University of London
Computing and Information Systems/Creative Computing
CO2209 Database systems
Coursework assignment 2 2018-19

Your coursework assignment should be submitted as a single PDF file, using the following file-naming conventions:

YourName_SRN_COxxxxcw#.pdf (e.g. MarkZuckerberg_920000000_CO2209cw2.pdf)

- ***YourName*** *is your full name as it appears on your student record (check your student portal);*
- ***SRN*** *is your Student Reference Number, for example 920000000;*
- ***COXXXX*** *is the course number, for example CO2209; and*
- ***cw#*** *is either cw1 (coursework 1) or cw2 (coursework 2).*

IMPORTANT NOTE: It is important that your submitted assignment is your own individual work and, for the most part, written in your own words. You must provide appropriate in-text citation for both paraphrase and quotation, with a detailed reference section at the end of your assignment. Copying, plagiarism and unaccredited and/or wholesale reproduction of material from books online sources, *etc.* is unacceptable, and will be penalised (see How to avoid plagiarism).

Summary
In **Part A**, we download a real database and examine its structure at various levels, and then run some queries on it. In **Part B**, we look at consequences of Primary Key choice. In **Part C**, we normalize an unnormalized relation. In **Part D**, there are some general knowledge questions.

Background
In coursework assignment 1, we created a 'toy' database, consisting of a small number of tables and just a few rows. Real databases are typically many orders of magnitude larger, in terms of numbers of relations, and the size of each relation. This impacts a crucial aspect of database usage, 'performance' (basically, how fast queries are processed). In this coursework assignment, our practical task will be to download a large database, and use it to learn more about real databases. (Please note that even this database is small compared to many existing ones.)

We are going to download the 'Mondial' database, which carries information about the countries of the world. This is supposedly based on the CIA's **World Fact Book** and a few other data sources. As with many real-world applications*, there may be inconsistencies in its documentation*.

Task 1
Go to this website: http://www.dbis.informatik.uni-goettingen.de/Mondial/ and find the paragraph titled 'Generating the Database under MySQL'.

Click on each of the following links:
http://www.dbis.informatik.uni-goettingen.de/Mondial/OtherDBMSs/mondial-schema-mysql.sql

http://www.dbis.informatik.uni-goettingen.de/Mondial/OtherDBMSs/mondial-inputs-mysql.sql

This will download two files. You do not download the data in the database directly. Instead, you download these two text files, which consist of SQL statements. The SQL statements in the first file (called **mondial-schema-mysql.sql**) create your tables, and then statements in

the second file (called **mondial-inputs-mysql.sql**) will *populate* (add the data to) the tables that the statements in the first file created.

The second file is about 1.5 MBytes in size, and consists of over 20,000 INSERT INTO statements.

To create the Mondial database, you must first download and then execute the SQL commands in these files.

These files may be downloaded and submitted to a 'front end' processor for your database, if you are using one (that is, the statements themselves will be processed).

Or, the contents of each of the files may be copied and pasted into a text editor (you will probably need to add a new file name extension to the files to do this, for example, changing *mondial-inputs-mysql.sql* to *mondial-inputs-mysql.sql.**txt***). Then the SQL statements in those files can be copied and pasted directly to the MySQL command-line processor if you are running MySQL in command-line mode. (Of course, you will *first* do this with the table-creation file, and *then* with the data inputs file.)

Tip: in running MySQL from the command line, if your operating system is Windows, you may want to make the DOS Command Prompt window larger than the default value. Go here to see how to do this:
http://www.isunshare.com/windows-10/change-command-prompt-window-size-in-windows-10.html

There are similar sites on the Web for earlier versions of Windows. You may have to reboot after doing this for the new window size to take effect.

Note: this is a large database, which may take up to half an hour to download. If you are using Linux, and if you encounter a problem with the download, come to the course discussion board and see if your problem has been answered there.

