University of London International Programmes

Computing and Information Systems/Creative Computing

CO2209 Database systems

Coursework assignment 2 2017–18

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 in 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 coursework 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).

Background
In the Coursework assignment 1, you created a 'toy' database, consisting of fewer than 10 tables and a few dozen rows. Real-world 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, your practical task will be to download such a database, and use it to learn more about real databases. You are going to download the 'Mondial' database, which carries information about the countries of the world. This is supposed to be based on the CIA's World Fact Book among other sources.
Go to this website:
http://www.dbis.informatik.uni-goettingen.de/Mondial/

Find the paragraph entitled 'Generating the Database under MySQL' and 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

You do not download the data in the database base directly, but rather you download statements in the first file that will create the tables (called mondial-schema-mysql.sql); and then statements in the second file (called mondial-inputs-mysql.sql), which will populate them.

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

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).

Alternatively, the whole files, which are displayed when you click their links, 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, by changing mondial-inputs-mysql.sql to mondial-inputs-mysql.sql.txt). Then the 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 schema creation file, and **then** with the data inputs file.)

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, use the forum to see if your problem has been answered there.

From this same website:
http://www.dbis.informatik.uni-goettingen.de/Mondial/
Download the documents which display the structure of the database; there are three, all of which carry the same information, but show it in different ways:

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

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

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

These documents let us see how the data in the separate tables is related; namely, which tables hold data relating to the same things. Note that the Relational Schema shown here is only a broad outline schema. It does not show datatypes for the attributes, or which attributes are Primary or Foreign Keys, or other constraints.

The Entity/Relationship diagram is a very simple one, which omits cardinality and participation constraints. Do not 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.

Important notes 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'.)

- Additionally(this is just an opinion), its designers made at least one poor choice, : 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'. The 'Country' field of these other relations is a Foreign Key for

Country.Code. It would have been a better idea to label these attributes ‘Country-Code’. We will look at a way to improve this.

- The Mondial database as implemented using MySQL does NOT enforce Foreign Key integrity. (That is, it would be possible to have a ‘Country’ field in a relation which has no matching ‘Code’ field in the relation Country.)

Coursework assignment 2

NOTE: If you have questions about any of these tasks, or need help in completing them, please use the CO2209 course discussion forum. Please take care not to post anything that will form part of an answer.

A. The Mondial database

A1. Compiling a description of the tables you have downloaded

This compilation will provide basic information about the database tables you have set up. If you work as a Database Administrator, or have to work with a database that you yourself did not create, you will have to do something like this first. Combined with the Referential Dependency diagram, E/R diagram, and Relational Schemas you have already downloaded, you would have the materials you need to start to understand your database's structure.

You do not actually have to write anything here, but rather, just copy in the results of running some commands:

To do this, you will need the SQL commands

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

Note that for '<tablename>' you will need to substitute the names of each of the tables listed by **SHOW TABLES**. 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 – as 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:

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.
mysql> **SHOW TABLES**; – information about the tables will be sent to OutputLog.txt as well as being shown on the screen;
mysql> **DESCRIBE BORDERS**;
mysql>**SELECT COUNT(*) FROM BORDERS**;
mysql> **SHOW INDEX FROM BORDERS;**
- and so on... for the first **five** relations in this database (COUNTRY to DESERT);

mysql> **notee**; – turns it off;

