Databases Coursework 1, Spring 2019

Overview

This coursework covers material from weeks 13-17.

Marks and weights

This coursework is worth 60% of your unit mark, divided up as follows:

- Two SQL exercises.
- One modelling exercise.
- One normalisation exercise.

Each exercise is worth 25% of coursework 1 which is 15% of the unit mark.

Submission rules

For this coursework, you must submit the following. Anything else will incur a penalty. If you submit a file in an incorrect format (i.e. word document where PDF is required) your work will not be marked and you may automatically get 0 marks for the exercise in question.

- SQL exercises: your solutions must be placed in the template SQL scripts at the indicated positions and you must not modify the file name or layout. This is because the scripts will be read by a program that relies on the markers in the file to distinguish which lines belong to which question.
- Modelling exercise: you must submit one SQL script and one PDF file containing your solutions to this exercise. Call your files "modelling.sql" and "modelling.pdf".
- Normalisation exercise: you must submit one PDF file called "normalisation.pdf".

Your submission must be exactly five files: two PDF files and three SQL scripts.

Marking is anonymous: your files must not contain your name, username, e-mail address etc. Instead, you should put your candidate number (a 5-digit number – not your username) in your PDF files.

SQL exercises

Your SQL scripts must run on MariaDB as installed on the lab machines. You may use your own computer but if you use a different database or a different version and use features or syntax that do not work in on the lab machines, you will lose marks. For example, MariaDB ignores column-level foreign key constraints and does not support inline views using the WITH keyword on the lab version so any use of these will count as a mistake.

You are encouraged to check your solutions to the SQL exercises in three ways before submitting them:

- Run the file on a lab machine (with the `\.` command in MariaDB) and check both that the file runs without errors and that it produces the output you were expecting.
- Compare your results (the tables output by MariaDB) with other students.
- Run the provided checker script. This will tell you whether or not the syntax of the marker comments in the file is correct for the program that will be used to help mark it. The checker script does not tell you if your solutions are correct or not!

Diagramming software

The modelling exercise asks you to draw a diagram and you may want a diagram for the normalisation exercise too. It is probably quickest to draw this by hand and scan it in – remember the file you submit must be in PDF format – or you may use diagramming software. `draw.io` is a free online service that lets you draw diagrams using Crow's Foot notation (in the "entity relation" section).

Plagiarism

- This coursework is individual work.
- The files that you submit must be your own work (except that you must use the templates provided for your SQL files).
- You may not share the contents of your submission, especially your SQL code, with any other students. You may compare the results that you get (the tables that are output by MariaDB in response to your queries) with other students.
- Your solutions must not be made available to anyone else – for example, if you want to store all your coursework on a git repository, make the repository private.
- You may not copy your solutions from previous years' students – we do check this.
- Any material that you use which is not your own work must be properly referenced.
- Material that is quoted or otherwise directly lifted from another source must be indicated as such.

You can discuss the coursework with other students and share tips. The limit between helping each other to learn and plagiarism is that you must not share any files, such as code, that you have submitted or intend to submit. You are allowed to share with other students the tables that get output by running your SQL code and you can that you get the same answers but you must not share the SQL code itself.

If you have a question about your code, ask the unit staff in the labs.

We had cases in the past where A sent their code to B who "just wanted to have a look" but then uploaded a copy of A's code almost unchanged. Both students were caught (this kind of thing is not hard to spot) and punished for plagiarism.

Modelling exercise

This exercise is worth 25% of coursework 1. Your solutions for this exercise must be in a PDF file that you should call "modelling.pdf" and a create/drop script called "modelling.sql". Include your 5-digit candidate number but not any other personally identifying information (name, username etc.) in the PDF file.

Pick an application or website/service of your choice (you may not choose Flickr, which is the example).

1. Describe in your own words some of the features of the application/website that you have chosen. You do not need to describe all the site's features, just the ones that you will work with in the other points of this exercise.
2. Give a possible schema showing how your chosen features could be backed using a database, by drawing an ER diagram in Crow's Foot notation containing at least the following:
    a. 3 tables
    b. one one-to-many relationship
    c. one many-to-many relationship
3. Write a create/drop script for the exact tables in your model.
4. Describe two use cases of your chosen application or website in terms of SQL queries that involve some form of JOIN. Your queries should be consistent with your ER diagram and SQL script.

