UNIVERSITY OF LONDON | INTERNATIONAL PROGRAMMES

Student Name:	Herdy Cen
Date Submitted:	13th March 2019
Degree Title:	Computing and Information Systems
Subject:	Database Systems
Local Institution:	Singapore Institute of Management
Student ID:	170280563

Table of Contents

Part A

Question 1

(a) According to the E/R diagram, Province and City are both weak entities out of the three entities.

(b) The relationship of City to Country is Capital.

(c) No, the database will not list all of the cities within a single country. It will list the cities which are the capital city in each country, located nearby the airports and the headquarters of existing organizations in each country.

Question 2

(a) 43

(b) 33

(c) - country

- dependent
- independence
- wasdependent
- government

(d) - country

- independence
- dependent
- government

(e) The link connecting field ‘country’ in ISMEMBER with ‘code’ in COUNTRY which is drawn from ISMEMBER pointing to COUNTRY implies that ‘country’ in ISMEMBER is functionally dependent on ‘code’ in COUNTRY.

(f) This means that there is no relationship connection between both ‘population’ even though it has the same field name either on table CITY or COUNTRY. It can be implied that each ‘population’ in each table belong to the table itself and has no connection with the other table. One shows the population in a city while the other one displays the population in a country. It is the same case as ‘name’ attribute in ORGANIZATION and COUNTRY in which the attribute ‘name’ are both identical but no link is drawn.

(g) The field ‘country ‘ in the table ORGANIZATION represents the origin of the organization in a random order of country name. Meanwhile, field ‘country’ in ISMEMBER is connected to ‘code’ in COUNTRY and according to the database, field ‘country’ in ISMEMBERS represents the country code of a country within a certain organization.

Question 3

```
DESCRIBE city;

+-------------+---------------+-------+------+----------+-------+
| Field       | Type          | Null  | Key  | Default  | Extra |
+-------------+---------------+-------+------+----------+-------+
| Name        | varchar(35)   | NO    | PRI  | NULL     |       |
| Country     | varchar(4)    | NO    | PRI  | NULL     |       |
| Province    | varchar(35)   | NO    | PRI  | NULL     |       |
| Population  | int(11)       | YES   |      | NULL     |       |
| Longitude   | float         | YES   |      | NULL     |       |
| Latitude    | float         | YES   |      | NULL     |       |
+-------------+---------------+-------+-----+----------+-------+
6 rows in set (0.01 sec)
```

```
SELECT count(*)
FROM city;

+----------+
| count(*) |
+----------+
|     3111 |
+----------+
1 row in set (0.02 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM city;
```

```
+-----------+----------------+----------------+------------------+-------------------+
| Table      | Non_unique  | Key_name    | Seq_in_index  | Column_name  |
+-----------+----------------+----------------+------------------+-------------------+
|  city       |                  0 |  PRIMARY   |                   1 | Name              |
|  city       |                  0 |  PRIMARY   |                   2 | Country          |
|  city       |                  0 |  PRIMARY   |                   3 | Province         |
+-----------+----------------+-----------------+-----------------+-------------------+
```

```
+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A            |            2932 |        NULL |     NULL |          |
|  A            |            2984 |        NULL |     NULL |          |
|  A            |            2997 |        NULL |     NULL |          |
+-------------+---------------+--------------+------------+--------+
```

```
+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE     |              |                         | YES       | NULL         |
|  BTREE     |              |                         | YES       | NULL         |
|  BTREE     |              |                         | YES       | NULL         |
+--------------+------------+---------------------+-----------+--------------+
3 rows in set (0.21 sec)
```

```
DESCRIBE economy;

+--------------+-------------+-------+------+----------+-------+
| Field        | Type        | Null  | Key  | Default  | Extra |
+--------------+-------------+-------+------+----------+-------+
| Country      | varchar(4) | NO    | PRI  | NULL     |       |
| GDP          | float       | YES  |      | NULL     |       |
| Agriculture  | float       | YES  |      | NULL     |       |
| Service      | float       | YES  |      | NULL     |       |
| Industry     | float       | YES  |      | NULL     |       |
| Inflation    | float       | YES  |      | NULL     |       |
+--------------+-------------+-------+-------+---------+--------+
6 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM economy;

+----------+
| count(*) |
+----------+
|      238 |
+----------+
1 row in set (0.00 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM economy;

+--------------+----------------+----------------+------------------+-------------------+
| Table        | Non_unique  | Key_name    | Seq_in_index  | Column_name  |
+--------------+----------------+----------------+------------------+-------------------+
|  economy    |                    0 |  PRIMARY   |                    1 | Country           |
+--------------+----------------+-----------------+-----------------+-------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A             |              238 |        NULL |     NULL |          |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE      |                |                           | YES       | NULL         |
+--------------+------------+---------------------+-----------+--------------+
1 row in set (0.06 sec)
```