A1(a) Include the requested information for the first five tables, plus the size of each of these tables, which you can find out by executing the commands 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;
```

A1(b) Answer the following questions.

(1) What is the total size of the Mondial database?

(2) What are the largest, and the smallest, relations in the Mondial database in terms of total bytes?

(3) 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?

(4) 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?

(5) 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?

Please start your answer for A1(b) on a new page.

[2 hours, 10 marks]

A2. Queries on the Mondial database

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 dataset that results. Use 'LIMIT 10' at the end of your query if your query returns more than 10 results.

Please start your answers for A2 on a new page.

A2(a) What is the query that will list the area of Iran?

A2(b) What is the query that will list the names of the countries which have a greater area than Iran?

A2(c) 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.)

A2(d) What is the query that will list the total GDP for all countries?

A2(e) What is the query that will list the country with the highest inflation?

A2(f) What is the query that will list the (code for) names of countries which belong to the World Health Organization (WHO)? (Remember to use **LIMIT 10**.)

A2(g) What is the query that will list the (code for) names of countries which do **not** belong to WHO? (Hint: this is going to require a *set difference*. See **subject guide**, Volume 1, Page 90.)

A2(h) Someone wants to find out the economic data about Iran, and doesn't remember the code for that country. So, they do a join on the 'country' table, which allows them to use the country's name directly. This is the query they put to the system, and no error was reported.

> **SELECT** * **FROM** economy, country **WHERE** country.name = 'Iran';

Even though the first query was executed without any problem, a friend suggested that they should have entered the following query (which is longer):

> **SELECT** * **FROM** economy, country **WHERE** country.name = Iran' **AND**
> country.code=economy.country;

Is there a difference between these two queries, besides the fact that the second one is longer? If there is a difference, what is it?

A2(i) What is the query that will list the world religions, and for each one, the total number of countries where each is represented? (Hint: use **GROUP BY**).

A2(j) What is the query that will list the world religions that are present in at least 12 countries, and for each one, the total number of countries where it is represented?

[10 hours, 4 marks each. Total: 40 marks]

B. Database design

B1. Choosing the primary key

Aaryan, Bob, Carol and Dawud formed their own database design consultancy. For their first job, a small 'crammer' college hired them to design a database for the following situation:

> Students take mock examinations in subjects, and receive a mark. If they want to try to improve their grade, they can take the examination again. The college wants to keep a record of who has taken which examinations, on what date, and what mark they received. All four designers agreed that the relation which would hold this information should have the attributes shown below, but they couldn't agree on what the primary key of the relation should be.

Student-ID	Subject	Date	Mark

Aaryan proposed that Student-ID be designated as the primary key.
Bob proposed that Student-ID + Subject be designated as the primary key.
Carol proposed that Student-ID+ Subject + Date be designated as the primary key.
Dawud proposed that Student-ID+ Subject + Date + Grade be designated as the primary key.

Who was right? Or were they all wrong? Or does it make any difference? Motivate your answer by giving **at least three** different examples of what might go wrong during data entry if the incorrect choice of key is made when creating the relation.

Please start your answers for B1 on a new page.

[1 hour, 5 marks]

B2. Consider the following table, which records the books and articles written by lecturers. (Lecturers are identified by Employee numbers, books by ISBNs, and articles by Serial Item and Contribution Identifiers (SICI) Note that the lecturer whose employee number is P22234 has written one book and two articles, and the lecturer whose employee number is K39423 has written two books and one article.

