-WARNING: Google docs sometimes do strange things to quote marks - it turns straight quotes into curly quotes, and the shell doesn’t like curly quotes!

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "generic/debian12"
  config.vm.synced_folder ".", "/vagrant"
  #config.vm.network "forwarded_port", guest: 8080, host: 8080
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update -y
    apt-get install -y git git-man apt-file
    apt install -y gdb
    apt install -y maven
    apt install -y openjdk-17-jdk
    apt install -y unzip
    apt install -y maven
    apt install -y openjdk-17-jdk
    apt install -y mariadb-{server,client}
    apt-get install -y libreadline-dev
    apt install -y shellcheck
    systemctl start mariadb
    systemctl enable mariadb
    mysqladmin -u root -p create mydatabase
    mysql -u root -p -e 'source /vagrant/sample-data.sql'
SHELL
config.vm.provision :shell, privileged: false, inline: <<-SHELL
    git config --global user.name "ali"
    git config --global user.email "jardine64@gmail.com"
    SHELL
  config.vm.provision "file", source: "~/.vimrc", destination: "~/.vimrc"
end
```

Practice Exam

QUESTION

You call fork() and see a return value of 404. Which of the following is correct?

Fork succeeded, you are in the child process, and the pid of the parent was 404.

Fork succeeded, and the new process belongs to the user with a user id of 404.

The call failed because the process you were trying to fork was not found.

Fork succeeded, you are in the parent process, and the pid of the child process is 404.

You need to check the errno variable to see if there was an error during the fork call. If there was no error, then you are in the parent process, and the pid of the child is 404.

You need to check the errno variable to see if there was an error during the fork call. If there was no error, then you are in the child process, and the pid of the parent was 404.

2 points

QUESTION

Type the command (one word) that you would use on a Linux system to print the name of the current user to the terminal: whoami

2 points

QUESTION

Given a file that shows up as follows in ls -l:

-rw-rw-r-- 1 Breda staff 1024 Jan 1 10:01 logfile

Suppose that group users contains Alice, Breda and Carole; group tech contains Alice and Breda; and group staff contains Breda and Carole.
What are the access rights for the three users on this file?

- Alice: Can read, but not write or execute
- Can read, write and execute.Can read and write, but not execute.Can read, can not write, but can execute.Can read, but not write or execute.Can not read, but can write and execute.Can not read, can write, can not execute.Can not read or write, but can execute.Can neither read, write nor execute.
- Breda: Can read and write, but not execute (because she's the user, and also because she's in the staff group.)
- Can read, write and execute.Can read and write, but not execute.Can read, can not write, but can execute.Can read, but not write or execute.Can not read, but can write and execute.Can not read, can write, can not execute.Can not read or write, but can execute.Can neither read, write nor execute.
- Carole: Can read and write, but not execute (because she's in the staff group)
- Can read, write and execute.Can read and write, but not execute.Can read, can not write, but can execute.Can read, but not write or execute.Can not read, but can write and execute.Can not read, can write, can not execute.Can not read or write, but can execute.Can neither read, write nor execute.

In all cases, "execute" refers to running the program as ./logfile.

2 points

QUESTION

A colleague has written a shell script called star. When they directly address it as ./star or /home/bob/code/star, it works, but just typing star gives an error message. Mark the following as true or false:

- True False The colleague has forgotten to set the +x bit on the script.
- True False To run a shell script without the ./ prefix, the file has to have the extension .sh.
- True False The error message appears because the folder containing the script is not on the shell's PATH.

3 points

QUESTION

The command sed (stream editor) reads from standard input, applies a transformation based on its arguments and prints the result on standard output.
One of its arguments is described in the manual page as follows:
y/source/dest/
  Transliterate the characters in the pattern space which appear in source

to the corresponding character in dest.

The pattern space refers to the input.
For example, echo aabbccddee | sed y/bc/xb/ would print aaxxbbddee.
Based on this information, consider the following commands (lines starting with $ are commands, the rest is output):

```
$ cat colors.txt
Gules
Azure
Vert
Sable
Purpure
$ cat colors.txt | sed y/aeiou/eioua/ | tail -n 3
```

What will be printed on the second line of the output of the last command?
Enter your answer in the text box exactly as the command would print it.
Sebli

2 points

QUESTION

You are currently on branch develop and you would like to merge commits from branch fred into branch george. Complete the git commands to do this, assuming you have no uncommitted changes in your local files. You do not need to do any git status to check the state of anything for this question.
You must not type any leading or trailing spaces, and you must use exactly one space where spaces are required. Do not use any quote marks.
(Your terminal will not mind if you break these rules, but the automarker will and you will lose marks for it.)

- git checkout george
- git merge fred

2 points

QUESTION

Select the correct statement describing the normal location of certain files in a POSIX system.

/bin contains the "recycle bin" where deleted files are moved to. /usr contains folders that can be written to by normal (non-root) users. /etc contains mount points for removable drives such as USB sticks or DVDs.

/bin contains executable programs. /home contains folders that can be written to by normal (non-root) users. /etc contains system-wide configuration files.

/bin contains executable programs. /usr contains folders that can be written to by normal (non-root) users. /etc contains mount points for removable drives such as USB sticks or DVDs.

/bin contains the "recycle bin" where deleted files are moved to. /home contains folders that can be written to by normal (non-root) users. /etc contains system-wide configuration files.

/bin contains the "recycle bin" where deleted files are moved to. /home contains folders that can be written to by normal (non-root) users. /etc contains mount points for removable drives such as USB sticks or DVDs.

/bin contains the "recycle bin" where deleted files are moved to. /usr contains folders that can be written to by normal (non-root) users. /etc contains system-wide configuration files.

/bin contains executable programs. /home contains folders that can be written to by normal (non-root) users. /etc contains mount points for removable drives such as USB sticks or DVDs.

/bin contains executable programs. /usr contains folders that can be written to by normal (non-root) users. /etc contains system-wide configuration files.

2 points

QUESTION

The following script, with line numbers added, is an attempt to implement the "compiler helper exercise" to make a script called b that, when called with either ./b build example or ./b build example.c, should first compile the C program example.c to example and, if and only if this was successful, then run it.