A create-drop script as discussed in the lectures is an SQL script that first drops all tables (using DROP TABLE IF EXISTS) and then creates the tables. The order of the DROP and CREATE statements must be such that the script runs through whether or not the tables exist, which means you need to pay attention to foreign key relationships. A create-drop script also does not contain any data to insert into the tables (in a real application, such "default" data would go into a separate script).

In your use cases, you may need to include parameters such as "the username of the current user" in a page to display a user's profile. In this case, put a ? mark in your query to indicate the parameter(s) and explain which parameter is which.

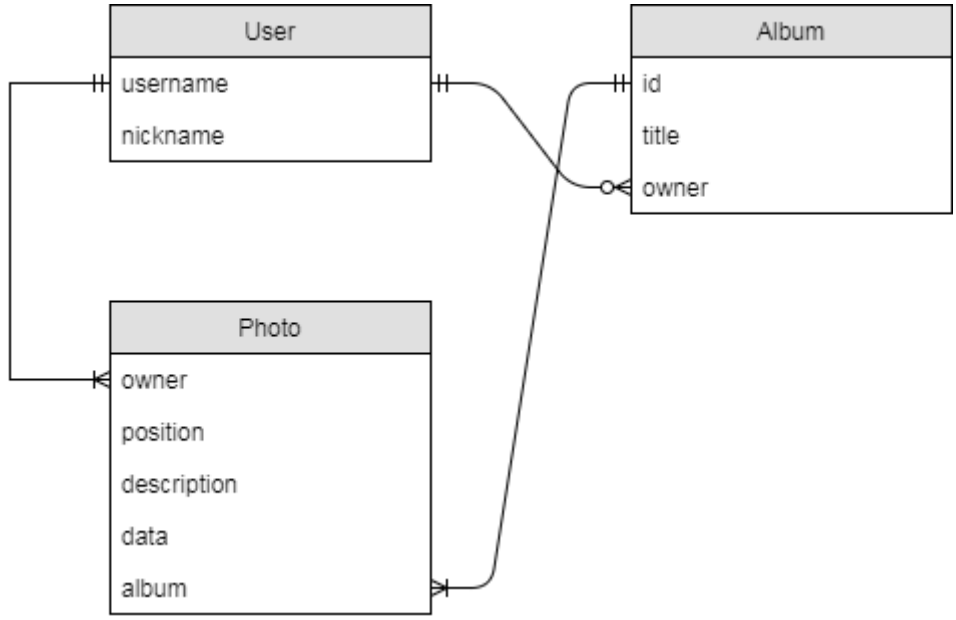
Aim for roughly 1-2 pages of text in your solution for this exercise, not including the diagram or the SQL script. The following Flickr example is deliberately kept very brief.

Example - Flickr

Note: many of the choices made in this example are bad choices.

Flickr is a website where people can upload, edit, link to and share photos. Photos can be sorted into albums.

The main entities are User, Album and Photo with the following relationships:



The following lines would go in your create/drop script.

```
DROP TABLE IF EXISTS Photo;
DROP TABLE IF EXISTS Album;
DROP TABLE IF EXISTS User;

CREATE TABLE User (
  username VARCHAR(100) PRIMARY KEY,
  nickname VARCHAR(100)
);

CREATE TABLE Album (
  id INTEGER PRIMARY KEY, -- albums are public
  owner VARCHAR(100),
  title VARCHAR(100),
  FOREIGN KEY (owner) REFERENCES User(username)
);

CREATE TABLE Photo (
  owner VARCHAR(100),
  position INTEGER,
  description VARCHAR(1000) NULL,
  data BLOB,
  album INTEGER NULL,
  FOREIGN KEY (album) REFERENCES Album(id),
  FOREIGN KEY (owner) REFERENCES User(username),
  PRIMARY KEY (owner, position)
);
```

Our use cases are:

1. Viewing a user's photostream displays their latest photos, hovering over a photo shows the user's nickname and the photo description. This could be done with the following query where ? is the username of the user whose stream we are looking at:

```
SELECT p.description, p.data, u.nickname FROM Photo p
JOIN User u ON u.username = p.owner
WHERE u.username = ?
ORDER BY p.position DESC;
```