DESCRIBE ismember;

```
+-----------------+---------------+-------+------+-----------+--------+
| Field           | Type          | Null  | Key  | Default   | Extra  |
+-----------------+---------------+-------+------+-----------+--------+
| Country         | varchar(4)    | NO    | PRI  | NULL      |        |
| Organization    | varchar(12)   | NO    | PRI  | NULL      |        |
| Type            | varchar(35)   | YES   |      | member    |        |
+-----------------+---------------+-------+------+-----------+--------+
3 rows in set (0.01 sec)
```

SELECT count(*)
FROM ismember;

```
+----------+
| count(*) |
+----------+
|     8008 |
+----------+
1 row in set (0.04 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM ismember;
```

```
+---------------+----------------+----------------+------------------+--------------------+
| Table         | Non_unique  | Key_name    | Seq_in_index  | Column_name    |
+---------------+----------------+----------------+------------------+--------------------+
|  ismember     |              0 |  PRIMARY   |                 1 | Country           |
|  ismember     |              0 |  PRIMARY   |                 2 | Organization     |
+---------------+----------------+-----------------+-----------------+--------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A           |           225 |       NULL |    NULL |         |
|  A           |          7772 |       NULL |    NULL |         |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE     |             |                      | YES      | NULL        |
|  BTREE     |             |                      | YES      | NULL        |
+--------------+------------+----------------------+-----------+--------------+
2 rows in set (0.00 sec)
```

```
DESCRIBE language;

+-------------+---------------+-------+------+----------+-------+
| Field       | Type          | Null  | Key  | Default  | Extra |
+-------------+---------------+-------+------+----------+-------+
| Country     | varchar(4)    | NO    | PRI  | NULL     |       |
| Name        | varchar(50)   | NO    | PRI  | NULL     |       |
| Percentage  | float         | YES   |      | NULL     |       |
+-------------+---------------+-------+-------+----------+------+
3 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM language;

+----------+
| count(*) |
+----------+
|      144 |
+----------+
1 row in set (0.02 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM language;
```

```
+---------------+----------------+----------------+------------------+--------------------+
| Table         | Non_unique  | Key_name    | Seq_in_index  | Column_name    |
+---------------+----------------+----------------+------------------+--------------------+
|  language     |                  0 |  PRIMARY   |                    1 | Name               |
|  language     |                  0 |  PRIMARY   |                    2 | Country            |
+---------------+----------------+-----------------+-----------------+--------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed      | Null    |
+-------------+---------------+--------------+------------+--------+
|  A            |                74 |        NULL |    NULL |          |
|  A            |              144 |        NULL |    NULL |          |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE     |               |                         | YES       | NULL         |
|  BTREE     |               |                         | YES       | NULL         |
+--------------+------------+----------------------+-----------+--------------+
2 rows in set (0.04 sec)
```

```
DESCRIBE organization;

+-----------------+---------------+-------+------+----------+-------+
| Field           | Type          | Null  | Key | Default  | Extra |
+-----------------+---------------+-------+------+----------+-------+
| Abbreviation    | varchar(12)   | NO    | PRI | NULL     |       |
| Name            | varchar(80)   | NO    | UNI | NULL     |       |
| City            | varchar(35)   | YES   |     | NULL     |       |
| Country         | varchar(4)    | YES   |     | NULL     |       |
| Province        | varchar(35)   | YES   |     | NULL     |       |
| Established     | date          | YES   |     | NULL     |       |
+-----------------+---------------+------  +------+----------+-------+
6 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM organization;

+----------+
| count(*) |
+----------+
|      153 |
+----------+
1 row in set (0.00 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM organization;
```

```
+------------------+----------------+----------------------+------------------+--------------------+
| Table            | Non_unique  | Key_name            | Seq_in_index  | Column_name    |
+------------------+----------------+----------------------+------------------+--------------------+
|  organization    |              0 |  PRIMARY            |                 1 | Abbreviation      |
|  organization    |              0 |  OrgNameUnique   |                 1 | Name                 |
+------------------+----------------+----------------------+-----------------+--------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A            |            153 |        NULL |     NULL |          |
|  A            |            153 |        NULL |     NULL |          |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE     |              |                         | YES      | NULL        |
|  BTREE     |              |                         | YES      | NULL        |
+--------------+------------+---------------------+-----------+--------------+
2 rows in set (0.09 sec)
```

```
DESCRIBE politics;

+----------------+----------------+-------+------+-------------+------+
| Field          | Type           | Null  | Key  | Default     | Extra |
+----------------+----------------+-------+-------+------------+------+
| Country        | varchar(4)     | NO    | PRI  | NULL        |      |
| Independence | date             | YES  |      | NULL        |      |
| Dependent      | varchar(4)     | YES  |      | NULL        |      |
| Government   | varchar(120)  | YES  |      | NULL        |      |
+----------------+-----------------+-------+------+------------+-------+
4 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM politics;

+----------+
| count(*) |
+----------+
|      238 |
+----------+
1 row in set (0.00  sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM politics;
```

```
+--------------+----------------+----------------+------------------+-------------------+
| Table        | Non_unique  | Key_name    | Seq_in_index  | Column_name  |
+--------------+----------------+----------------+------------------+-------------------+
|  politics      |                    0 |  PRIMARY   |                    1 | Country          |
+--------------+----------------+-----------------+-----------------+-------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A             |              238 |        NULL |     NULL |          |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE      |                |                         | YES       | NULL         |
+--------------+------------+---------------------+-----------+--------------+
1 row in set (0.11 sec)
```

```
DESCRIBE population;

+----------------------+-------------+------+------+---------+-------+
| Field                | Type        | Null | Key  | Default | Extra |
+----------------------+-------------+------+------+----------+------+
| Country              | varchar(4)  | NO   | PRI  | NULL    |       |
| Population_Growth    | float       | YES  |      | NULL    |       |
| Infant_Mortality     | float       | YES  |      | NULL    |       |
+----------------------+-------------+-------+------+---------+------+
3 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM population;

+----------+
| count(*) |
+----------+
|      238 |
+----------+
1 row in set (0.04 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM population;
```

```
+--------------+----------------+----------------+------------------+-------------------+
| Table        | Non_unique  | Key_name    | Seq_in_index  | Column_name  |
+--------------+----------------+----------------+------------------+-------------------+
|  population  |                  0 |  PRIMARY   |                  1 | Country          |
+--------------+----------------+-----------------+-----------------+-------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed     | Null    |
+-------------+---------------+--------------+------------+--------+
|  A             |              238 |        NULL |     NULL |          |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE      |                |                           | YES       | NULL         |
+--------------+------------+---------------------+-----------+--------------+
1 row in set (0.05 sec)
```

```
DESCRIBE religion;

+-------------+--------------+-------+-----+---------+--------+
| Field       | Type         | Null  | Key | Default | Extra  |
+-------------+--------------+-------+-----+---------+--------+
| Country     | varchar(4)   | NO    | PRI | NULL    |        |
| Name        | varchar(50)  | NO    | PRI | NULL    |        |
| Percentage  | float        | YES   |     | NULL    |        |
+------------+---------------+-------+-----+-----------+-------+
3 rows in set (0.00 sec)
```

```
SELECT count(*)
FROM religion;

+----------+
| count(*) |
+----------+
|      454 |
+----------+
1 row in set (0.02 sec)
```