LECTURER-DETAILS

Lecturer	Book	Article
P22234	1 84195 525 6	1046-8188(199501)13:1<69:FTTHBI>2.0.TX;2-4
P22234	.null.	0002-8231(199601)47:1<23:TDOMII>2.0.TX;2-2
K39423	0 86104 068 6	0095-4403(199502/03)21:3<12:WATIIB>2.0.TX;2-J
K39423	978 1 84489 416 1	.null.

Although this relation does not violate the rule 'let every determinant be a candidate key', it demonstrates a very poor design. Explain why, and suggest an alternative way to represent the information held by it.

Please start your answers for B2 on a new page.

[1 hour, 5 marks]

B3. Part of a database will be a list of a country's Provinces, and Towns within each Province. The database will hold information as shown in the sample below. The Province of Beervaria has three towns: Saint Gaul, Trumpville, Murkle City; the Province of New Wales has the towns of Malwaria, and Putintown; the Province of Erehwon has the towns of Nada, Rien, Nullity, and Infinitesimal. The country is growing, and new Provinces may be created and new towns founded in the future.

Two designs have been proposed. (Primary keys are underscored.)

Province	**Towns**
Beervaria	"Saint Gaul, Trumpville, Murkle City"
New Wales	"Malwaria, Putintown"
Erehwon	"Nada, Rien, Nullity, Infinitesimal"

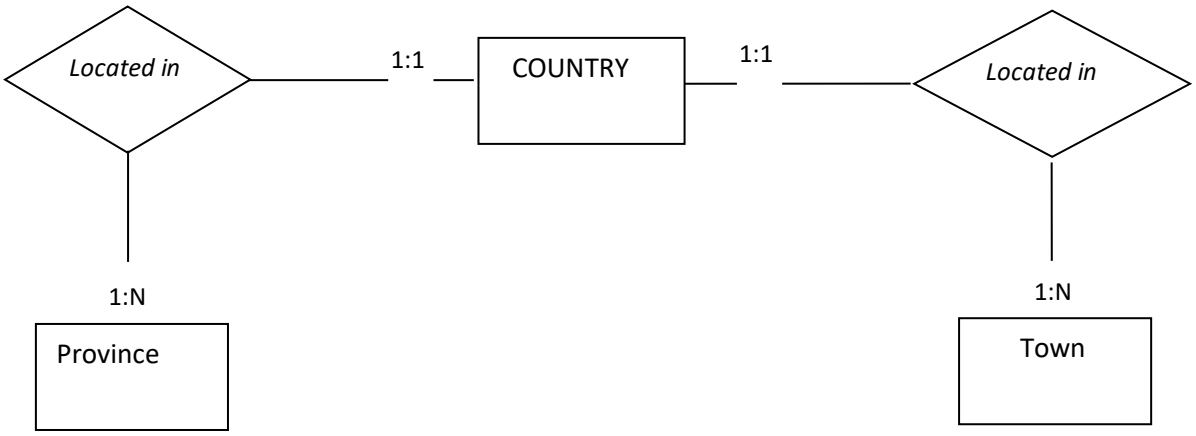
Province	**Town1**	**Town2**	**Town3**	**Town4**
Beervaria	Saint Gaul	Trumpville	Murkle City	NULL
New Wales	Malwaria	Putintown	NULL	NULL
Erehwon	Nada	Rien	Nullity	Infinitesimal

Critique these designs from the point of view of the ease of: **(1)** searching for the name of a Province, given the name of a town; **(2)** searching for the names of all Towns, given a Province; and **(3)** adding a new Town to a Province.

Please start your answers for B3 on a new page.

[1 hour, 5 marks]

B4. Suppose we wanted to expand the scope of our application in **B3**, and allow more than one Country to be represented in the database, each Country having several Provinces. Assume we have been given the following Entity/Relationship diagram for the new situation.



What is wrong with the way this diagram tries to represent the relationships described above? In database modelling, what sort of error is this called? Draw up an alternative E/R diagram that overcomes these defects.

Please start your answers for B4 on a new page.

[1 hour, 5 marks]

C. Getting help/Finding out more/advanced topics

The rapid advance of computer technology, especially the internet, has profoundly affected the world of databases. Whereas someone graduating in 1987 did not see much substantial change in the database world for the next 10 years, you will almost certainly see deep changes over the next decade. If your job involves significant use of databases, you will need to keep up with developments in this field. In addition, you may well want to extend your knowledge of databases beyond the fundamentals provided in this course.

There are many online resources which can help you keep up with developments in the field; and deepen your knowledge of existing technology. This section of the coursework assignment introduces you to some of them.

C1. The null controversy

Search the internet for articles which argue that NULL values should **not** be used in relational databases, and write a brief summary – which may be as short as five sentences – of some of the arguments against their use.

Be sure to consult this site: http://www.dbdebunk.com/2017/04/null-value-is-contradiction-in-terms.html#more

Please start your answer for C1 on a new page.

[3–6 hours, 15 marks]

C2 Go to the internet and find video presentations on the subject of 'NoSQL databases'. Write a short summary of the information they present, in no more than 250 words. List the URL of each video you watch, and its running time: their total (that is, combined) running times should not be less than 45 minutes. (Thus, you can watch one video which runs for an hour, or four which run for twelve minutes each, just so long as the total adds up to at least 45 minutes.)

Please start your answer for C2 on a new page.

[3–6 hours, 15 marks]

[Total: 100 marks]

[END OF COURSEWORK ASSIGNMENT 2]