```
1 | #!/bin/sh
2 | # Build helper script
3 | if [ $# -lt 2 ]
4 | then
 5 |     echo "Use: $0 [compile|build|run] PROGRAM"
 6 |     exit 2
 7 | fi
 8 | BASE="$(basename $2)"
 9 | SOURCE="${BASE}.c"
10 | case $1 in
```

```
11 |     compile)
12 |         gcc -Wall -std=c99 "$SOURCE" -o "$BASE"
13 |         ;;
14 |     run)
15 |         exec "./$BASE"
16 |         ;;
17 |     build)
18 |         $0 compile "$2" || $0 run "$2"
19 |         ;;
20 |     *)
21 |         echo "Unknown command '$1', valid ones are compile, build, run."
22 |         ;;
23 | esac
```

However, the script contains two mistakes. Where are they? Enter the affected lines as integers.

- The first mistake is on line
- 8
- The second mistake is on line
- 18

2 points

QUESTION

Enter the commands that you would run on a Debian Linux system as the root user to perform the following system administration tasks.
You must enter your commands with no leading or trailing spaces, no quote marks, and exactly one space between arguments. This is due to the way the automarker works, even if your terminal/shell would be more forgiving.

- Install the package called redstone (assume that it exists):
- apt install redstone
- Fetch the latest version of the catalog file from the repository server, that shows what the latest version of each package is:
- `apt-get update`
- `apt update (is the better version - it’s more modern)`

2 points

QUESTION

Enter the command that you would use in the shell to make a copy of a file called ruby. The copied file should be called diamond.

Enter your command with exactly one space between arguments, no quote marks and no slashes (you can assume all files affected are in the current folder, and readable/writable).

cp ruby diamond

QUESTION

Enter the command that you would use in the shell to delete a file called ruby

Enter your command with exactly one space between arguments, no quote marks and no slashes (you can assume all files affected are in the current folder, and readable/writable).

rm ruby

WEEK 5 Databases (SQL)

Normal Forms

1st normal form definition - Each table should have a primary key, and each column should only contain one value. The order of rows should not be used to convey meaning.

2nd normal form definition - Each non-key attribute in a table is fully dependent on the entirety of the primary key(s).

3rd normal form definition - Every non-key attribute in a table should depend solely on the primary key.

Schema 1

A school's database looks like this (it was set up by someone more used to spreadsheets):

stuId	name	gender	unit	grade
101	Fred	M	Mathematics	75
101	Fred	M	German	65
101	Fred	M	English	90
102	Sam	X	Mathematics	60
102	Sam	X	English	60
...	...	...	...	...

stuId is a student id that is unique per student. Students' names are not required to be unique, i.e. you can have two 'Fred's in the school. Gender is one of {M, F, X}. For each student and each unit they take, there is one row containing among other things the student name, unit name and the grade (0-100) that the student got on this unit. In the example above, we can see that Fred took three units (Mathematics, German and English). No two units have the same name but a unit name can appear several times in the database since many students can take the same unit. The first row of the example tells us that there is a student called Fred with id 101, who is male, and took the Mathematics unit and got a grade of 75 on it.

Students:

- Student Id
- Name
- Gender

Unit

- Unit Id
- Unit Name

Grades

