Topic Overview

Here is a breakdown of the general topics you should be prepared for on the midterm:

Theory/Supporting Technology

- ☐ The Data Engineering Lifecycle
  - ☐ What does the data engineering lifecyle look like?
  - ☐ Can you decribe a bit what is happening with each piece of the data engineering lifecycle?
  - ☐ What does it mean for a system to be reliable?
  - ☐ What does it mean for a system to be scalable?
  - ☐ What does it mean for a system to be maintainable?
- ☐ Data Generation and Storage Models
  - ☐ Discuss several forms of data generation and compare and contrast their strengths/weaknesses/use-cases.
  - ☐ Strengths and weaknesses of relational database systems
  - ☐ Strengths and weaknesses of NoSQL document databases?
  - ☐ Strengths and weaknesses of NoSQL graph databases?
- ☐ Shells and Remotes
  - ☐ What are the differences between a terminal emulator and a shell?
  - ☐ Navigating a file structure with a BASH shell using either absolute or relative paths
  - ☐ Basic fundamental shell utility programs: `ls`, `cat`, `wc`
  - ☐ Copying and removing files in a BASH shell
  - ☐ How can you get help for a given BASH program?
  - ☐ Redirecting standard input and output
  - ☐ Piping from command to command
  - ☐ **I will not ask for memorization of any particular flags.**
  - ☐ Accessing a remote system through SSH given the necessary information
  - ☐ Understanding basic `.ssh/config` options and profiles
  - ☐ Copying/moving files between systems

SQL

- ☐ Construction
  - ☐ Creating new tables from scratch
  - ☐ Choosing proper data types
  - ☐ Appending rows to a table
  - ☐ Importing and Exporting from/to CSV
  - ☐ Creating from a `SELECT` query
- ☐ Making selections
  - ☐ Choosing unique entries

    - ☐ Choosing desired columns
    - ☐ Filtering properly using `WHERE` and boolean operators
    - ☐ Sorting
    - ☐ Limiting output to a number of rows

- ☐ Calculations
    - ☐ Data type of outputs
    - ☐ Column operations
        - Basic arithmetic operations
        - Applying common, built-in functions or operations
    - ☐ Aggregates
        - Straightforward aggregates like `AVG()`, `SUM()`, `COUNT()`, etc
        - Order dependent aggregates like `PERCENTILE_CONT()` and `MODE()`
- ☐ Joins
    - ☐ Inner joins
    - ☐ Left/Right joins
    - ☐ Full outer joins
    - ☐ Cross joins
    - ☐ Self joins

Question Types

Questions will fall into several main divisions, of which I will include examples of each later in the study guide.

Theory: Largely, these will come either in the form of short answer, matching, or multiple choice type questions.

Qualitative: In general, these wouldn't deal with direct values in a table, but are more conceptual in understanding what a particular piece of SQL is doing.

- Given a general table and query, describe what the output would look like, or what properties it might have.
- Given a desired output, what properties might the query or initial table have needed to possess?
- Given a table and desired output, what would the query need to look like?

Quantitative: These will deal more directly with sample data in a table.

- For this particular query with this tabular data, what would the output be? (These will naturally be with small and simple tables, as you won't have a computer to aid you.)

Example Questions

1. You have a particular table in your database called `inventory` that follows the below schema and has at least one row of data.

Column Name	Data Type
`id`	`SERIAL`
`name`	`VARCHAR(20)`
`weight`	`REAL`
`price`	`NUMERIC(5,2)`
`stock`	`INT`

You then run the following query:

```
SELECT COUNT(weight) / COUNT(*) * 100::REAL
FROM inventory;
```

(a) How many columns are returned in the output?

A. 0
B. 1
C. 5
D. Impossible to tell

(b) How many rows are returned in the output?

A. 0
B. 1
C. The same as the number of rows in the `id` column
D. Impossible to tell

(c) For each column that is returned, what would be its corresponding data type?

Solution: Since there will only be a single column returned, we just need to determine a single data type. `COUNT(weight)` will be an integer, as will `COUNT(*)`. Dividing them then will also result in an integer (which is probably *not* what was desired here, but is what the query will do). Then the 100 has been cast to be a floating-point value, and multiplying anything by a floating point value results in a floating point value. So the final value would be a `REAL` type value (and, most likely, equal to 0).

(d) In a sentence or two, describe what this query is doing (or trying to do). I'm looking less for a line by line description of what is happening, and more an overall description of what the query is trying to achieve.

Solution: The query is computing the percentage of items that have a recorded weight.

2. Without any information about the table called `mystery`, you run the below query:

```sql
SELECT
  dim1 * dim2 * dim3 AS volume,
  |/(score::DECIMAL + 10) AS metric
FROM mystery
WHERE best_by + '3 days 10 minutes' < sold
ORDER BY score::DECIMAL
```

where any type conversions were **necessary** (not optional). The resulting table has the form:

Column Name	Data Type
`volume`	`NUMERIC`
`metric`	`DOUBLE PRECISION`

Write as *much detail as you can* about what you know about the table `mystery` from just this query and its results.