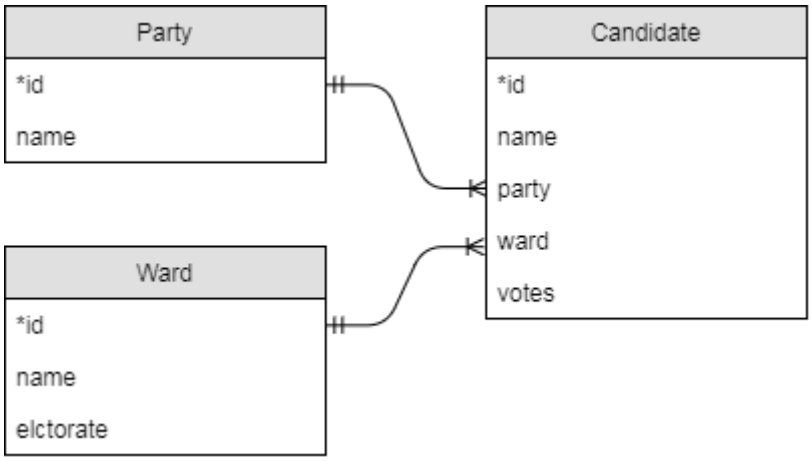
2. Browsing an album shows all photos in that album, the most recent (highest index) first and the album title at the top of the page. Hovering over a photo still displays the description and nickname. This can be done with the following query, where ? is the id of the album to browse.

```
SELECT p.description, p.data, u.nickname, a.title
FROM Photo p JOIN Album a ON p.album = a.id
JOIN User u on u.username = p.owner
WHERE a.id = ?
ORDER BY p.position DESC;
```

SQL exercise 1 – elections

This exercise is worth 25% of coursework 1. Your solutions must be in a file named elections.sql following the provided template. You must not change or add any lines starting with "`-- !`" as your file will be read and run by a program that uses these markers to distinguish the answers to individual questions. Your file will also be checked be a human marker.

In the years 2014 and 2015, councillors were elected for many wards in Bristol. In this exercise you will analyse some of the election results. The data are in the following format / schema:



In the sample database, the election data is in two databases 'elections14' and 'elections15' both with the same schema. When you are logged into MariaDB, the commands `USE elections14;` and `USE elections15;` let you switch between these databases.

For the purposes of this exercise you may assume the following rules, which are "almost true" in practice:

- Each ward has exactly one seat on the City Council and the candidate with the largest number of votes wins the ward whether or not they have an absolute majority.
- Each party can field at most one candidate per ward. For the purposes of this exercise, "Independent" counts as a party[1].
- Candidate names are not guaranteed to be unique, not even within a ward[2].

[1] If there were two independent candidates in a ward, one would create two distinct entries such as "Independent 1" and "Independent 2" for them in the party row of the raw data table and these would become separate entries in the Party table too.
[2] In practice, there are no known cases of two candidates with the same name standing in the same ward. I imagine that this would be fun if it happened.

Working with normalised data

The election data is in a normalised format. Normalising breaks data up into multiple tables; the first thing you should do when working with a normalised schema is figure out the join strategy, that is how to "put it together again".

For complex queries, first think about how you would solve the problem with multiple queries (taking answers from one query to build the next one) then find a way to combine the queries into one. Techniques here include JOINs and subqueries.

For example, consider the following question.

0. What party did Mark Bradshaw stand for in the 2014 election?

Note that there's no guarantee there'll be only one "Mark Bradshaw" so we may get more than one result.

The information we have to start with is a candidate name, so let's look him up. Select the elections14 database (`USE elections14;`) and type the text in bold.

```
[elections14]> SELECT party FROM Candidate WHERE name = 'Mark Bradshaw';
+-------+
| party |
+-------+
| 6     |
+-------+
```

Ok, that's a party ID so we can look that up in the Party table in a second step:

```
[elections14]> SELECT name FROM Party WHERE id = 6;
+--------+
| name   |
+--------+
| Labour |
+--------+
```

That answers our question but we used two queries. In this case, all we have to do is join the tables together (there are now two columns called "name", one each for party and candidate, so we need to say which one we mean):

```
SELECT Party.name FROM Party
JOIN Candidate ON Candidate.party = Party.id
WHERE Candidate.name = 'Mark Bradshaw';
```

Done! Alternatively we could use a subselect:

```
SELECT name FROM Party WHERE id =
(SELECT party FROM Candidate WHERE name = 'Mark Bradshaw');
```

Tasks

Answer the following questions by providing exactly one SQL statement per question as your answer. Your SQL statement must provide the right answer for both the 2014 and 2015 databases – so you cannot e.g. hard-code any party IDs since these can vary between elections. The "solutions" for the 2014 elections are given along with the exercises – the data that is, not the SQL statements.