- Student Id
- Unit Id
- Grade

Schema 2

The CIA world factbook contains geographical, political and military information about the world. Here is part of one table listing principal cities from 2015:

*city	country	pop	co_pop	capital
...	...	...	...	...
Paris	France	10.8M	66.8M	yes
Lyon	France	1.6M	66.8M	no
Marseille	France	1.6M	66.8M	no
Papeete	French Polynesia	133K	285K	yes
Libreville	Gabon	707K	1.7M	yes
...	...	...	...	...

We will assume for this exercise that city names are globally unique and therefore the "City" column has been chosen as the primary key for this table. The "pop" column lists the city's population and the "co_pop" lists the population of the country in which the city is located (with abbreviations K = 1000, M=1000000). The "capital" column is a Boolean yes/no value that is set to "yes" for exactly one city in each country. (While the capital is included in the table for every country however small, non-captial cities are only included if they are of international significance.)

City

- City_Name
- City_Population
- Country_Name

Country

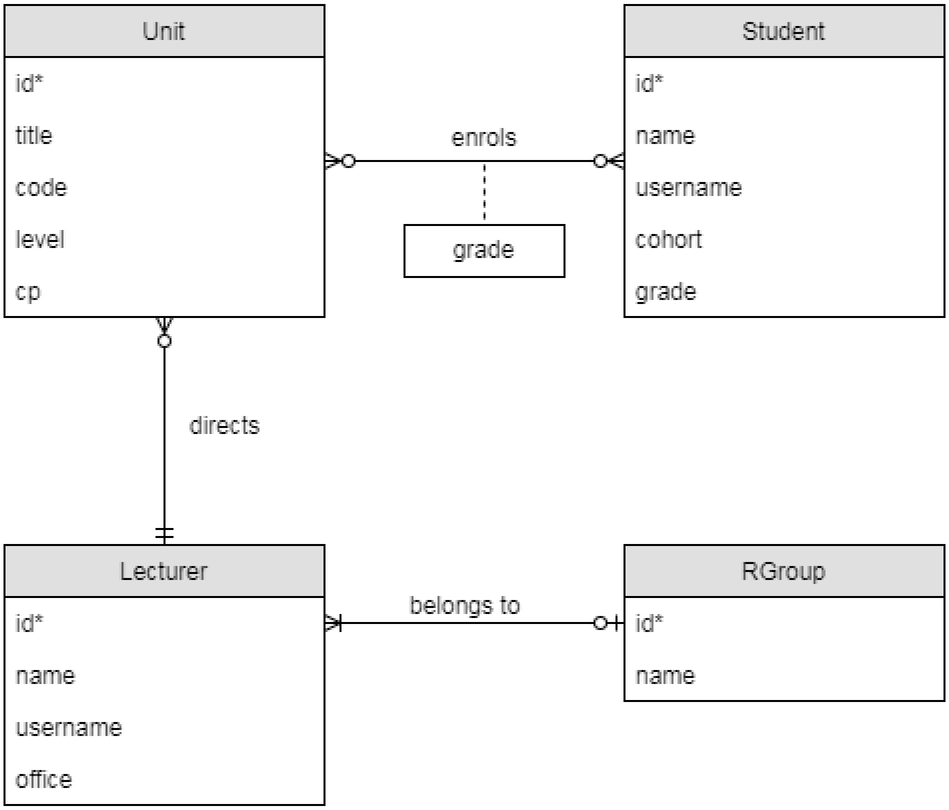
- Country_Name
- Country_Population
- Capital_Name

SQL

https://gvwilson.github.io/sql-tutorial/

WARNING: I screwed up the SQL queries when I sent them through an auto formatter (some of the tables are named census instead of Census, for example.)

Here is an ER diagram for a fictional university database:



The foreign key columns are not included in the tables - in this diagram, they are implied by the relationships, e.g. the Unit.director column comes from the "directs" relationship.

Looking at the diagram and the table schemas, answer the following questions for yourself:

- Which relationships are mandatory or optional? (For example, must every unit have at least one student enrolled?)
  - Optional:
    - Lecturer to RGroup
    - Student to Unit / Unit to Student
    - Lecturer to Unit
  - Mandatory:
    - Unit to Lecturer
    - RGroup to Lecturer
- Which relationships are one-one, one-many or many-many?
  - Lecturer to RGroup is a 1 to 0 or 1
  - Unit to Student is many to many (and the reverse)
  - Unit to student is a 0 or many to 1 relationship.
- How do the above affect the placement of foreign keys? For example, why is the foreign key for "lecturer belongs to research group" on the Lecturer table?

  - You will need research_group_id (the foreign key) in the lecturer table if you want to join the lecturers table to the research group table.

https://app.diagrams.net/#G1hG7Ta2BqUkGlLMSIWKXsD60Bp3KdBwyT#%7B%22pageId%22%3A%22fQq6pHCtnSW_8-8me6HY%22%7D Tried and failed to do the diagram (couldn’t figure out how to draw the lines with the feet properly)

The University of Bristol Hoverboard Society (HovSoc) wants to create a database to manage its membership and events. Each member has a name, an optional student number, a contact e-mail address and a hoverboard riding skill level (represented as an integer, minimum 0). We assume that e-mail addresses are unique among members.

The committee consists of some of the members, each of which has a unique committee role. We assume that committee roles do not change during the year and that each committee role must be filled every year.

An event has a date, a name, a location, an optional description and an organiser who must be a society member (not necessarily a committee member). An event is attended by a set of members. There is never more than one event at the same location on the same date but event names are not unique.