Task 2

From this same website (http://www.dbis.informatik.uni-goettingen.de/Mondial/) download the documents which display the logical structure of the database. There are three, all of which carry the same information, but which show it in increasingly 'physical' ways. The first one – the E/R diagram – shows abstract entity-types and their attributes, and how the entity-types are related to each other. The second one shows actual relations, but illustrates their links visually. The last file just lists relations and their attributes. In theory, all of them hold the same information, but presented in different ways.

An Entity/Relationship diagram:
http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-ER.pdf

A 'Referential Dependency' diagram:
http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-abh.pdf

A Relational Schema:
http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-RS.pdf

These documents should let you see how the data in the separate tables is related, namely which tables hold data relating to the same things. However, as in many real-world applications, *what is on paper and what exists in reality may not match perfectly.*

A note on vocabulary: in reading about relational databases, the following words are usually synonyms, that is, they mean the same thing: ‘relation’, ‘table’ (and sometimes, ‘file’); ‘attribute’, ‘field’, ‘column’; ‘tuple’, ‘row’. The first word is the ‘official’ name. Note that technically, a ‘relation’ is a special kind of ‘table’, that is, ‘table’ is a broader term than ‘relation’. But in general, in casual discussion, they mean the same thing. When dealing with non-technical clients, you will find it easier to talk about ‘tables’ and ‘fields’ and ‘rows’, rather than ‘relations’, ‘attributes’ and ‘tuples’.

The Entity/Relationship diagram is a very simple one, which omits cardinality and participation constraints. Don't worry too much about these documents until you think you have understood the concepts of data dependency, functional dependency, keys and normalization. They will be useful for constructing SQL statements to answer queries which require you to know which tables are linked to which other tables. Be aware that E/R diagrams assume that the world can be conceived in terms of entity-types, which have attributes, and which are related to each other. Sometimes this is only a very rough fit to reality, and we are forced to think of certain things as ‘entity-types’ which we might not naturally think of in this way.

Note that the Relational Schema shown here is only a broad outline schema. It doesn't show datatypes for the attributes, or which attributes are Primary or Foreign Keys, or other constraints.

Important note about the Mondial Database: the value of this database is that it is not a toy one. However, it is definitely out-of-date, and was inaccurate even when first put up on the web. (Remember: all large data sets must be assumed to be ‘dirty’. All documentation of large systems must be assumed to have inconsistencies or errors until proven otherwise.)

Additionally, its designers made at least one poor choice, in my opinion: they have **field names** (attributes, or columns) which are the same as **relation names**. So there is a *relation* called ‘Country’, and in some of the other relations, there is an *attribute* called ‘Country’ as well. The ‘Country’ field (attribute) of these other relations is a Foreign Key for Country.*Code*. That is, it matches an attribute in the Country table which has a different name. It would have been a better idea, in my opinion, to label *all* of these attributes ‘Country-Code’. There are other attribute-relation names which have the same problem. Don’t let these confuse you. If you consult the Referential Dependency Diagram you can see which attributes in which relations are actually ‘the same’ (from the same domain) and are thus possible meaningful natural JOIN links, despite having different names.

Also, note that the Mondial database, as implemented using MySQL, does NOT enforce Foreign Key – ‘referential’ – integrity. That is, it would be possible to have a ‘Country’ field (attribute) in a relation which has no matching ‘Code’ field (attribute) in the relation Country, although the relation Country should list every country.

NOTE: if you have questions about any of the following questions, or need help in completing them, please use the course discussion board.

Coursework assignment 2

Part A: Getting Information about the Database

Background: subject guide, volume 1, pages 106-123**.**

If you ever get a job as a Database Administrator, or have to work with a database that you yourself did not create, you will have to understand your database's structure, *i.e.* what tables are there, what data they hold, and how the data in each table is related to the data in other tables, and, of course, to what extent this data represents reality. We approach this problem using the method of abstraction – first, we try to get an overall picture of the database – the kinds of things it's holding data on, and how they are related – and then we get closer and closer to the actual data.