You are welcome to compare the data you get for the 2015 elections with other students. You must not share your SQL code with other students, however – that would be plagiarism.

Questions 1–5 are straightforward; questions 6–8 are a bit more involved (and so carry more marks).

1. **List the names of all parties that stood in the election, ordered by name.**
   The 2014 solution is: Conservative, Green, Independent, Independents for Bristol, Labour, Liberal Democrat, Respect, Trade Unionists and Socialists Against Cuts, UKIP.

2. **How many votes were cast in total across all of Bristol?** (This does not include spoiled ballots, and we assume each vote is for exactly one candidate.)
   The 2014 solution is: 78807.

3. **Which candidates stood in the Bedminster ward and how many votes did they get?** The order is not important for this question. The 2014 solution is:

Robin Victor Clapp	204
Thom Oliver	264
Sarah Helen Cleave	680
Catherine Slade	838
Mark Bradshaw	1745

4. **How many votes did the Liberal Democrats get in the Filwood ward?**
   The 2014 solution is 102.

5. **Produce a table with three columns called `name, party, votes` showing the results for the Hengrove ward, ordered by number of votes (highest first).**
   The 2014 solution is:

name	party	votes
Michael Frost	UKIP	912
Yvonne Eileen Clapp	Labour	780
Antony Skelding	Conservative	503
Sylvia Kathleen Doubell	Liberal Democrat	480
Graham Hugh Davey	Green	123
Neil Oliver Maggs	Respect	114
Mark Baker	Trade Unionists …	28

6. **Which rank (1st, 2nd, 3rd …) did Labour end up in Bishopsworth?**
   Your solution should be a table with one row and one column containing the answer as a number. Hint: count the number of parties that got the same number or more votes than Labour. You may assume no ties.
   The 2014 solution is 2 (Labour came second).

7. **How successful (in % of votes cast) was the Green party in each ward?**
   You do not need to sort the results. The 2014 solution is (one row per ward):

Avonmouth	4.3772
Bedminster	22.4605
…	…
Windmill Hill	24.0964

8. **Find all the wards where the Greens beat Labour and create a table with columns `ward,rel,abs` where `name` is the ward name, `rel` is the relative difference as a percentage of the *electorate* in the relevant ward and `abs` is the absolute difference in votes between the Greens and Labour in this ward.**
   The 2014 solution is:

ward	rel	abs
Bishopston	10.4630	1078
Redland	8.9236	776
Southville	3.8949	378
Stoke Bishop	0.5564	45

SQL exercise 2 – census

This exercise is worth 25% of coursework 1. Your solutions must be in a file named census.sql following the provided template. You must not change or add any lines starting with "-- !" as your file will be read and run by a program that uses these markers to distinguish the answers to individual questions. Your file will also be checked be a human marker.

Background – UK Geography

The United Kingdom of Great Britain and Northern Ireland (UK) is a country that contains the individual countries England, Wales, Scotland[3] and Northern Ireland[4]. The census data for this question comes from the Office of National Statistics (ONS) which is for England and Wales only. Scotland and Northern Ireland have separate statistical offices.

England itself can be divided into 9 regions: The North West, Yorkshire and The Humber, The North East, The West Midlands, The East Midlands, The South West, The East, The South East and London. The UK used to be further divided into counties but these are much less important nowadays than they used to. In fact there is a mix of counties, unitary authorities and boroughs — 152 in total.

The smallest relevant unit for political and statistical purposes is the electoral ward, often simply called ward. Wards elect their own councillors and national statistics are available at a ward level. There were 8570 wards at the time of the 2011 census. For example, the City of Bristol unitary authority contained 35 wards[5], of which the University of Bristol lies in the Cabot ward.