```
drop table if exists Events;
drop table if exists Committee;
drop table if exists Members;

CREATE TABLE IF NOT exists Members (
 email VARCHAR(100) NOT NULL,
 name VARCHAR(100) NOT NULL,
 student_number INT,
 skill_level INT,
 PRIMARY KEY (email)
);

CREATE TABLE IF NOT exists Committee (
 role VARCHAR(100) NOT NULL,
 year INT NOT NULL,
 member_email VARCHAR(100),
FOREIGN KEY (member_email) REFERENCES Members(email),
 PRIMARY KEY (role, year)
);


CREATE TABLE IF NOT exists Events (
 date VARCHAR(100) NOT NULL,
 name VARCHAR(100) NOT NULL,
 location VARCHAR(100),
 description VARCHAR(100),
 organizer VARCHAR(100),
attendee_email VARCHAR(100),
FOREIGN KEY (attendee_email) REFERENCES Members(email),
 PRIMARY KEY (date, name)
);
```

- Model a university database

Students:

- student_name, student_id

Units

- unit_name, unit_id, student_id, is_resit

Marks

- student_id, unit_id, mark, mark_is_confirmed

- List the names of all parties that stood in the election, ordered alphabetically by name.

```
SELECT name
FROM   party
ORDER  BY name;
```

- List the names of all parties that stood in the Bedminster ward.

```
SELECT   party.name
FROM     candidate
JOIN     party
WHERE    candidate.ward=
         (
                SELECT id
                FROM   ward
                WHERE  name ='Bedminster')
GROUP BY 1;
```

- How many votes did Labour get in the Stockwood ward?

```
SELECT party.name           AS party,
       SUM(candidate.votes) AS votes
FROM   candidate
       INNER JOIN party
               ON candidate.party = party.id
       INNER JOIN ward
               ON candidate.ward = ward.id
WHERE  ward.name = 'Stockwood'
       AND party.name = 'Labour'
GROUP  BY 1;
```

- List the names, parties and number of votes obtained for all candidates in the Southville ward. Order the candidates by number of votes obtained descending (winner comes first).

```
SELECT candidate.name       AS name,
       party.name           AS party,
       SUM(candidate.votes) AS votes
FROM   candidate
       INNER JOIN party
               ON candidate.party = party.id
       INNER JOIN ward
```

```
               ON candidate.ward = ward.id
WHERE  ward.name = 'Southville'
GROUP  BY 1,2
ORDER  BY 3 DESC;
```

- List the name, party and number of votes obtained for the winner only in the Knowle ward. (Hint: apart from changing the ward name, you only need one small modification to the statement from the last question. You may assume no ties.)

Oc

1. The university of Bristol is situated in the Cabot ward (ward names are not always distinct, but this one is). Find the names and codes of the CLU, region and country containing the Cabot ward (CLU = county level unit = "row in County table").

```
SELECT county.name   AS county_name,
       county.code   AS county_CLU,
       county.parent AS region_code,
       ward.code     AS ward_code,
        ward.name     AS ward_name,
        region.name AS region_name
        country.code as country_code,
        country.name as country_name
FROM   county
       LEFT JOIN ward
              ON county.code = ward.parent
      LEFT JOIN region
                  ON region.code = county.parent
      LEFT JOIN country
                  ON country.code = region.parent
WHERE  ward.name = 'Cabot';
```

2. If you used multiple SQL queries for the last question, do it in one single query now. (In other words, find a join strategy for the tables you need.)
3. Find the number of women in occupation class 1 (managers etc.) in the Cabot ward. You may use ward code for Cabot that you found in the first query and the occupation id 1 directly - you do not need any JOINs for this query.

```
SELECT SUM(data) AS num
FROM   Statistic
WHERE  gender = 1
       AND occid = 1
       AND wardid = 'E05001979';
```

4. For the Stoke Bishop ward (E05002003), list the 9 occupation class names and the number of men in each occupation. Your table should have two columns called name and number. You can use the provided ward code, you do not need to join on the ward name.

```
SELECT Occupation.name,
       SUM(Statistic.data) AS number
FROM   Occupation
       LEFT JOIN Statistic
              ON Occupation.id = Statistic.occid
                 AND Statistic.gender = 0
                 AND Statistic.wardid = 'E05002003'
GROUP  BY Occupation.name;
```

We will soon be able to do more interesting statistical queries on the census data but for that we need SQL's statistical functions which we will learn next week.

Here's a slightly more tricky question to finish off with. It can be done with only the techniques that we have learnt so far.

- Find all ward names that are not unique, and print them in alphabetical order (only once each).

There are 400 distinct such names in total (for example, there are 21 wards called 'Abbey') so your query will produce quite a long table. Your query might also take a while to execute, there are faster ways to do this but not with the material we've learnt so far. The table starts "Abbey, Alexandra, All Saints" and ends "Worsley, Wyke, Yarborough".

```
SELECT name,
       COUNT(name) AS c
FROM   Ward
GROUP  BY 1
HAVING c > 1
ORDER  BY 1 ASC;
```

Sam contribution:

```
SELECT DISTINCT Ward.name
FROM Ward
WHERE Ward.name IN (
     SELECT Ward.name
     FROM Ward
     GROUP BY Ward.name
     HAVING COUNT(*) > 1
)
ORDER BY Ward.name ASC;
```

- How many votes were cast in all of Bristol in the 2014 elections?

```
SELECT SUM(votes)
FROM   candidate
```

- How many votes were cast in the 'Windmill Hill' ward and what percentage of the electorate in this ward does this represent? Your statement should produce a table with one row and two columns called 'votes' and 'percentage'.

