Coursework 2

COVID Pandemic Scenario

Members of Group 38

Max Jayatillake (180047619)
Yilmaz Karakus (190245234)
Ayushi Ramesh Khetani (190222176)
Edgar Eugene Adega Kager (190073671)

Participation

All group members have participated in the making of this coursework.

Introduction

The UK government has requested a DB solution for the Covid crisis and for future pandemics. The two main tasks that this DB will support are data analysis on infection rates and the 'Test and Trace' process.
The infection rates will be monitored on a regional level with 4 countries (England, Wales, Scotland, Northern Ireland), 48 Counties and 317 Districts. Data on the total number of infections, the number of new infections per day and infection per 100,000 population will be collected. Each district will have a tier ranging from tier 1 to 3.
The 'Test & Trace' system will store test results for people. For each person, identification and employment status will be stored. Each person is associated with a "bubble" which is a group of people the person has been in contact with. People will have a trace for all the places they have visited, such as workplaces, restaurants, etc.
If a person tests positive, everybody in the bubble and trace shall get notified.

Assumptions

- Countries are derived from Counties which are derived from Districts.
- As a result, attributes such as population, infection per 100k people, number of infected are all derived from sum of those attributes in child Districts. These will be calculated and updated at the end of every day.
- Everybody is registered to the system, regardless of disease and status of disease
- All people's tests are recorded, and people can take multiple tests for multiple dieseas. A history is kept for analytical reasons.
- Only three tier levels are possible, Tier – 1, Tier – 2 and Tier – 3
- There is one trace system across multiple "Place" (of work/entertainment) and Restaurants and Workplaces are derivations of "Place" (this helps future expansion).
- Bubbles are unique in that 1 person can only belong to 1 bubble and they must not have intersections across bubbles. (People in one bubble are only in that bubble).
- The government then decide what they wish to do with this information themselves, all notifications and stuff are decided by them, this system just allows for information on who should be by what relation.
- Only one phone number and one email per "Place" and "Person" is needed to notify them
- All the Ministers speeches should be stored

Entity Relationship Model – Textual Form

Entities

District	(*DistrictID*, DistrictName, County, NumberOfInfections, Population, Tier)
County	(*CountyID*, CountyName, Country)
Country	(*CountryName*, Minister)
Minister	(*MinisterID*, Name, Statement, Country)
Person	(*PeopleID*, Gender, DOB, Ethnicity, Nationality, Profession, Bubble, Contact, District)
Bubble	(*BubbleID*, Person)
Test	(*TestID*, Person, Date, Outcome)
Place	(*PlaceID*, District, Name, Address, Contact)