Getting information about the Database at the Semantic/Conceptual Level
When we use the words 'semantic', or 'conceptual' or 'logical' in connection with data, we are referring to the *information* the data is supposed to convey, abstracted away from *how* it does this, from its actual physical implementation on paper, or magnetic disc, or any other physical medium.

In the Mondial database, we are given three levels of non-physical descriptions of the data: an E/R diagram – sometimes called the 'conceptual' or 'semantic' level; a Referential Dependency diagram, and a Relational Schema, proceeding from a higher level of abstraction to a lower one.

An E/R diagram is at a higher level of abstraction than a Referential Dependency diagram, which in turn is more abstract than a Relational Schema.

Note that at each level, there is always a choice about how much information to include – the examples here are fairly minimal – thus the E/R diagram doesn't have cardinality and participation constraints, and the relational schema does not include data types (see page 106 of **volume 1** of the **subject guide** for further discussion of the levels of information design).

Question 1
Look at the **E/R Diagram**, and in particular at the Entity-types Country, Province, and City. To answer some of these questions, you will need to do **question 3** of this coursework assignment, so you may want to skip them and return to them after you have completed **question 3**.

(a) Review page 109 of the **subject guide, volume 1**. Of the three entity-types mentioned above (the underscored names), which are 'weak' entity types?
(b) What is the relationship of City to Country?
(c) According to this E/R diagram, will the database list all of the cities of a given country? If not, what cities will it list? In other words, to be included in this database, what relationship to a country must a city have, according to the E/R diagram? (We will assume that there is an agreed definition of 'city' which has determined what names to include in the database.)

[3 marks]

Question 2
Look at the **Referential Dependency Diagram** ('Mondial.abh.pdf')**.** Answer the following:

- **(a)** How many tables are shown in this diagram?
- **(b)** How many tables are listed when you run the SHOW TABLES command in **question 3** of this coursework?
- **(c)** According to the Referential Dependency Diagram, what are the attributes (fields) of the POLITICS table?
- **(d)** What are the attributes (fields) which are listed when you run the DESCRIBE POLITICS command in the next section of this coursework?

Look at the tables called ISMEMBER, COUNTRY, ORGANIZATION, and CITY in the **Referential Dependency Diagram.**

- **(e)** The field (attribute) 'country' in ISMEMBER has a link drawn from it, to the field 'code' in COUNTRY. What does this mean?
- **(f)** The tables CITY and COUNTRY both have fields called 'population', but there is not a link drawn between them. Is this an oversight, or can you think of a reason that there is no link drawn? [HINT: what does it mean for two fields to be 'linked'? Note that some fields are linked even if they do not have the same name.]
- **(g)** Looking at the tables ORGANIZATION and ISMEMBER, we can see that they both have a field called 'Country', although these fields are not linked to each other. What is the field 'Country' telling us in the table ORGANIZATION? What is it telling us in the table ISMEMBER? In other words, what role does this field play in each table?

[7 marks]

What to submit: answers to **Part A, Question 1 (a)-(c)** and **Question 2 (a)-(g).** Please follow the numbering conventions in this coursework assignment, and start each answer on a new line. Thus, your answers should look like this:

Part A
Question 1
(a) Your answer…
(b) Your answer …

Question 2
(a) Your answer…
(b) Your answer …

Getting information about the Database at the Logical Level
Although documentation describing the database at the conceptual and logical level can be useful, the only ultimate 'documentation' is the data itself. Here you will get the actual relational schema, along with a bit of purely 'physical' information (namely, information about how the tables are indexed).

Compiling a schema of the tables you have downloaded
This compilation will provide basic information about the database tables you have set up. Unlike the documentation in the previous section, this will show you what the database 'intension' actually is.

Combined with the Referential Dependency diagram, the E/R diagram, and the rather general Relational Schema you have already downloaded, you would have the materials you need to start to understand your database's structure. The document called Mondial.RS.pdf is a Relational Schema, but you can generate one with more information using MySQL itself.

You don't actually have to write anything here, but rather, just copy in the results of running some commands. You will get some basic information about all of the tables, and then some detailed information.