*split into 3 separate tables for show index

```
SHOW index
FROM religion;

+-----------------+----------------+-----------------+------------------+--------------------+
| Table           | Non_unique  | Key_name      | Seq_in_index  | Column_name    |
+-----------------+----------------+-----------------+------------------+--------------------+
|  religion       |              0 |  PRIMARY     |                  1 | Name              |
|  religion       |              0 |  PRIMARY     |                  2 | Country           |
+-----------------+----------------+------------------+-----------------+--------------------+

+-------------+---------------+-------------+-------------+--------+
| Collation    | Cardinality  | Sub_part    | Packed      | Null    |
+-------------+---------------+--------------+------------+--------+
|  A           |              37 |        NULL |     NULL |         |
|  A           |            417 |        NULL |     NULL |         |
+-------------+---------------+--------------+------------+--------+

+--------------+------------+---------------------+----------+---------------+
| Index_type  | Comment | Index_comment   | Visible  | Expression   |
+--------------+------------+---------------------+----------+---------------+
|  BTREE     |             |                       | YES      | NULL        |
|  BTREE     |             |                       | YES      | NULL        |
+--------------+------------+----------------------+-----------+--------------+
2 rows in set (1.20 sec)
```

```
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;
```

```
+-----------+-------------------+---------------+
| Database | Table              | Size in MB  |
+-----------+-------------------+---------------+
| mondial  | ismember          |          0.45 |
| mondial  | city              |          0.27 |
| mondial  | province          |          0.17 |
| mondial  | geo_river         |          0.08 |
| mondial  | located           |          0.08 |
| mondial  | geo_sea           |          0.08 |
| mondial  | locatedon         |          0.05 |
| mondial  | ethnicgroup       |          0.05 |
| mondial  | island            |          0.05 |
| mondial  | islandin          |          0.05 |
| mondial  | geo_island        |          0.05 |
| mondial  | organization      |          0.03 |
| mondial  | country           |          0.03 |
| mondial  | economy           |          0.02 |
| mondial  | encompasses       |          0.02 |
| mondial  | religion          |          0.02 |
| mondial  | geo_source        |          0.02 |
| mondial  | mergeswith        |          0.02 |
| mondial  | river             |          0.02 |
| mondial  | borders           |          0.02 |
| mondial  | geo_desert        |          0.02 |
| mondial  | mountain          |          0.02 |
| mondial  | sea               |          0.02 |
| mondial  | geo_estuary       |          0.02 |
| mondial  | mountainonisland |             0.02 |
| mondial  | continent              |             0.02 |
| mondial  | geo_lake               |             0.02 |
| mondial  | lake                   |             0.02 |
| mondial  | politics               |             0.02 |
| mondial  | desert                 |             0.02 |
| mondial  | geo_mountain           |             0.02 |
| mondial  | language               |             0.02 |
| mondial  | population             |             0.02 |
+----------+------------------------+--------------+
33 rows in set (0.19 sec)
```