Relationships

Person *Resides_In* District ⇒ N: 1

Place *Belongs_To* District ⇒ N: 1

District *Belongs_To* County ⇒ N: 1

County *Belongs_To* Country ⇒ N: 1

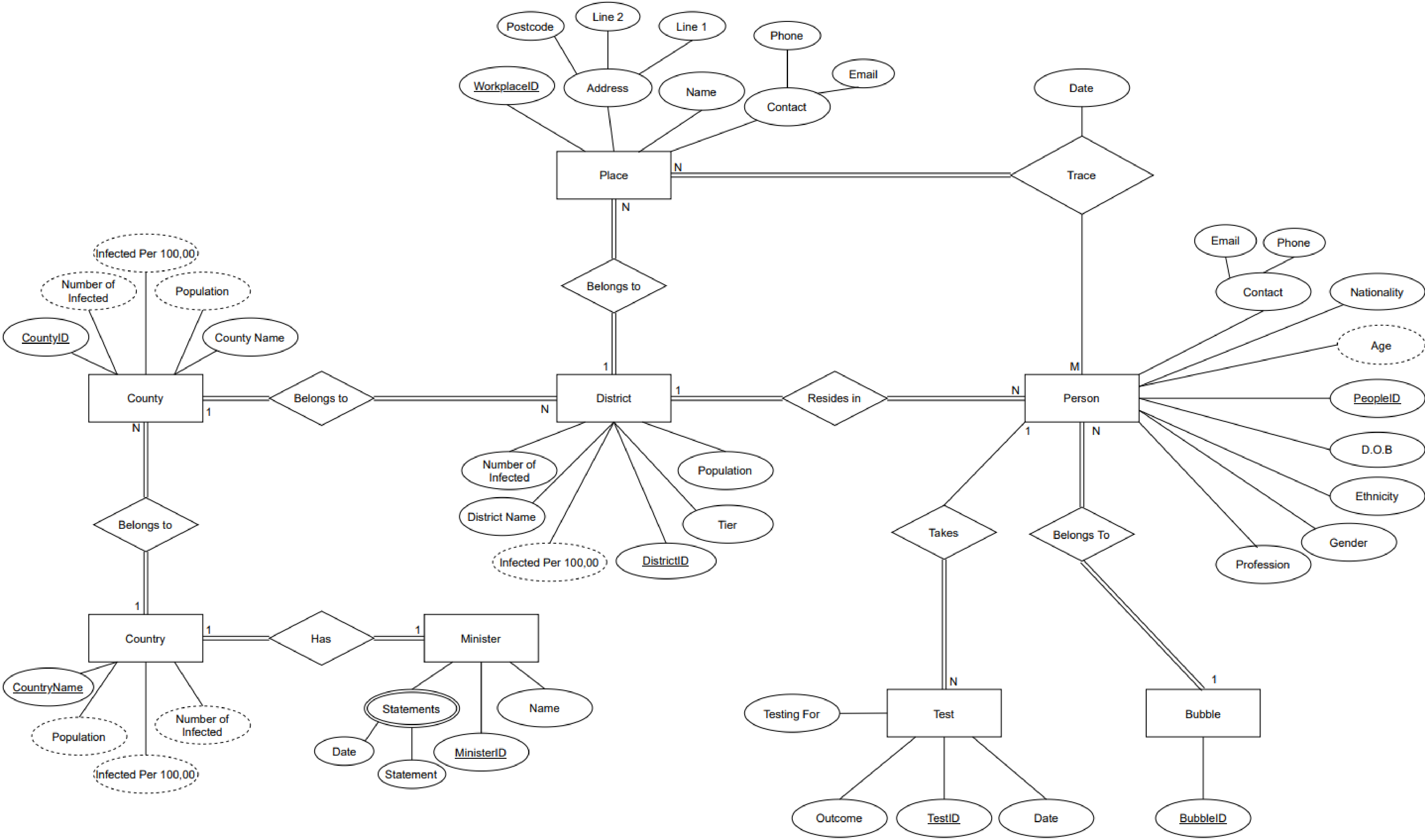
Person *Belongs_To* Bubble ⇒ N: 1

Country *Has* Minister ⇒ 1: 1

Person *Takes* Test ⇒ 1: N

Place *Trace* Person ⇒ N: M

Entity Relationship Model – Diagram



Relational Schema

The relational schema was created by translating our ER Diagram

Legend: Primary Key Foreign Key

District

DistrictID	DistrictName	NumberOfInfection	Population	Tier	CountyID

County

CountyID	CountyName	CountryName

Country

CountryName	MinisterID

Minister

MinisterID	Name

Statement

StatementID	Statement	Date	MinisterID

Person

PersonID	Gender	DOB	Ethnicity	Nationality	Profession	Email	Phone	DistrictID	BubbleID

Bubble

BubbleID

Test

TestID	TestingFor	Outcome	Date	PersonID

Place

PlaceID	Name	Line1	Line2	Postcode	Phone	Email	DistrictID

Trace

Date	PlaceID	PersonID

Normalisation

Normalisation is a technique to reduce data redundancy by dividing larger tables into smaller tables and linking them with relationships without the loss of information. This technique is done by applying a series of rules for each form.

1st Normal Form (1NF)

The rules for 1NF for a design:

- Attributes must be single values
- Each record is unique (multi-valued attributes are removed)

One of the steps that we had to take when generating the table for the first normal form was Trace. Our initial implementation had separate entities for both WorkPlace and Restaurants and a Trace table where a relationship was made. This however meant that we had a multi-valued attribute for our Trace table, for the place the trace was for, as it could tie to either a WorkPlace entity or a Restaurant entity.

To demonstrate normal form, instead of having two separate entities, WorkPlaceTraceId and RestaurantTraceId, we decided to encompass WorkPlace and Restaurant into one entity, Place.

The reason behind this is that having two entities created redundancy in that, records have a valid entry for either only WorkPlaceId or RestarauntId and not both, however both were still stored. This further created complications when a person had their workplace as a restaurant, and thus generalising them under one entity is simpler. Having only one Place entity also means that for future expansion, including other sectors such as Cinemas and Gyms is possible and simple under this design.

Before

TraceId	WorkPlaceId	RestaurantId

WorkPlaceId	WorkPlaceName	DistrictId

RestaurantId	RestaurantName	DistrictId

After
(With deletion of WorkPlace and Restaurant entities)

TraceId	PlaceId	PersonId

PlaceID	Name	Address

Another step we had to take was make each attribute atomic in that it can't be divided into smaller values. Our initial implementation of Place included attributes called "Address" and "Contact" which was not atomic. To resolve this, we had split this into multiple atomic fields: Line1, Line2, Postcode, Phone, Email and DistrictId. This also now allows us to pull up Places within a certain District.

Before

PlaceID	Name	Address	Contact

After

PlaceID	Name	Line1	Line2	Postcode	Phone	Email	DistrictID

As a result, our schema satisfies the requirements for 1st Normal Form.

2nd Normal Form (2NF)

The rules for 2NF for a design:

- Must already be 1NF
- All non-key attributes are fully dependent on the primary key

The second rule can be summarised by saying that an entire row is dependant on a primary attribute (key) in the table. This means that to retrieve a record in a table, we can call on its ID within the table as all other columns are dependant on it.

Throughout our database design, each table is given a primary key as a unique identifier with all other keys being dependant on that unique identifier. This is shown in our Person entity, our District entity for examples and is consistent throughout.

District

DistrictID	DistrictName	NumberOfInfection	Population	Tier	CountyID

Here, the fields DistrictName, NumberOfInfection, Population, Tier and CountyId are dependant on the DistrictId which is our primary key.

As a result, our schema satisfies the requirements for 2nd Normal Form.

3rd Normal Form (3NF)

The rules for 3NF for a design:

- Must already be 2NF
- Has no transitive functional dependencies

A transitive functional dependency is when changing a non-key column might cause any of the other non-key columns to change. This also means that no non-key attribute is dependant on any other non-key attribute. This was not innate in our database design and thus all relations in our schema already satisfied the requirements for 3rd Normal Form.