To do this, you will need three SQL commands:

SHOW TABLES; *and*
DESCRIBE <tablename>; *and*
SELECT COUNT(*) FROM <tablename>.

SHOW TABLES just gives us the **name** of every table (relation) in our database.
The other two commands let us get data about a particular table: its structure, and how many rows (tuples) it has.

Note that for '<tablename>' you will need to substitute the name of one of the tables listed by **SHOW TABLES.** We will look at just a few of them. Use of a word processor or a simple text editor can make this a very quick operation if you are working with MySQL directly from the DOS prompt. Note that for earlier versions of Windows, to paste into the Command Prompt, you may need to right-click – CTRL-V may not work.

It will be useful to be able to output what you see on the screen to an output file; you can use the command 'TEE' to do this, as follows (user input in bold):

mysql> **TEE** D: OutputLog.txt – whatever shows on the screen is also copied to the file OutputLog.txt which I have placed on my D: disc in this example, but which can be located anywhere you like, provided you give the full Path name.

mysql> **SHOW TABLES;** – information about the tables will be sent to OutputLog.txt as well as being shown on the screen;

mysql> **DESCRIBE COUNTRY**;
mysql>**SELECT COUNT(*) FROM COUNTRY;**
mysql> **SHOW INDEX FROM COUNTRY;**
- and so on ... for the following relations in this database;
- (In other words, substitute the names below, one at a time, for the name COUNTRY in the commands above.)

- CITY
- ECONOMY

- ISMEMBER
- LANGUAGE
- ORGANIZATION
- POLITICS
- POPULATION
- RELIGION

mysql> **NOTEE;** – turns it off;

Question 3
Include the requested information for the tables listed above, plus the size of each of these tables, which you can find out by executing the SQL statement below.

To see the size in megabytes of each of your tables, do this:

```
SELECT
    table_schema as `Database`,
    table_name AS `Table`,
    round(((data_length + index_length) / 1024 / 1024), 2) `Size in MB`
FROM information_schema.TABLES
WHERE table_schema = 'mondial'
ORDER BY (data_length + index_length) DESC ;
```

What to submit: answer to **Part A, Question 3** – the results of running the commands above for the tables named above.

[5 marks]

Question 4
Answer the following questions:

(a) What is the total size of the Mondial database in megabytes?
(b) What are the largest, and the smallest, relations in the Mondial database in terms of total bytes?
(c) For any two relations (in any database) is it the case that the relation with the largest cardinality must be the largest in terms of total bytes of data? If not, can you give an example from this database which shows this?
(d) For any two relations, is it the case that the relation with the largest degree (number of columns) must be the largest in terms of total bytes of data? If not, can you give an example from this database which shows this?
(e) If, given two relations, one is larger than the other in terms of both cardinality and degree, is it *necessarily* larger than the second one, in terms of total bytes of data? If *not*, can you give an example from this database which shows this?

What to submit: answers to **Part A, Question 4 (a)-(e)** Please follow the numbering conventions in this coursework assignment, and start each answer on a new line. Thus, your answers should look like this:

(a) ***Your answer…***
(b) ***Your answer …***

[5 marks]

Queries on the Mondial database

A note about SQL: SQL's tables do not conform completely to the definition of relations. In particular, the tables which result from a query can have duplicate tuples (rows), which in most cases is not what we want, and violates the definition of 'relation'. To avoid this, always use the **DISTINCT** keyword, as in **SELECT DISTINCT …**

Show not just your query, but the data set that results. Use 'LIMIT 5' at the end of your query if your query returns more than five results. Note that queries which involve COUNTs and SUMs, *etc,* will return just one value. Look at the example shown in **coursework assignment 1** to see how to submit your answers: each answer should include **(1)** the original natural language query, **(2)** the SQL commands to answer it, and **(3)** the data set that results (limited to the first five tuples if there are more than five in the result).