```
with t as (
select electorate from Ward where name = 'Windmill Hill'
),
t2 as (
select sum(votes) as votes from Candidate join Ward ON Candidate.ward =
Ward.id
where Ward.name ='Windmill Hill')
SELECT t2.votes, t2.votes/t.electorate as percentage
FROM t,t2;
```

- List the names, parties and percentage of votes obtained for all candidates in the Southville ward. Order the candidates by percentage of votes obtained descending.

```
WITH t AS (SELECT candidate.NAME       AS can_name,
                  party.NAME          AS party_name,
                  SUM(candidate.votes) AS votes
           FROM   candidate
                  JOIN ward
                    ON candidate.ward = ward.id
                  JOIN party
                    ON party.id = candidate.party
           WHERE  ward.NAME = 'Southville'
           GROUP  BY 1,2),
       t2 AS (SELECT SUM(votes) AS total_votes
           FROM   candidate)
  SELECT can_name,
         party_name,
         votes / t2.total_votes AS percentage
  FROM   t,
         t2
  ORDER  BY 3 DESC;
```

- How successful (in % of votes cast) was the Conservative party in each ward?

```
WITH t
     AS (SELECT party.NAME           AS party_name,
                ward.NAME            AS ward_name,
                SUM(candidate.votes) AS votes
         FROM   candidate
                JOIN ward
                  ON candidate.ward = ward.id
                JOIN party
                  ON party.id = candidate.party
         GROUP  BY 1,2),
     t2
     AS (SELECT SUM(votes) AS total_votes
         FROM   candidate)
SELECT ward_name,
       votes / t2.total_votes AS percentage
FROM   t,
       t2
WHERE  party_name = 'Conservative'
ORDER  BY 2 DESC;
```

- Which rank did Labour end up in the 'Whitchurch Park' ward? Your statement should produce a table with a single row and column containing the answer as a number. You can assume no ties.

```
WITH t
     AS (SELECT party.NAME           AS party_name,
                ward.NAME            AS ward_name,
                SUM(candidate.votes) AS votes
```

```
         FROM   candidate
                JOIN ward
                  ON candidate.ward = ward.id
                JOIN party
                  ON party.id = candidate.party
         GROUP  BY 1,2),
     t2
     AS (SELECT t.party_name,
                t.ward_name,
                RANK()
                  OVER (
                    partition BY ward_name
                    ORDER BY votes DESC ) AS rank
         FROM   t)
SELECT rank
FROM   t2
WHERE  ward_name = 'Whitchurch Park'
       AND party_name = 'Labour';
```

- What is the total number of votes that each party got in the elections? Your result should be a table with two columns party, votes.

```
SELECT party.name           AS party_name,
       SUM(candidate.votes) AS votes
FROM   candidate
       JOIN party
         ON party.id = candidate.party
GROUP  BY 1;
```

- Find all wards where the Green party beat Labour and create a table with two columns ward, difference where the difference column is the number of Green votes minus the number of Labour votes. Your table should be ordered by difference, with the highest one first.

```
WITH t
     AS (SELECT party.NAME           AS party_name,
                ward.NAME            AS ward_name,
                SUM(candidate.votes) AS votes
         FROM   candidate
                JOIN ward
```

```
                  ON candidate.ward = ward.id
                JOIN party
                  ON party.id = candidate.party
         GROUP  BY 1, 2),
     t2
     AS (SELECT t.ward_name,
                SUM(CASE
                      WHEN t.party_name = 'Labour' THEN t.votes
                    END) AS labour_votes,
                SUM(CASE
                      WHEN t.party_name = 'Green' THEN t.votes
                    END) AS green_votes
         FROM   t
         GROUP  BY 1)
SELECT ward_name,
       green_votes - labour_votes AS difference
FROM   t2
WHERE  green_votes > labour_votes
ORDER  BY 2 DESC;
```

1. How many women work in sales and customer service occupations and live in the Cabot ward of Bristol (E05001979)?

```
SELECT SUM(s.data) AS women
FROM   statistic AS s
       JOIN occupation AS o
         ON s.occid = o.id
WHERE  s.wardid = 'E05001979'
       AND s.gender = 1
       AND o.name = 'Sales and customer service occupations';
```

2. How many people work in sales and customer service occupations and live in the Cabot ward of Bristol (E05001979)?

```
SELECT SUM(s.data) AS people
FROM   statistic AS s
       JOIN occupation AS o
         ON s.occid = o.id
WHERE  s.wardid = 'E05001979'
       AND o.name = 'Sales and customer service occupations';
```

3. How many people work in caring, leisure and other service occupations (occupation class 6) in all of the City of Bristol CLU (E06000023)?

```sql
SELECT SUM(s.data) AS folk
FROM   statistic AS s
       JOIN occupation AS o
         ON s.occid = o.id
       JOIN ward AS w
         ON s.wardid = w.code
WHERE  w.parent = 'E06000023'
       AND o.id = 6;
```

4. In the Cabot ward (E05001979), produce a table listing the names of the 9 occupation classes and the number of people in each of the classes in this ward.

