PRIMARY KEY,
  G REAL,
  H TEXT,
  I DATE
    REFERENCES tab2 (D)
  CHECK (F > G)
);
```

F	G	H	I
2	1.1	Jake	2022-01-30
8	6.445	Katy	2022-04-24
0	-0.24	Henry	2022-01-30

Using this information, for each of the following queries, determine whether that query would run successfully or not. If it does not, explain directly what error would occur / why it occured.

(2) (a)
```
INSERT INTO tab3 VALUES
(7, -4, 'Katy', '2022-02-14');
```

(2) (b)
```
ALTER TABLE tab3 ADD FOREIGN KEY (H) REFERENCES tab1 (A);
```

(2) (c)
```
UPDATE tab2
SET E = -4.83;
```

(2) (d)
```
DELETE FROM tab1
WHERE A = 'Katy';
```

(2) (e)
```
ALTER TABLE tab3 ALTER COLUMN G SET DATA TYPE INT;
```

(2) (f)
```
ALTER TABLE tab1 ADD PRIMARY KEY (C);
```

2. Suppose you have the simple table below (named `regexes`) which contains only a single column of text values:

value
text
'APR 24 2022 12:23:42 -- 3321-2635-2673-3481 -- Hazel Western paid $50.23'
'MAR 18 2022 04:12:56 -- 8352-1634-8492-1642 -- Bob Johanssen paid $100.56'
'JAN 31 2022 15:02:14 -- 0234-1004-2835-1666 -- Karen Hinter paid $19.20'

For each of the below regular expression matching queries, determine what the output table would look like.

(2) (a)
```
SELECT (regexp_match(value, '-(\d{4})'))[1] FROM regexes;
```

A.
3481
1642
1666

B.
2022
2022
2022

C.
3321
8352
0234

D.
2635
1634
1004

(2) (b)
```
SELECT (regexp_match(value, '.*(\d)\.'))[1] FROM regexes;
```

A.
3
6
0

B.
0
0
9

C.
2
1
3

D.
3
8
0

(2) (c)
```
SELECT (regexp_match(value, '[A-Z][a-z]*'))[1] FROM regexes;
```

A.
A
M
J

B.
H
B
K

C.
APR
MAR
JAN

D.
Hazel
Bob
Karen

(2) (d)
```
SELECT (regexp_match(value, '[A-Z][a-z]+'))[1] FROM regexes;
```

A.
A
M
J

B.
H
B
K

C.
APR
MAR
JAN

D.
Hazel
Bob
Karen

(2) (e)
```
SELECT (regexp_match(value, '\d{3}[5-9]'))[1] FROM regexes;
```

A.
2635
NULL
2835

B.
NULL
8352
1666

C.
2022
2022
2022

D.
3321
8352
0234

(12) 3. Suppose you have a table named `enigma`, for which you know nothing of its contents. However, when you run the three queries below, you get the shown output. Determine a possible schema *and contents* for `enigma` that would satisfy all the below queries. **There is not a unique answer here. But your potential table needs to be consistent with the queries and outputs.** Make sure to indicate your column names and types as well as the actual row contents.

Query 1

```
SELECT
  COUNT(*),
  SUM(c1)
FROM enigma;
```

Output 1

count	sum
5	25

Query 2

```
SELECT c1
FROM enigma
WHERE to_tsvector(c2) @@
  to_tsquery(
    '(raccoon | swirl) & !clean'
  );
```

Output 2

c1
4
5
7

Query 3

```
SELECT (
  SELECT COUNT(*)
  FROM REGEXP_SPLIT_TO_TABLE(c2, ' ')
  )
FROM enigma
WHERE c1 = (
  SELECT COUNT(*)
  FROM REGEXP_SPLIT_TO_TABLE(c2, ' ')
  );
```

Output 3

count
4
5
7
3
6

I've included a full blank page on the next page you can use for scratch-work if you need. Just make sure it is clear to me where your final table is written down.

4. Many GPS tracking units are capable of outputting tabulated data, which frequently involves a simple timestamp alongside the determined latitude and longitude measurements. Suppose you have such a table, which also includes a serial column to uniquely determine the ordering of the points, as seen below:

gps_tracks	
id	*serial*
time	*timestamp*
latitude	*double precision*
longitude	*double precision*

Here you can safely assume that the serial `id` increases in steps of 1 with no gaps (which isn't always guaranteed for serial columns). Use this information to write queries to answer the following questions. You can assume that any necessary extensions have already been added to the database. All of these could be done as a single query, but you can use multiple queries if you prefer (though nothing that would require a user to read a value off one table and enter it into a different query).

(6) (a) What is the straight-line distance between the initial point in the table and the final point? Note that this is not the actual distance traveled, but just the distance directly between the first and last point.

(6) (b) What is the total actual distance traveled moving from point to point along the sequence of coordinates?

(6) (c) Between which two `ids` was the speed the greatest? As a reminder, speed is a measure of distance over elapsed time.

5. You have the following table of information about western states in the continental US.

name *text*	**neighbor_count** *int*	**vote_color** *text*	**year_admitted** *int*	**num_reps** *int*	**avg_elev_ft** *int*
Oregon	4	blue	1859	5	3300
Washington	2	blue	1889	10	1700
Idaho	6	red	1890	2	5000
California	3	blue	1850	53	2900
Nevada	5	blue	1864	4	5500
Montana	4	red	1889	1	3400
Wyoming	6	red	1890	1	6700
Utah	6	red	1896	4	6100
Colorado	6	blue	1876	7	6800
Arizona	4	red	1912	9	4100
New Mexico	4	blue	1912	3	5701

Using this table, determine the output of the following queries.

(6) (a)
```
SELECT
  SUM(neighbor_count) AS result
FROM states
WHERE avg_elev_ft > 3000
GROUP BY year_admitted
HAVING COUNT(*) > 1
ORDER BY year_admitted
;
```

(6) (b)

```
WITH
heights (name, tier) AS (
  SELECT
  name,
  CASE
    WHEN avg_elev_ft < 3000 THEN 'Low'
    WHEN avg_elev_ft < 5000 THEN 'Mid'
    ELSE 'High'
  END
  FROM states
)

SELECT
  h.tier,
  vote_color,
  COUNT(*)
FROM states AS s
JOIN heights AS h
  ON s.name = h.name
GROUP BY h.tier, vote_color
HAVING SUM(num_reps) > 5
ORDER BY h.tier, vote_color
;
```

(6) (c)

```
SELECT name
FROM states as s1
WHERE EXISTS (
  SELECT 1
  FROM states as s2
  WHERE s1.neighbor_count < s2.neighbor_count
  ) AND name !~ '^[N-Z]'
ORDER BY name
;
```

(4 (bonus)) 6. Compare and contrast a relational database with a classic spreadsheet like Excel. Where does each excel in your opinion? What are the weaknesses of each?

Thanks for the great semester, and sticking with me as I taught this class for the first time! Hopefully you got a lot out of it, and it proves useful in the future awesome things I know you'll go on to do! Never hesitate to drop me a line if you think there is something I could help you with going forward in the future. And enjoy your summer!