Question 5

- **(a)** What is the query that will list the name, population, and population density of each country? (Note: 'Population density' can be defined as the ratio of Population to Area.)
- **(b)** What is the query that will list the religions found in Japan?
- **(c)** What is the query that will list the names of the countries which have at least one religion in common with Japan?
- **(d)** What is the query that will list the total GDP for all countries added together? (This will be a single value.)
- **(e)** What is the query that will list the name of the country with the lowest rate of inflation? (Note that there may be 'ties', in which case, list the first five.)
- **(f)** What is the query that will list the names of countries which are members of Interpol? (Remember to use **LIMIT 5**.)
- **(g)** What is the query that will list the names of countries which are **not** members of Interpol? (Hint: this is going to require a *set difference*.) See subject guide, volume 1, page 90.
- **(h)** Is there any difference in these two queries, besides the fact that the second one is longer? If so, what is the difference?

```
SELECT Name, GDP, Agriculture, Service, Industry, Area, Population
  FROM Country JOIN  Economy
  WHERE  name = 'Japan'
  LIMIT 1;
```

```
SELECT Name, GDP, Agriculture, Service, Industry, Area, Population
FROM Country JOIN  Economy
WHERE  name = 'Japan' AND country.code=economy.country
 LIMIT 1;
```

- **(i)** What is the query that will list the world's languages, and for each one, the total number of countries where each is spoken? (Hint: use **GROUP BY**).
- **(j)** What is the query that will list the world's languages which are spoken in at least 5 countries, and for each one, the total number of countries where they are spoken? (HINT: use **GROUP BY** and **HAVING**).

What to submit: answers to **Part A, Question 5 (a)-(j)**, in the requested format.

[4 x 10 marks]

Part B: Choosing the Primary Key

This exercise is about the consequences of choosing the right attributes to make up the Primary Key, or not doing so. You may wish to read pages 59-60 of the **subject guide, volume 1**, before doing it. Note that the 'Primary Key' is a particular 'Candidate Key' that we choose. There will often be only one Candidate Key in a relation, in which case it is automatically the Primary Key.

The following table records the results for would-be actors who are undergoing a preliminary screening audition for a particular role in a play. Any actor who is turned down for a role, in the preliminary screening, is not allowed to re-audition for that role again. (Everyone who passes will be re-auditioned later along with others who passed the preliminary screening, and that data will be put in a separate table.)

Actor	Role	AuditionDate	Result
Aaryan Chaudary	Julius Caesar	2018-09-13	reject
Aaryan Chaudary	Brutus	2018-09-14	pass
Barry Evans	Julius Caesar	2018-09-13	pass
Isaac Biko	Cassius	2018-09-13	pass
Isaac Biko	Julius Caesar	2018-09-13	pass

The primary key of this table is **Actor** + **Role**.

Question 1
What bad consequences could follow if, in creating the table – before we added the data – we defined the primary key of the table as:

(a) Actor alone?
(b) Role alone?
(c) AuditionDate Alone?
(d) Actor + Role + AuditionDate?
(e) Actor + Role + AuditionDate + Result?

[10 marks]

Question 2
Suppose it is decided to allow actors to re-audition for roles for which they were initially rejected, at a later date. Would we need to change the definition of the primary key? If we didn't change it, what problem might arise? If we changed it, what would the new key be? Explain your answer.

[5 marks]

What to submit: answers to **Part B, Question 1 (a)-(e)** and **Part B, Question 2;** please start each answer on a new line.

Part C: Normalizing an Unnormalized Relation

You may wish to read the **subject guide volume 1**, pages 131-141, and to re-read **Appendix IV** of coursework assignment 1 before attempting this question.

The following table holds information about the annually-recorded weight of particular prize sheep, and about the veterinarians (vets) who weigh them. Each sheep belongs to a particular owner. No sheep is owned by more than one owner. Each sheep's birthdate and current owner is recorded. Every year, a veterinarian weighs each prize sheep and records its weight. The date of the weighing, and the ID number of the vet, plus the mobile phone number of the vet, is recorded. Vets have only one mobile phone number. The Primary Key of the table recording this information is **SheepID+WeighingDate**. The table has not been normalized beyond First Normal Form. That is, there are no 'repeating groups', but there may be Partial and Transitive Dependencies.