```sql
SELECT o.name,
       SUM(s.data) AS d
FROM   occupation AS o
       LEFT JOIN statistic AS s
              ON s.occid = o.id
       LEFT JOIN ward AS w
              ON s.wardid = w.code
WHERE  w.code = 'E05001979'
GROUP  BY 1;
```

5. Find the working population, ward name and CLU name for the smallest ward (by working population) in the 2011 census.

```sql
SELECT w.name      AS ward_name,
       c.name      AS CLU_name,
       SUM(s.data) AS pop
FROM   statistic AS s
       JOIN ward AS w
         ON s.wardid = w.code
       JOIN county AS c
         ON w.parent = c.code
GROUP  BY 1,2
ORDER  BY 3 ASC
LIMIT  1;
```

6. The same as the last question, but now produce a table with two rows, one for the smallest and one for the largest ward. There's no quicker way than repeating the last query twice, the question is how to stick the two "copies" together.

```
(SELECT w.name      AS ward_name,
        c.name      AS CLU_name,
        SUM(s.data) AS pop
 FROM   statistic AS s
        JOIN ward AS w
          ON s.wardid = w.code
        JOIN county AS c
          ON w.parent = c.code
 GROUP  BY 1,2
 ORDER  BY 3 ASC
 LIMIT  1)
UNION
(SELECT w.name      AS ward_name,
        c.name      AS CLU_name,
        SUM(s.data) AS pop
 FROM   statistic AS s
        JOIN ward AS w
          ON s.wardid = w.code
        JOIN county AS c
          ON w.parent = c.code
 GROUP  BY 1,2
 ORDER  BY 3 DESC
 LIMIT  1);
```

7. Find the average size of a ward's working population in the London (E12000007) region.

```
WITH t
     AS (SELECT w.NAME      AS ward_name,
                SUM(s.data) AS pop
         FROM   statistic AS s
                JOIN ward AS w
                  ON s.wardid = w.code
                JOIN county AS c
                  ON w.parent = c.code
         WHERE  c.parent = 'E12000007'
         GROUP  BY 1)
SELECT AVG(pop)
```

```
FROM   t;
```

8. The same as the last question but now for every region - your query should produce a table with one row per region. The intention here is not to repeat the above query 9 times.

```
WITH t
     AS (SELECT w.NAME      AS ward_name,
                c.parent    AS region_id,
                SUM(s.data) AS pop
         FROM   statistic AS s
                JOIN ward AS w
                  ON s.wardid = w.code
                JOIN county AS c
                  ON w.parent = c.code
         GROUP  BY 1,2)
SELECT region.NAME,
       Avg(pop)
FROM   t
       JOIN region
         ON t.region_id = region.code
GROUP  BY 1;
```

9. Produce a table that lists, for each of the 9 regions of England, the percentage of people in managerial (class 1) occupations who are women.

```
WITH t
     AS (SELECT r.NAME      AS region_name,
                SUM(CASE
                      WHEN s.gender = 0 THEN s.data
                    END)    AS males,
                SUM(CASE
                      WHEN s.gender = 1 THEN s.data
                    END)    AS females,
                SUM(s.data) AS total
         FROM   statistic AS s
                JOIN ward AS w
                  ON s.wardid = w.code
                JOIN county AS c
                  ON w.parent = c.code
                JOIN region AS r
                  ON c.parent = r.code
                JOIN occupation AS o
                  ON s.occid = o.id
         WHERE  o.id = 1
         GROUP  BY 1)
SELECT region_name,
       females / total AS percent_female_managers
FROM   t
GROUP  BY 1;
```

10. For all CLUs in the London (E12000007) region, produce a table with three columns called CLU, occupation and count such that:
    a. CLU is the CLU name.
    b. count is the number of people of the occupation class in question in the given CLU.
    c. occupation is the name of the occupation class.
    d. Only rows with count >= 10000 appear in the table.
    e. The table is sorted by count ascending.

```
SELECT c.name      AS CLU,
       o.name      AS occupation,
       SUM(s.data) AS count
FROM   statistic AS s
       JOIN ward AS w
         ON s.wardid = w.code
       JOIN county AS c
         ON w.parent = c.code
       JOIN region AS r
         ON c.parent = r.code
       JOIN occupation AS o
         ON s.occid = o.id
WHERE  r.code = 'E12000007'
GROUP  BY 1,2
HAVING count >= 10000
ORDER  BY count DESC;
```

11. Create a table with three columns occupation, women and men and one row per occupation class.
    a. The occupation column should list the occupation class names.

b. The women and men columns in each row should list the total number of women resp. men in the row's occupation class in the whole dataset.
c. The intention here is not to have to copy-paste a subquery 9 times.

```
SELECT o.name   AS occupation,
       SUM(CASE
             WHEN s.gender = 0 THEN s.data
           end) AS males,
       SUM(CASE
             WHEN s.gender = 1 THEN s.data
           end) AS females
FROM   statistic AS s
       JOIN ward AS w
         ON s.wardid = w.code
       JOIN county AS c
         ON w.parent = c.code
       JOIN region AS r
         ON c.parent = r.code
       JOIN occupation AS o
         ON s.occid = o.id
GROUP  BY 1;
```

12. The same as question 9, but now with a 10th row in the table listing the value for all of England. You can use the string 'England' for the region column.