Question 4

(a) 1.7MB

(b) The largest is ISMEMBER while

The smallest are BORDERS, CONTINENT, DESERT, ECONOMY, ENCOMPASSES, GEO_DESERT, GEO_ESTUARY, GEO_LAKE, GEO_MOUNTAIN, GEO_SOURCE, LAKE, LANGUAGE, MERGESWITH, MOUNTAIN, MOUNTAINONISLAND, POLITICS, POPULATION,RELIGION, RIVER, SEA

(c) Yes, according to the result in question 3, it can be implied that the higher the cardinality of a relation, the higher the data size in bytes for any two relation in any databases.

(d) Yes, by taking an example of relation geo_lake and encompasses which have the same total size. Both have the same number of degree and a slight difference in cardinality and it is safe to conclude that the both relations are almost identical in size. Hence, the number of degrees in a relation matters the same as cardinality.

(e) Yes, the higher the cardinality and the degree in a relation, the more data it is consisted of. Thus, resulting in a higher total byte of data.

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.)

```
SELECT name, population, population/area as “population density”
FROM country
LIMIT 5;
```

```
+--------------------------+--------------+----------------------------+
| name                     | population   | population density         |
+--------------------------+--------------+----------------------------+
| Austria                  |    8023244   |  95.68567680381634   |
| Afghanistan              |   22664136 |  35.00252664092664    |
| Antigua and Barbuda  |      65647     | 148.52262443438914  |
| Albania                  |    3249136  | 113.01342608695653   |
| American Samoa        |     65628    |  329.7889447236181    |
+---------------------------+--------------+----------------------------+
5 rows in set (0.14 sec)
```

(b) What is the query that will list the religions found in Japan?

```
SELECT religion.name
FROM religion, country
WHERE religion.country = country.code AND country.name = “Japan”;
```

```
+-----------+
| name       |
+-----------+
| Buddhist |
+-----------+
1 row in set (0.07 sec)
```

(c) What is the query that will list the names of the countries which have at least one religion in common with Japan?

```
SELECT country.name
FROM religion, country
WHERE religion.country = country.code AND
          religion.name =  (SELECT religion.name
                            FROM religion, country
                            WHERE religion.country = country.code AND
                            country.name = “Japan”)
LIMIT 5;
```

```
+-----------+
| name      |
+-----------+
| Bhutan    |
| Brunei    |
| Sri Lanka |
| India     |
| Japan     |
+-----------+
5 rows in set (0.00 sec)
```

(d) What is the query that will list the total GDP for all countries added together? (This will be a single value.)

```
SELECT sum(gdp)
FROM economy;
```

```
+-----------------------+
| sum(gdp)              |
+-----------------------+
| 34101945.1100111 |
+-----------------------+
1 row in set (0.06 sec)
```