SheepID	Owner	Birthdate	WeighingDate	Vet	Weight	VetPhoneNum
K3922	McNab013	2013-05-12	2013-08-14	M330	22	7633088852
K3922	McNab013	2013-05-12	2014-06-02	S929	34	7609865463
K3922	McNab013	2013-05-12	2015-08-02	M330	43	7633088852
K3922	McNab013	2013-05-12	2016-07-30	P301	53	7682907965
K3922	McNab013	2013-05-12	2017-08-12	P301	52	7682907965
K3922	McNab013	2013-05-12	2018-07-22	S929	51	7609865463
T8832	McNab013	2012-03-26	2012-08-14	K339	19	7602907550
T8832	McNab013	2012-03-26	2013-09-01	S929	26	7609865463
T8832	McNab013	2012-03-26	2014-08-15	K339	32	7602907550
T8832	McNab013	2012-03-26	2015-07-28	K339	40	7602907550
T8832	McNab013	2012-03-26	2016-08-11	T975	42	7646746741
P9742	Smith002	2014-05-10	2014-09-11	K339	14	7602907550
P9742	Smith002	2014-05-10	2015-08-10	S300	25	7629920821
P9742	Smith002	2014-05-10	2016-08-29	K339	35	7602907550
P9742	Smith002	2014-05-10	2017-09-01	S929	44	7609865463
P9742	Smith002	2014-05-10	2018-08-22	S929	51	7609865463
M3110	Smith002	2015-05-12	2015-09-12	S300	15	7629920821
M3110	Smith002	2015-05-12	2016-09-10	S300	26	7629920821
M3110	Smith002	2015-05-12	2017-08-17	K339	34	7602907550

Question 1
Identify the Functional Dependencies in this table. Your answer should conform to the following format: If, taken together, attributes A and B determine C, show it this way: $A + B \rightarrow C$

[5 marks]

Question 2
This table is susceptible to update, deletion, and insertion anomalies. Examine the following table, which lists ten possible problems. Some of these problems are examples of normalization anomalies, and some are not.

Problem	Update Anomaly	Deletion Anomaly	Insertion Anomaly	None of these.
a. Instead of entering '2014-08-15' for one of the weighing dates, '2104-08-15' could be entered instead.				
b. We cannot insert information about a new vet (such as their mobile phone number) until they have weighed a sheep.				
c. If we delete all the tuples about a particular sheep, we lose all the information about that sheep.				
d. If we delete all the tuples for sheep K3922, we lose information about Vet M330's mobile phone number.				
e. We cannot enter ownership and birthdate information about a sheep until it has been weighed.				
f. A Vet could record a sheep's weight incorrectly.				
g. If a Vet changes their mobile phone number, we could record the change in some of the tuples with thier ID, and not in others.				
h. If a Vet changes their mobile phone number, we could record the new number incorrectly.				
i. If we delete information about a Vet, we lose their mobile phone number.				
j. If sheep is sold to a new owner, we could change some of its Owner values but not others.				

[10 marks]

Question 3

Split the original table into tables in BCNF, specifying the Primary Key of each table, and showing the first and last tuples for each table (following the order of tuples in the original table).

[5 marks]

What to submit: answers to **Part C, Question 1,** and **Part C, Question 2 (a)-(j),** and **Part C, Question 3;** please start each answer on a new line.

Part D: General Knowledge

Write brief definitions (four or five sentences) for the following terms, as they are used in the database context:

(a) Deadlock
(b) Data Dictionary
(c) Query Optimization
(d) View
(e) SQL injection

What to submit: answers to **Part D, (a)-(e)**; please start each answer on a new line.

[5 marks]

[Total: 100 marks]

[END OF COURSEWORK ASSIGNMENT 2]