```
WITH t
AS
  (
           SELECT   r.name AS region_name,
                    SUM(
                    CASE
                             WHEN s.gender=0 THEN s.data
                    end) AS males,
                    SUM(
                    CASE
                             WHEN s.gender=1 THEN s.data
                    end)        AS females,
                    SUM(s.data) AS total
           FROM     statistic   AS s
           JOIN     ward        AS w
           ON       s.wardid = w.code
           JOIN     county AS c
           ON       w.parent = c.code
           JOIN     region AS r
           ON       c.parent = r.code
           JOIN     occupation AS o
           ON       s.occid =o.id
           WHERE    o.id = 1
           GROUP BY 1)

(SELECT region_name,
        females / total AS percent_female_managers
 FROM   t)
UNION
(SELECT "england"                 AS region_name,
        SUM(females) / SUM(total) AS percent_female_managers
 FROM   t
 GROUP  BY 1);
```

JDBC

specify arguments with

```
mvn exec:java  -Dexec.args="test"
```

WEEK 4 Debugging (GBD)

WEEK 3

4.1 File Permissions

4.2 Shell Scripting

Write a shell script in a file called b (for build) that does the following:

- Your script should run under any Bourne-compatible shell (e.g. not just bash), and it should be written so that you can call it with ./b.
- ./b compile NAME should compile the file of the given name, so for example ./b compile hello should run gcc -Wall -std=c11 -g hello.c -o hello.
- However, your script should accept both ./b compile hello and ./b compile hello.c as input, and do the same thing in both cases, namely compile hello.c. The output file for gcc in both cases should be called just hello.
- If the source file you provided as an argument does not exist (adding .c if necessary) then the script should print an error message and return a nonzero exit status - not invoke the C compiler.
- ./b run NAME should run the program, assuming it exists in the current folder, so both ./b run hello and ./b run hello.c should run ./hello. If it does not exist, again print an error message and exit with a nonzero status, don't try to run the program.
- ./b build NAME should first compile the C source file, and then if the compile was successful it should run the program. If the compile failed, it should not try and run the program.
- If you call ./b without any parameters, or ./b COMMAND with a command other than compile or run or build, it should print some information on how to use it. If you cal l ./b compile or another command with no filename at all, then the script should print an error message and exit with a nonzero exit status.

```
#! /bin/sh
USAGE="Usage: ./b filename[.c] compile/build/run"

if [ ${#} -ne 2 ]; then
echo "$USAGE"
exit 1
fi

if [ "$2" != "compile" ] && [ "$2" != "run" ] && [ "$2" != "build" ]; then
echo "$USAGE"
exit 1
fi
```

```
#strips the .c from the filename if it's given.
FILENAME=$(basename "${1}" .c)

# could also use test with the -f flag.
if [ ! -e "$FILENAME".c ]; then
echo "$1 doesn't exist"
exit 1
fi

if [ "$2" = "compile" ]; then
gcc -Wall -std=c11 -g "$FILENAME.c" -o "$FILENAME"
fi

if [ "$2" = "build" ]; then
gcc -Wall -std=c11 -g "$FILENAME.c" -o "$FILENAME" && ./"$FILENAME"
fi

if [ "$2" = "run" ]; then
    if [ ! -e "$FILENAME" ]; then
    echo "$FILENAME doesn't exist"
    exit 1
    fi
    ./"$FILENAME"
fi
```

Exercise: think of an example in a shell script where pipefail makes a difference, that is where the last command in a pipe could succeed even if a previous one fails. As a counter-example, cat FILE | grep STRING would fail even without pipefail if the file does not exist, because grep would immediately get end-of-file on standard input.

```
cat fake_file.txt | echo "success"
```

5. Build Tools

WEEK 2 (GIT)

WEEK 1

2.2 Pipes

Find one-line commands that do the following:
(Assuming your current directory is /usr/share/dict/)

1. **The first word in the file**

```
cat /usr/share/dict/words | head -n 1
```

Answer: A

2. **The last word in the file**

```
cat /usr/share/dict/words | tail -n 1
```

Answer: zygotes

3. **The number of words in the words file**

```
cat words | wc -w
```

Answer: 104334

4. **The 6171st word in the file**

```
cat /usr/share/dict/words | head -n 6171 | tail -n 1
```

Answer: Etna (in the old words list this was ‘Bristol’

5. **All words containing the letter Q, capitalised**

```
cat words | grep "Q"
```

[My answer with wc -w was 100 words]

6. **All words starting with the letter X (where x would also count)**

```
cat /usr/share/dict/words | grep -i "^X"
```

[My answer with wc -w was 106 words]

7. **All words ending in j**

```
cat words | grep 'j$'
```

[My answer with wc -w was 7 words]

8. **The number of words containing the letter Q, capitalised or not**

```
cat words | grep -i "Q" | wc -w
```

Answer: 1600

9. **The first 5 words containing the letter sequence cl**