Solution: The table must have at least 6 columns, as determined by the number of different column identifiers mentioned. I can further narrow down what data types they might be according to how they are used and the resulting outputs:

- At least one of `dim1`, `dim2` or `dim3` must be a `NUMERIC` type. The others could also be `NUMERIC` or they could be `INTEGER`, but they can not be any sort of floating-point type, since the output in the end is `NUMERIC`.
- `score` is most likely a number in the form of a `TEXT` field (something like `'15'`). That seems to be the best explanation for why it would need to be converted to fixed-point in the calculation (you can't add text and numbers) and why it was converted in the ordering (text and numbers order differently). If it was already a number-type, then there would be no need to convert it for ordering.
- `best_by` needs to be some sort of `TIMESTAMP` or `DATE` type, since it is being added to an interval. Arguably I guess it could be `TEXT` that just gets added to the interval string, but the contents of the string would suggest a interval.
- `sold` would also need to be a sort of `TIMESTAMP` or `DATE` type so that it could be easily compared to the sum of `best_by` and the interval.

In general, `metric` being a `DOUBLE PRECISION` type at the end doesn't really tell us anything, since square roots always return a floating point type.

3. Suppose I wanted to import the below CSV file (saved at `C:\Data\important.csv`) into a Postgresql database. Write out the necessary commands to create the table and import the data.

```
id,name,p1,p2,p3,total,submitted
1,Bill,7,8,2,17,2022-01-25 18:00
2,Nancy,7,7,7,21,2022-01-26 15:15
3,Jacob,5,10,5.6,20.6,2022-01-25 23:47
4,Sebastian,9.5,10,10,29.5,2022-01-29 19:34
```

Solution: I would probably write something like this:

```
CREATE TABLE important (
  "id" BIGSERIAL,
  "name" TEXT,
  p1 NUMERIC(4,2),
  p2 NUMERIC(4,2), -- see note below
  p3 NUMERIC(4,2),
  total NUMERIC(5,2),
  submitted TIMESTAMP
);

COPY important
FROM 'C:\Data\important.csv'
WITH (FORMAT CSV, HEADER);
```

All the current values of p2 are integers, but given the similarity in name to the others that *do* have fractional values, it doesn't seem inconceivable that this column could occasionally get fractional values as well.

4. You have a table named `special` in your database, that looks as can be seen below:

id *SERIAL*	**name** *TEXT*	**cola** *INT*	**colb** *NUMERIC(4,2)*	**colc** *INT*
1	Angel	3	4.50	9
2	Bob	2	2.00	5
3	Charlie	NULL	4.10	4
4	Angel	5	12.40	10
5	Charlie	8	NULL	7

(a) What would be the output of the below query?

```
SELECT
  name,
  colb / (colc / cola) AS o1,
  2 * colc + colb AS o2
FROM special
WHERE colb IS NOT NULL
ORDER BY o1
```

Solution: Initially, I'd just look at the table before ordering. The filter that `colb` is not `NULL` means we are only looking at the first 4 rows. So those would look like:

name *TEXT*	**o1** *NUMERIC*	**o2** *NUMERIC*
Angel	1.50	22.50
Bob	1.00	12.00
Charlie	NULL	12.10
Angel	6.20	32.40

So then ordering, with `NULL`s at the start, would give us:

name *TEXT*	**o1** *NUMERIC*	**o2** *NUMERIC*
Charlie	NULL	12.10
Bob	1.00	12.00
Angel	1.50	22.50
Angel	6.20	32.40

(b) What would be the output of the below query?

```
SELECT
  min(colc - cola) AS mind,
  percentile_disc(0.5) WITHIN GROUP (ORDER BY name) AS midname,
  sum(colb + colc) AS summy
FROM special
WHERE id % 2 > 0;
```

Solution: It is clear that we have aggregate functions here, so I'll figure out what the table would look like *before* those were applied, and then compute them accordingly. So just applying the filter, we will just keep the odd id rows:

id *SERIAL*	**name** *TEXT*	**cola** *INT*	**colb** *NUMERIC(4,2)*	**colc** *INT*
1	Angel	3	4.50	9
3	Bob	`NULL`	4.10	4
5	Charlie	8	`NULL`	7

Then `colc - cola` will only be non-null in the first and 3rd rows, of which -1 would be the smallest. The median name is clearly Bob, as it would be the middle one when ordered alphabetically. And then, again, `colb + colc` is only non-null for the first two rows, which if we add them we have $(4.5 + 9) + (4.10 + 4) = 21.6$. So our final table would be:

mind *INT*	**midname** *TEXT*	**summy** *NUMERIC*
-1	Bob	21.6

5. You have the table (named `teachers`) of teachers in your local area with a schema given below, where I have also added a quick description of each column.