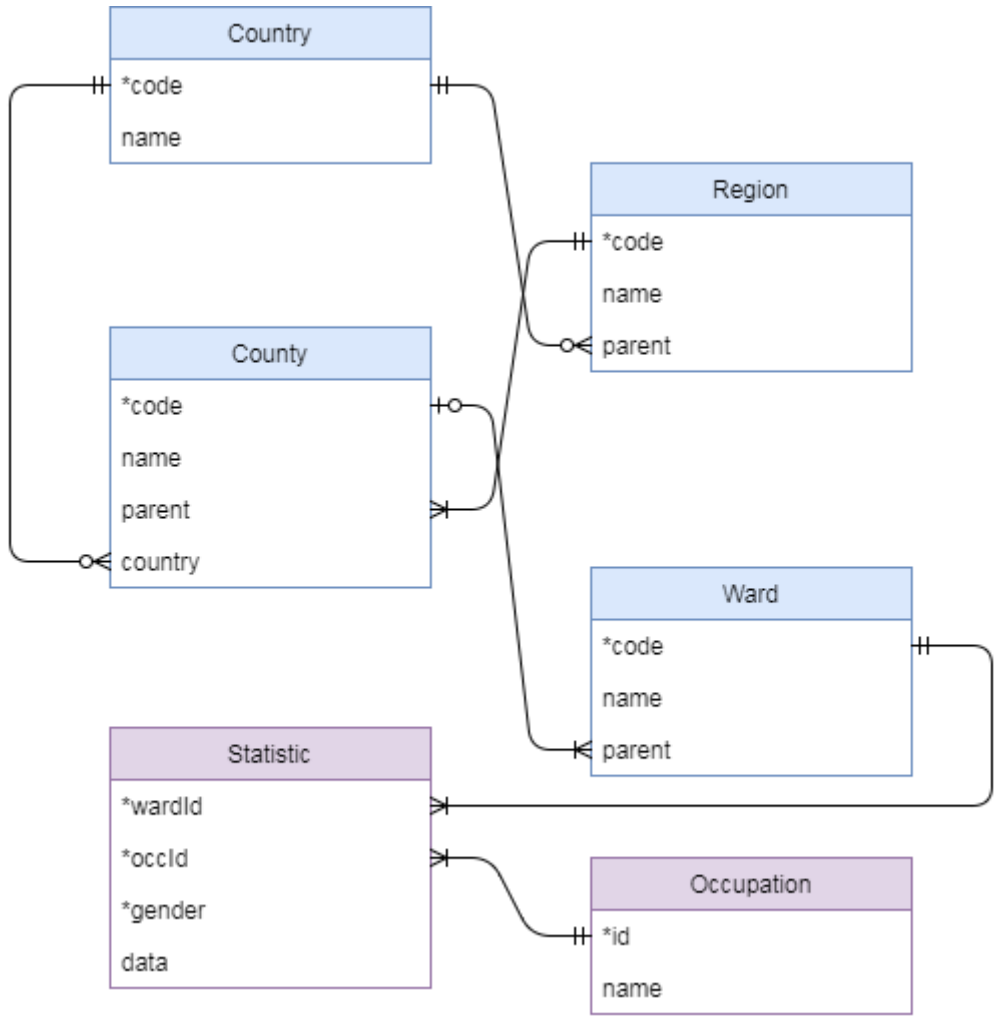
Each statistical unit (ward, county, unitary authority etc.) is assigned a 9-character code by the ONS of the form Xaabbbbbb where X is E for England and W for Wales. The first two digits 'aa' identify the type of unit: 05 is a ward, 06 a unitary authority; in England only E12 is a region, E08 is a borough (metropolitan), E09 is a borough of London and E10 is a county. The last 6 digits identify the individual unit. Finally, the codes for all of England and Wales are E92000001 and W92000004.

Since a picture says more than a page of words, and an interactive map more than a book, you may find the following website useful: http://ukdataexplorer.com/census/.

[3] This may change in the future.
[4] The Republic of Ireland is not part of the United Kingdom.
[5] In 2011 these were: Ashley, Avonmouth, Bedminster, Bishopston, Bishopsworth, Brislington East, Brislington West, Cabot, Clifton, Clifton East, Cotham, Easton, Eastville, Filwood, Frome Vale, Hartcliffe, Henbury, Hengrove, Henleaze, Hill1elds, Hor1eld, Kingsweston, Knowle, Lawrence Hill, Lockleaze, Redland, St George East, St George West, Southmead, Southville, Stockwood, Stoke Bishop, Westbury-on-Trym, Whitchurch Park and Windmill Hill.