```
cat words | grep "cl" | head -5
```

Answer: Barclay, Barclay’s, Chiclets, Chiclets’s, Cyclades

10. **All words containing the sequence “kp” but not “ckp”**

```
cat words | grep "kp" | grep -v "ckp"
cat /usr/share/dict/words | grep "[^c]kp"
```

Answer: Kirkpatrick, Kirkpatrick’s, breakpoints, workplace, workplace’s, workplaces

11. **The last 15 words of exactly two letters**

```
cat words | grep '^..$' | tail -15
 cat /usr/share/dict/words | grep -E "^.{2}$" | tail -n 15
```

Answer: um, up, us, vi, vs, we, wk, wt, xi, xv, xx, yd, ye, yo, yr

12. **All words from the first 100 words of the list which contain the letter y (I’m assuming lower case only)**

```
cat words | head -100 | grep "y"
```

Answer: Aaliyah, Aaliyah’s, Abby, Abby’s, Abernathy, Abernathy’s

13. **The first five words from the last 100 words on the list which contain the letter y (case insensitive)**

```
cat words | tail -100 | grep -i "y" | head -5
```

Answer: zestfully, zippy, zoology, zoology’s, zygote

14. **All three-letter words with no vowels (case insensitive)**

```
cat words | grep "^...$" | grep -v '[aeiou]'
```

’

```
cat /usr/share/dict/words | grep -E '^[^aeiou]{3}$'
```

15. **All words of exactly 7 letters, where the third one is an e and the word ends "-ded". This kind of search is really useful for crosswords. There are 14 words of this form, can you guess them?**

```
cat /usr/share/dict/words | grep -Ei '^..e.ded$'
```

```
amended blended breaded dreaded emended fielded kneaded pleaded
sledded speeded trended upended wielded yielded
```

Bonus: Find all words that start with a P (whether capitalised or not), and contain at least four instances of the letter a. Putting a * after something in a regular expression searches for any number of repetitions of this, including 0 so for example 'a*' would find words with any number of the letter a, including 0 (which is not what you want here). You need single quotes to stop the shell from expanding the *. Can you guess the words? There are 14 hits in the solution but essentially five words: two demonyms and three nouns which are not proper nouns, all with possessive and plural forms (bar one which is its own plural).

```
cat /usr/share/dict/words | grep -Ei '^P.*a.*a.*a.*a.*'
```

```
Panamanian Panamanian's Panamanians Paraguayan Paraguayan's Paraguayans
paraphernalia paraphernalia's parliamentarian parliamentarian's
parliamentarians phantasmagoria phantasmagoria's phantasmagorias
```

2.3 Regular Expressions

1. **Study the documentation for the -w option. Contrive a file such that grep PATTERN FILE returns two different lines but grep -w PATTERN FILE returns only one line.**

```
    -w, --word-regexp
            The expression is searched for as a word (as if
surrounded by ‘[[:<:]]’ and ‘[[:>:]]’; see re_format(7)).  This
option has no effect if -x is also specified.
```

File could be:

```
a
aa
```

So:

```
printf "a\naa" > test.txt // just creates that file.
grep a test.txt // gives 2
grep -w a test.txt // gives 1
```

2. **You'll have seen beforehand that you can count the results of a search with grep PATTERN FILE | wc -l. However, grep also has a -c option which counts matches. Can you find the situation where the wc -l approach and the -c approach produce different results? Can you explain why?**

wc -l counts the exact match of a word pattern. -c counts the number of lines that the pattern appears. so in a situation where in one line " PATTERN PATTERN PATTERN" wc = 3 but -c will be 1

3. **Some words have different spelling between British English and American English. For example, 'encyclopaedia' is valid in British English but not American. Can you write a regular expression that would match both of these words, but nothing else? How about matching both 'color' (American) and 'colour' (British)?**

```
grep -Ei 'Encyclopa?edia' test.txt
grep -Ei 'colou?r' test.txt
```

4. **UK postcodes follow a general schema of two letters followed by one number, followed by an optional space, then another number, followed by two more letters. Can you write a regular expression that would match such sequences?**

```
grep -E '[A-Za-z]{2}[0-9] ?[0-9][A-Za-z]{2}' test.txt
```

5. **In practice, the above is a simplified version of the system, and a better UK postcode validator regex is known to be ^(([A-Z]{1,2}[0-9][A-Z0-9]?|ASCN|STHL|TDCU|BBND|[BFS]IQQ|PCRN|TKCA) ?[0-9][A-Z]{2}|BFPO ?[0-9]{1,4}|(KY[0-9]|MSR|VG|AI)[-]?[0-9]{4}|[A-Z]{2} ?[0-9]{2}|GE ?CX|GIR ?0A{2}|SAN ?TA1)$. Try breaking apart this monster to understand what is being tested, and find an example that would match the schema described for the fourth question but fail to match this expression.**

I recommend copying it into something like https://regexr.com/ which will help give colours to each block. It allows for B1 (instead of BS1) and also supports various other letter only post codes (ASCN etc).
Technically this regex only looks for A-Z, whereas the version above was more permissive and allowed lowercase letters.