Column Name	Data Type	Description
`id`	`SERIAL`	Unique identifying integer
`name`	`TEXT`	Full name of the teacher
`sex`	`CHAR(1)`	Sex of teacher: M or F
`grade`	`INT`	Grade level taught. Kindergarden is 0.
`yr_exp`	`INT`	Years of teaching experience
`salary`	`NUMERIC(8,2)`	Yearly salary in US dollars
`peak_deg`	`VARCHAR(3)`	Peak degree obtained: HS,BS/BA,MS,PhD

Write out queries to answer the following questions.

(a) What is the average salary of high school (grades 9-12) teachers with graduate degrees?

Solution:

```sql
SELECT
  avg(salary)
FROM teachers
WHERE grade BETWEEN 9 AND 12 AND peak_deg IN ('MS','PhD');
```

There are definitely multiple ways you could do the filtering on this, but this is probably the most concise I'm currently seeing.

(b) What Ms. or Mrs. Johnson has been teaching for the longest?

Solution:

```sql
SELECT
    name
FROM teachers
WHERE name ILIKE '%Johnson%' AND sex = 'F'
ORDER BY yr_exp DESC
LIMIT 1;
```

If I was really worried about some female with a first name of "Johnson", I could probably remove the trailing wildcard character from my pattern match, thereby forcing the "Johnson" to be at the end of the full name.

6. All the following options best describe ways that the "maintainability" of a system could be improved except for one. Which is the odd one out?
   - A. Providing transparency into the runtime behavior and internals of the system.
   - B. Providing good documentation
   - **C. Providing load balancing across multiple systems**
   - D. Avoiding any dependency on a single machine, which then can not be taken easily offline
   - E. Providing good default behavior, but giving freedom to override defaults if needed

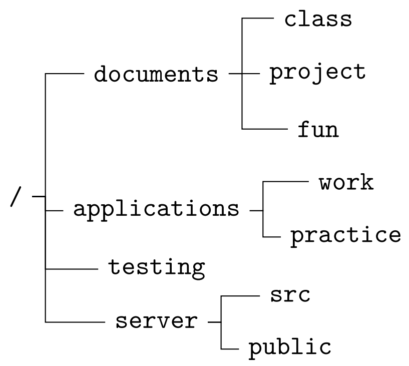
7. Match the below terms to the description that best matches. Each term will only connect to a single description.

E	Acts on data in near-realtime, as it arrives
D	Excels at modeling many-to-many relationships between data
F	Interface for programatically getting or setting information from a server
B	Focused around joins and data normalization
C	Stores only the difference between the previous version of the data and the new version
A	Data is stored together with related data, so lookups are efficient

A. Document Databases	C. Change Data Capture	E. Stream processing
B. Relational Databases	D. Graph Databases	F. API

8. **Pick 1** of the below two options to respond to in 4-6 sentences. Circle whichever you are responding to.
   - (a) One of your team members comes back from an industry conference where a speaker was discussing the merits of keep the data system as efficient as possible by keeping just a single copy of all data files at any given time. How might you respond?
   - (b) One of your team members comes back from an industry conference where a speaker was discussing the merits of keeping the data system as simple as possible by keeping all variables/observations of data in one table in your SQL database. How might you respond?

9. Suppose you have a folder tree given by the below image. For each of the following tasks, write a BASH command what would accomplish the desired task.



(a) You are currently in the folder named `practice` and desire to copy the file named `important.txt` in that folder into the different folder named `class`. *You can use only relative paths.*

Solution:

```
cp important.txt ../../documents/class/important.txt
```

(b) You are currently in the folder `class` and want to know how many files or folders are inside the `src` folder. You can use either relative or absolute paths.

Solution:

```
ls /server/src | wc
```

(c) You are currently in the `work` folder and would like to write to the file `contents.txt` a list of all the files currently in this folder.

Solution:

```
ls > contents.txt
```

10. On your laptop, you have defined an `.ssh/config` file with the following contents:

```
Host lab
    HostName 123.456.789.007
    User rob
    Port 1412

Host work
    HostName 314.159.265.358
    User robert

Host homeserv
    HostName 271.828.182.845
    User bobby
```

All remotes are running an SSH server, and SSH keys have been set up for each. Use this information to write out BASH commands that could be run *on your laptop* to achieve the following tasks.

(a) Copy the file `results.csv` from the home directory of the lab server over to the home directory of the work server, keeping the filename the same.

Solution:

```
scp lab:results.csv work:results.csv
```

(b) Display the contents of the file `todos.txt` in the home directory of the home server on your laptop screen.

Solution:

```
ssh homeserv cat todos.txt
```