Geographical data format

For this exercise I have provided the key geographical data in the tables in the following figure. While the country and region tables hold exactly what they say (though regions only apply to England), the county table holds a mix of counties, unitary authorities and boroughs—but for England only. The ward table finally holds the names of all wards (England and Wales) and for wards in England, the higher-level entity in the county table that they belong to.



In the provided lab database, you can use the command `USE census;` to select the census database. Let's look at the Bristol ward of Cabot (type the text in bold):

```
MariaDB [census]> SELECT * FROM Ward WHERE name = 'Cabot';

+-----------+-------+-----------+
| code      | name  | parent    |
+-----------+-------+-----------+
| E05001979 | Cabot | E06000023 |
+-----------+-------+-----------+
```

We see that the Cabot ward has code E05001979 and that it belongs to E06000023 which is the City of Bristol, a unitary authority. From here we can work our way upwards:

```
MariaDB [census]> SELECT * FROM County WHERE code = 'E06000023';
+-----------+------------------+-----------+-----------+
| code      | name             | parent    | country   |
+-----------+------------------+-----------+-----------+
| E06000023 | Bristol, City of | E12000009 | E92000001 |
+-----------+------------------+-----------+-----------+

MariaDB [census]> SELECT * FROM Region WHERE code = 'E12000009';
+-----------+-----------+------------+
| code      | parent    | name       |
+-----------+-----------+------------+
| E12000009 | E92000001 | South West |
+-----------+-----------+------------+

MariaDB [census]> SELECT * FROM Country WHERE code = 'E92000001';
+-----------+---------+
| code      | name    |
+-----------+---------+
| E92000001 | England |
+-----------+---------+
```

From this information you should be able to find the join strategy for the census database. However, you need to take into account that the wards in Wales do not have any county/region information (the Ward.parent column is NULL for these wards).

KS608: Occupation

Question KS608 on the 2011 census asked all UK citizens in employment and between the ages of 16–74 to classify themselves as one of the following 9 groups. These are also represented in the Occupation table.

id	name
1	Managers, directors and senior officials
2	Professional occupations
3	Associate professional and technical occupations
4	Administrative and secretarial occupations
5	Skilled trades occupations
6	Caring, leisure and other service occupations
7	Sales and customer service occupations
8	Process, plant and machine operatives
9	Elementary occupations

The 2011 census statistics per ward and gender are available in the Statistic table with the following schema. Gender is 1 for female and 0 for male; in the 2011 census these were the only gender options.

```
CREATE TABLE Statistic (
  wardId VARCHAR(10),
  occId INTEGER,
  gender INTEGER, -- 1 = female, 0 = male
  data INTEGER,
  PRIMARY KEY (wardId, occId, gender),
  FOREIGN KEY (wardId) REFERENCES Ward(code),
  FOREIGN KEY (occId) REFERENCES Occupation(id)
);
```

For example, the entry

wardId	occId	gender	data
E05000938	1	1	298

means that in ward E05000938, 298 women said they worked as managers, directors or senior officials.

Tasks

For each of the following questions, place your SQL queries in the appropriate sections of the file census.sql and then upload this file as part of your coursework submission. You must use this template file and keep all lines starting with `--!` exactly as they are and place your answers in the correct parts of the file. You can include empty lines and SQL comments wherever you like as long as you do not start any lines with `--!`.

Your answer must consist of exactly one SQL query per question. Do not include the data table that MariaDB returns when you run your query – we will run your queries ourselves while marking your coursework. Answers without SQL queries or queries with syntax errors that do not run in MariaDB will give 0 marks.

You may use the occupation ids directly rather than joining on to the occupation table (see question 0 for an example), except if the question asks you to list the occupation class name. When a code for a geographical entity is given in the question alongside its name, you may use this code directly too (again see question 0). All questions refer to the census year 2011.

0. How many men work in professional occupations and live in the Cotham ward of Bristol (E05001982)?

Solution:

```
SELECT data FROM Statistic WHERE wardId = 'E05001982'
AND occId = 2 AND gender = 0;
```

1451 men.

