COMP1204: Data Management
Coursework Two: SQL

Xiaoke Li
31951473

May 20, 2022

1 The Relational Model

1.1 EX1

Relation:
dataset(dateRep, day, month, year, cases, deaths, countriesAndTerritories, geoId, countryterritoryCode, popData2020, continentExp)

attributes and types

1. dateRep: TEXT
2. day: INTEGER
3. month: INTEGER
4. year: INTEGER
5. cases: INTEGER
6. deaths: INTEGER
7. countriesAndTerritories: TEXT
8. geoId: TEXT
9. countryterritoryCode: TEXT
10. popData2020: INTEGER
11. continentExp: TEXT

1.2 EX2

Minimal set of FDs:
(Any two country in the world may or may not have the same population, thus here let assume they will have same value)

1. dateRep → day
2. dateRep → month
3. dateRep → year
4. (day,month,year) → dateRep
5. geoId → countriesAndTerritories
6. geoId → countryterritoryCode
7. geoId → continentExp
8. geoId → popData2020
9. (dateRep,geoId) → cases
10. (dateRep,geoId) → deaths
    Contrary
11. countriesAndTerritories → geoId
12. countryterritoryCode → geoId

1.3 EX3

list all the potential candidate keys

1. dateRep,geoId
2. dateRep,countriesAndTerritories
3. dateRep,countryterritoryCode

1.4 EX4

My primary key:
dateRep,geoId
Reason:

1. It is candidate key
2. geoId is short and Easy to distinguish
3. Each geoId is unique

2 Normalisation

2.1 EX5

Partial-key dependencies:

- dateRep→ day
- dateRep→ month
- dateRep→ year
- geoId → countryterritoryCode
- geoId → continentExp
- geoId → popData2020
- geoId → continentExp

Then we divide it into 3 tables

1. dateRep,geoId → cases, deaths
2. dateRep → day, month, year
3. geoId → countriesAndTerritories, countryterritoryCode, popData2020, continentExp

2.2 EX6

2NF Normal Form

- Cases(dateRep,geoId, cases, deaths)
- Date(dateRep, day, month, year)
- Country(geoId, countriesAndTerritories, countryterritoryCode, popData2020, continentExp)

The primary keys for their respective tables are:
-Date - dateRep
-Country - geoId
-Cases - dateRep, geoId

The foreign keys for their respective tables are:
-Date - None
-Country - None
-Cases - dateRep, geoId

2.3 EX7

There are no more transitive dependencies in those relations.

2.4 EX8

The relations are the same as which is in 2NF, because there are no transitive dependencies. All the non key attributes are either determined by key.

2.5 EX9

It is a BCNF. Every determinant is a candidate key!

3 Modelling

3.1 EX10

```
sqlite3 coronavirus.db
.open coronavirus.db // create new database
.mode csv
//Then use DataGrip -> Import Data From File(Automatic import)
sqlite3 coronavirus.db .dump > database.sql
```

3.2 EX11

```
create table Date
(
    dateRep Text    not null
        primary key,
    day     INTEGER not null,
    month   INTEGER not null,
    year    INTEGER not null
);

create table Country
(
    geoId                   TEXT    not null
        constraint Country_pk
            primary key,
    countriesAndTerritories TEXT    not null,
    countryterritoryCode    TEXT    not null,
    popData2020             integer not null,
    continentExp            TEXT    not null
);

create table Cases
(
    dateRep TEXT not null,
    geoId   TEXT not null,
    deaths  integer,
    cases   integer,
    constraint Cases_pk
        primary key (dateRep, geoId)
);
```

3.3 EX12

```
INSERT INTO Date (dateRep, day, month, year)
SELECT DISTINCT dateRep, day, month, year
FROM dataset;

INSERT INTO Country (geoId, countryterritoryCode, continentExp, countriesAndTerritories, po
SELECT DISTINCT geoId, countryterritoryCode, continentExp, countriesAndTerritories, popData
FROM dataset;

INSERT INTO Cases (dateRep, geoId, cases, deaths)
SELECT DISTINCT dateRep, geoId, cases, deaths
FROM dataset;
```

3.4 EX13

```
sqlite3 coronavirus.db < dataset.sql
sqlite3 coronavirus.db < ex11.sql
sqlite3 coronavirus.db < ex12.sql
// use test.db to replace coronavirus.db
```

4 Querying

4.1 EX14

```
SELECT sum(cases) AS total_cases, sum(deaths) AS total_deaths
FROM Cases;
```

4.2 EX15

```
SELECT Cases.dateRep,cases
FROM Cases
INNER JOIN Date ON Date.dateRep = Cases.dateRep
WHERE Cases.geoId = 'UK'
ORDER BY year,month,day ASC;
```

4.3 EX16

```
select countriesAndTerritories AS country, Cases.dateRep AS date,
       Cases.cases, Cases.deaths
from Cases
natural join Country
natural join Date
GROUP BY country,dateRep
ORDER BY countriesAndTerritories ASC ,year,month,day ASC
```

4.4 EX17

```
SELECT Country.countriesAndTerritories ,
       round((sum(cases) * 1.0 / popData2020) * 100, 2) AS "cases %",
       round((sum(deaths) * 1.0 / popData2020) * 100, 2) AS "deaths %"
FROM Cases
NATURAL JOIN Country
GROUP BY  countriesAndTerritories,popData2020;
```

4.5 EX18

```
SELECT countriesAndTerritories AS "country",
       ((CAST(sum(deaths) AS REAL) * 100) / sum(cases)) AS "deaths % cases"
FROM Cases
INNER JOIN Country ON Cases.geoId = Country.geoId
GROUP BY country
ORDER BY "deaths % cases" DESC
LIMIT 10;
```

4.6 EX19

```
SELECT Cases.dateRep AS "date",
       sum(deaths) OVER win1 AS "cumulative UK deaths",
       sum(cases) OVER win1 AS "cumulative UK cases"
FROM Cases
NATURAL JOIN Date
WHERE geoid='UK'
WINDOW win1 AS (ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
ORDER BY year, month, day ASC ;
```