(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.)

```
SELECT country.name
FROM economy, country
WHERE economy.country = country.code AND
        inflation = (SELECT min(inflation)
                     FROM economy)
LIMIT 5;
```

```
+-------+
| name  |
+-------+
| Nauru |
+-------+
1 row in set (0.04 sec)
```

(f) What is the query that will list the names of the countries which are members of Interpol? (Remember to use LIMIT 5.)

```
SELECT country.name
FROM ismember, country
WHERE ismember.country = country.code AND
            organization = “Interpol” AND type = “Member”
LIMIT 5;

+-------------------------+
| name                    |
+-------------------------+
| Albania                 |
| Algeria                 |
| Andorra                 |
| Angola                  |
| Antigua and Barbuda |
+--------------------------+
5 rows in set (0.04 sec)
```

(g) What is the query that will list the names of the countries which are not members of Interpol? (Hint: this is going to require a *set difference*.) See subject guide, volume 1, page 90.

```
SELECT country.name
FROM ismember, country
WHERE ismember.country = country.code AND
            Country.name not in (  SELECT country.name
                                    FROM ismember, country
                                    WHERE ismember.country = country.code AND
                                          organization = “Interpol” AND
                                          type = “Member”)

LIMIT 5;
```

```
+--------------------+
| name               |
+--------------------+
| Afghanistan        |
| American Samoa |
| Anguilla           |
| Bermuda            |
| Bhutan             |
+---------------------+
5 rows in set (0.10 sec)
```

(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;
```

```
+-------+-----------+--------------+----------+----------+---------+---------------+
| Name  | GDP       | Agriculture  | Service  | Industry | Area    | Population    |
+-------+-----------+--------------+----------+----------+---------+---------------+
| Japan |    152000 |            2 |       34 |       64 | 377835  |  125449703    |
+-------+-----------+--------------+----------+----------+---------+---------------+
1 row in set (0.02 sec)
```

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

```
+-------+-----------+--------------+----------+----------+---------+---------------+
| Name  | GDP       | Agriculture  | Service  | Industry | Area    | Population    |
+-------+-----------+--------------+----------+----------+---------+---------------+
| Japan | 2679200   |          2.1 |     40.2 |     57.7 | 377835  |  125449703    |
+-------+-----------+--------------+----------+----------+---------+---------------+
1 row in set (0.00 sec)
```

Yes, there is a difference. The first query does not have any join predicate and thus displaying the attributes which belong to the first tuple with country code of ‘A’ in economy table. However, the second query joins the table with join predicate that links both tables thus resulting with the real Name, GDP, Agriculture, Service, Industry, Area, Population of the country Japan.

(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**).

```
SELECT name, count(country) as total_number
FROM language
GROUP BY name
LIMIT 5;
```

```
+--------------------+-----------------+
| name               | total_number  |
+--------------------+-----------------+
| Afghan Persian   |                 1 |
| Afrikaans          |                 1 |
| Albanian           |                 2 |
| Arabic             |                 3 |
| Armenian           |                 3 |
+--------------------+-----------------+
5    rows in set (0.11 sec)
```

(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**).

```
SELECT name, count(country) as total_number
FROM language
GROUP BY name
HAVING count(country)>4
LIMIT 5;
```

