Coursework-Data Managment

mg5u19

May 2020

1 The Relational Model

1.1 EX1

Data Set	Data Type
dateRep	int
day	int
month	int
year	int
cases	int
deaths	int
countriesAndTerritories	string
geoId	string
countryterritoryCode	string
popData2018	int
continentExp	string

Table 1: Dataset Table

1.2 EX2

Functional Dependencies:

$\{dateRep\} \longleftrightarrow \{day, month, year\}$

$\{countriesAndTerritoryCode, geoId\} \longrightarrow \{countriesAndTerritories, continentExp, popData2018\}$

$\{countriesAndTerritoryCode, dateRep\} \longrightarrow \{cases, deaths, \}$

1.3 EX3

Potential candidate keys:
geoId and countryterritoryCode.

1.4 EX4

Primary key:
The primary key should be {countriesAndTerritories, dataRep} because it is compact, unique and definite for every country. From it we can access the country code, continent, popData, geoId, cases and deaths.

2 Normalisation

2.1 EX5

Partial key dependencies:

{countriesAndTerritories} $\longrightarrow$ {continentExp,geoID,popData2018,countryTerritoryCode}

$\{dateRep\} \longrightarrow \{day, month, year\}$

Additional Relations:

{countriesAndTerritories,dateRep} $\longrightarrow$ {cases,deaths}

2.2 EX6

In order to convert the relation into a 2nd normal form we need to distribute the data into three tables: one table with key countriesAndTerritories and attributes countryTerritoryCode, continentExp, geoId and popData2018, another with key dateRep and attributes day, month and year, and last one with key dateRep, countriesAndTerritories with attributes cases and deaths.

2.3 EX7

Transitional Dependencies:

{geoId} $\longrightarrow \{continentExp, countryTerritoryCode, popData2018\}$

$\{countryTerritoryCode\} \longrightarrow \{popData2018\}$

2.4 EX8

At the end I have 5 tables:

1. Date Table - {dateRep, day, month, year}, Primary Key - dateRep
2. Cases Table - {dateRep, countriesAndTerritories, cases, deaths}, Primary Key - dateRep and countriesAndTerritories
3. Countries Table - {countriesAndTerritories, geoId}, Primary Key - countriesAndTerritories
4. Locations Table - {geoId, countryTerritoryCode, ContinentExp}, Primary Key - geoId
5. Population Table - {countryTerritoryCode, popData2018}, Primary Key - countryTerritoryCode

I distributed the data in that way because it is the most optimal way to access and find the information needed. I decided to separate the countryTerritoryCode and popData2018 in a new table because when I don't have a code, there is no popData2018 available.

2.5 EX9

Yes, my relation is in Boyce-Codd Normal Form because I don't have any partial key dependencies, nor any transitive dependencies. Also, the non-prime keys are only dependent on the prime ones.

3 Modelling

3.1 EX11

I have included an index on the countriesAndTerritories column on the Cases table in order to search for the country I need faster and more efficient.

4 Querying

4.1 EX14

```
SELECT SUM(cases),SUM(deaths) FROM Cases
```

4.2 EX15

```
SELECT dateRep,cases
FROM Cases
WHERE countriesAndTerritories = 'United_Kingdom'
ORDER BY SUBSTR(dateRep,7,4),SUBSTR(dateRep,4,2),
SUBSTR(dateRep,1,2) ASC;
```

4.3 EX16

```
SELECT Locations.continentExp,Cases.dateRep,
SUM(Cases.cases) AS sumOfCases,SUM(Cases.deaths) AS sumOfDeaths
FROM Cases
LEFT JOIN Countries
ON Cases.countriesAndTerritories = Countries.countriesAndTerritories
LEFT JOIN Locations
ON Locations.geoId = Countries.geoId
GROUP BY continentExp,dateRep
ORDER BY SUBSTR(dateRep,7,4),SUBSTR(dateRep,4,2),
SUBSTR(dateRep,1,2) ASC;
```

4.4 EX17

```
SELECT Cases.countriesAndTerritories,
CAST(SUM(Cases.cases)AS FLOAT)/CAST(Populations.popData2018 AS FLOAT)*100
AS percentageOfCases,
CAST(SUM(Cases.deaths)AS FLOAT)/CAST(Populations.popData2018 AS FLOAT)*100
AS percentageOfDeaths
FROM Cases
LEFT JOIN Countries
ON Cases.countriesAndTerritories = Countries.countriesAndTerritories
LEFT JOIN Locations
ON Countries.geoId = Locations.geoId
LEFT JOIN Populations
ON Populations.countryTerritoryCode = Locations.countryTerritoryCode
GROUP BY Cases.countriesAndTerritories;
```

4.5 EX18

```
SELECT countriesAndTerritories,
CAST(deaths AS FLOAT)/CAST(cases AS FLOAT) AS percentDeathsOfPop-
ulation
FROM Cases
ORDER BY percentDeathsOfPopulation DESC
LIMIT 10;
```

4.6 EX19

```
SELECT dateRep,countriesAndTerritories,cases,deaths,
SUM (cases) OVER(ORDER BY SUBSTR(dateRep,7,4),
SUBSTR(dateRep,4,2),
SUBSTR(dateRep,1,2) ASC) AS sumOfCases,
SUM(deaths) OVER(ORDER BY SUBSTR(dateRep,7,4),
SUBSTR(dateRep,4,2),
SUBSTR(dateRep,1,2) ASC) AS sumOfDeaths
FROM Cases
WHERE countriesAndTerritories = 'United_Kingdom';
```