Questions 1–5 are fairly straightforward. Questions 6 and 7 are slightly more tricky. Questions 8 and 9 are bonus questions that you should only attempt if you have time to spare.

1. How many women work in sales and customer service occupations and live in the Bishopston ward of Bristol (E05001975)?

2. How many people work in skilled trades occupations and live in the city centre ward of Manchester (E05000697)?

3. For the residents of Portland ward (E05008884), list how many people worked in each class of occupation as a table with two columns `num_people,occ_class`. The second column should contain the name (i.e. "Managers . . . ") of the occupation class.

4. Find the working population, code, ward name and county-level unit name (entry in the county table) of the smallest ward by working population in the dataset. Hint: sort on size and then take the first row to get the minimum.

5. How many wards have at least 1000 working inhabitants?

6. Find the average size of a ward's working population for each region of England. Your output should contain two columns called `name,avg_size`.
   Hint: this one would be easy if you could use two queries and a temporary table. To do it in one query, use an inline view.

7. For all county-level units (CLUs) in the North West region (E12000002), produce a table in the following format

CLU	occupation	gender	N
Cumbria	Associate professional …	female	10054
Lancashire	Caring leisure …	male	10088
Cheshire West …	Sales and customer …	female	10107
Cheshire East	Associate professional …	female	10178

   with the following conditions:

   - "N" is the number of residents of the given occupation class and gender in the given CLU.
   - "gender" is the string 'male' or 'female'.
   - Ignore the "..." in the CLU and occupation columns, just return this from the table directly.
   - The table is sorted by "N" and only values greater or equal to 10000 appear in the table.

Questions 8 and 9 are bonus questions — only attempt these if you have time to spare!

8. Produce a table that lists the following data for each of the 9 regions of England: the region name, the number of men in managerial (id=1) occupations, the number of women in managerial occupations and the proportion of women in managerial occupations (e.g. the number of women divided by the working population). Sort the table by increasing proportion. Hint: find a `SUM(...)` expression that, when run over a whole region's data, only counts the women. There are several ways to do this.

9. This is the same as question 6, except that in addition to the regions you should include two more rows for all of England and the whole dataset. The names in these rows should be (without quotes) "England" and "All".

Note: in past years, we have caught some students who did most of the work on their own but copied their solutions to questions 8 and 9 from other students. Please do not do this – not everyone is expected to complete questions 8 and 9, but if you do hand in solutions to these or any other questions, they must be your own work.

Normalisation exercise

This question is worth 25% of coursework 1. Your solutions must be in a PDF file and the file should be called "normalisation.pdf". Do not include any personally identifying information (name, username etc.) in your PDF file – instead, use your 5-digit candidate number. This enables anonymous marking.

For each of the schemas on the next page, answer the following questions:

1. Identify the candidate key(s) in every table.
2. Identify the key and non-key attributes in every table.
3. Determine which normal forms from (1NF, 2NF, 3NF, BCNF) the schema does or does not satisfy. Give evidence to support your answer.
4. If the schema is not in BCNF, normalise it as far as possible (up to BCNF). This means give a new schema (either as an ER diagram or SQL CREATE TABLE statements) that is a normalised version of the original.

For question 3, you need to show that you have understood the definitions by giving evidence of particular features of the table(s) in question that make or break the normal form. Saying e.g. "this table is in 2NF because (copy of definition of 2NF)" loses you marks on question 3 even if the table is in 2NF. From past exams, I know that some students write this whether or not the table really is in 2NF, so this doesn't show that you've understood the definition.

Although not required, a good starting point is to identify all the non-trivial functional dependencies in the table.

- If a table is e.g. not in 3NF, you need to identify at least one functional dependency in the table that breaks 3NF and explain why.
- If a table is e.g. in 3NF, the argument is a bit harder because you're claiming that all FDs in the table don't violate 3NF. One way to argue this is to identify all FDs; there may be short cuts however.
- Normal forms stack, so if a table is not in 2NF then you can say "therefore it's not in 3NF or BCNF either" and get full marks for the 3NF/BCNF part. In fact, doing this will get you more marks than analysing 3NF again in the same table, since it shows that you have understood that normal forms stack.

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.

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	10.843M	66.8M	yes
Lyon	France	1.609M	66.8M	no
Marseille	France	1.605M	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.)