# **Design process**

## Data Cleansing

1. deleted the column with no column heading because it didn’t seem to have any significance with the rest of the data.
2. deleted the column year-month because this can be generated with a query.
3. formatted all the date fields to match SQL date, that being yyyy-mm-dd
4. separated the date columns and the string columns so I could trim all the string columns to check for leading and trailing spaces.
5. highlighted all duplicates (using conditional formatting) of NHI because these should be unique, there were duplicates so I removed them but kept the duplicates data for future use.
6. checked for any errors in the data (conditional formatting), there was none.
7. set all names to proper string formatting.
8. substituted spaces between names into a comma only 1 instance so last names with spaces weren’t affected. Then split the names into first and last for doctor patient and surgeon
9. deleted rows that the people that haven’t been born yet. But kept the data for future use.
10. I changed a person’s DOB that was 1756 into 1956, assumed it was a typo.

**All the data I removed or changed is held within an excel spread sheet for future use.**

A picture containing timeline

Description automatically generated

### Excel data into appropriate tables and csv files

* I created new sheets for each table in excel and copied all the data that is required. Including a primary key ID for all the tables except the patient (they have NHI).
* From here I wrote a little python script to convert each sheet into a csv file.

## Choice of entities

After noun analysis I chose 6 tables but then after creating a few databases I realised to create the data base to be normalised there is only a need for 5:

1. Patient
2. Doctor
3. Surgeon
4. Department
5. Referral

## Choice of attributes

There was no need to have “Patient Age”, “Days waiting” and “year-month” as an attribute in a table because they can be a calculated field.

### Department

**DepartmentID(PK) – Department Name (All Simple)**

Table, Excel

Description automatically generated

### Doctor

**DoctorID(PK) – Doctor First name – Doctor Last Name – Doctor Type(All Simple)**

Table

Description automatically generated

### Patient

**NHI(PK) – Patient First Name – Patient Last Name – Gender – DOB – DoctorID(FK) (All Simple)**

Table

Description automatically generated

### Surgeon

**SurgeonID(PK) – Surgeon First Name – Surgeon Last Name – DepartmentID(FK) (All Simple)**

Table

Description automatically generated

### Referral

**ReferralID(PK) – Referral Date – Health Target Eligible – Added to Waitlist Date – FSA Date – DoctorID(FK) – NHI(FK) – DepartmentID(FK) – SurgeonID(FK) (All Simple)**

Text, table

Description automatically generated

* For all the tables I generated an ID Number as the Primary Key except for Patient which I used the NHI.

## Connectivity/Relationships

Diagram

Description automatically generated

Diagram

Description automatically generated

* The Patient has one doctor and doctors can have many patients (M:1)
* Department can have many surgeons and surgeons have only one department (1:M)
* Patients have can have many referrals and referrals are for only one patient(1:M)
* Department can have many referrals and referrals can be for only one department (1:M)
* Surgeons can have many referrals and referrals can be for only one surgeon(1:M)
* Doctor can send many referrals and referrals are sent by only one doctor(1:M)

## Bridging table/Composite Table

There was no many to many relationships so there was no need for a bridging table.

## Extent of Normalisation

### 1NF



### 2NF



### 3NF



## Interesting things about design

The way I designed the database there was no need for:

* Bridging table – (apart from referral is kind of the bridging table between all tables)
* Composite table
* Subtype/Supertype entities
* Recursive entities
* Multivalued Attributes

One thing I did think about is there could have been a recursive entity if Surgeon could be head of department which manages surgeons, but that isn’t case.

Another was multivalued attributes, could a doctor be a GP and Private, or Internal and GP? In the specification or data provided, that wasn’t the case either.

A bridging table could have been needed if surgeons where apart of many departments, if a patient can have many doctors, but the specification did not specify this.

The only constraint that I came across was attribute “Health Target Eligible” which was a “yes” or “no” only.

## Version Control

Graphical user interface, text, application, email

Description automatically generated