```
+---------+-----------------+
| name    | total_number  |
+---------+-----------------+
| English  |               21 |
| French   |               10 |
| German  |                 5 |
| Russian  |                 6 |
| Spanish  |                 8 |
+----------+----------------+
5 rows in set (0.00 sec)
```

Part B

Question 1

(a) If the primary key is reduced into Actor alone, the table will only allow one Actor to be auditioned. It means that an Actor can only have one Role and AuditionDate to choose from. For the rejected Result, there is no opportunity to do a re-audition since the Actor row can only be filled once with a particular name. Thus, making the table unreliable.

(b) Similar to the condition where only Actor is the primary key. By having Role alone, the table will only allow one role to be auditioned. For having only one role, the other attribute will be a single unique value. Therefore, it cannot be considered as a preliminary screening audition since there are only one Role which can only be filled with one exact Actor, AuditionDate and Result.

(c) If AuditionDate replace the current primary keys, AuditionDate should be unique and thus there will only be one date for every row. When the data are added, it can be concluded that the table only allows one event to take place in a single date.

(d) By having Actor + Role + AuditionDate as the primary keys, the table will have a consequence of prohibiting an actor to have more than one audition for different Role within the same date and a Role to be auditioned by more than one Actor in the same day.

(e) By having the whole attribute as the primary key, the table will not allow an Actor to audition for the Role they chose in the same day if the first two attempts were not successful.

Question 2

Yes, we would need to change the primary key. If the primary keys remain the same, there will be a duplicate tuple of Actor and Role whenever the actors re-audition for the previous roles. The new key would be the AuditionDate since it is stated in the question that Actor is allowed to do re-audition at a later date.

Part C

Question 1

- SheepID -> Owner, Birthdate
- SheepID, WeighingDate -> Weight
- WeighingDate -> Vet
- Vet -> VetPhoneNum

Question 2

Problem	**Update Anomaly**	**Deletion Anomaly**	**Insertion Anomaly**	**None of these.**
a. Instead of entering '2014-08-15' for one 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 their 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.	✓			

Question 3

SheepID	**Owner**	**Birthdate**
K3922	McNab013	2013-05-12
M3110	Smith002	2015-05-12

SheepID	**WeighingDate**	**Weight**
K3922	2013-08-14	22
M3110	2017-08-17	34

WeighingDate	**Vet**
2013-08-14	M330
2017-08-17	K339

Vet	**VetPhoneNum**
M330	7633088852
K339	7602907550

Part D

(a) Deadlock in a database is a condition where two or more transactions are waiting for one another to give up their locks. The transaction that are involved in a deadlock should wait for one another to unlock in order to go on. The activity will be stopped and remain at the current condition forever unless the deadlock is detected by database management system and one of the transactions is terminated.

(b) Data dictionary is a set of information that holds the standard description of a database. It is consisted of mostly the contents, format, structure, meaning, source, usage and relationship between its element. It has all its data from the tables in the database. Data dictionary is visualized as a table that holds few columns with specific attributes and rows.

(c) Query optimization is the attempt to execute a query from user in a most efficient way. It analyses the execution mechanism to aid the execution as efficiently as possible. This mechanism uses the statistics of the related data for the main consideration to determine the best option. The data will then be used to assist the multiple access methods, such as possible different joins and removing unnecessary columns, to generate the optimal result.

(d) View in a database is a virtual table which is consisted of the result of a query which is generated from one or more table. It has the features of table in which it can be joined, grouped, dropped, inserted and updated. However, it does not store any data and it is just presented virtually. Nonetheless, whenever the real table is modified, the view will follow accordingly even though it is just a logical form of it.

(e) SQL injection is a process of injecting a malicious code through web input. The injection is executed with a purpose of harming or even destroying a database. Once the database is attacked, the attacker can modify the database, read the source code and even write files to the server. SQL injection can attack any website or web application that benefit from an SQL-based database.

References

(2013, January 24). Retrieved February 19, 2019, from https://db.apache.org/derby/docs/10.0/manuals/develop/develop75.html

Database SQL Tuning Guide. (2017, July 24). Retrieved February 19, 2019, from https://docs.oracle.com/database/121/TGSQL/tgsql_optcncpt.htm#TGSQL194

Deadlocks. (n.d.). Retrieved February 19, 2019, from https://docs.oracle.com/javadb/10.6.2.1/devguide/cdevconcepts28436.html

Fontaine, D. (2018, March 09). Database Normalization and Primary Keys. Retrieved February 19, 2019, from https://tapoueh.org/blog/2018/03/database-normalization-and-primary-keys/

Khillar, S. (2017, October 02). Retrieved February 19, 2019, from http://www.differencebetween.net/technology/difference-between-primary-key-and-unique-key/

Kononow, P. (2018, October 01). What is Data Dictionary - Dataedo Data Glossary. Retrieved February 19, 2019, from https://dataedo.com/kb/data-glossary/what-is-data-dictionary

Rouse, M. (2005, September). What is data dictionary? - Definition from WhatIs.com. Retrieved February 19, 2019, from https://searchmicroservices.techtarget.com/definition/data-dictionary

Rouse, M. (2010, January). What is SQL injection? - Definition from WhatIs.com. Retrieved February 19, 2019, from https://searchsoftwarequality.techtarget.com/definition/SQL-injection

SQL injection. (n.d.). Retrieved February 19, 2019, from https://portswigger.net/web-security/sql-injection

What is SQL Injection (SQLi) and How to Fix It. (n.d.). Retrieved February 19, 2019, from https://www.acunetix.com/websitesecurity/sql-injection/