

A person in a suit

Description automatically generated

ICL-1305 Database Design and Management

TOBY BEEVERS

*Cambria ID: 21075206 Bangor ID: 500704883*

APPLIED DATA SCIENCE DEGREE APPRENTICESHIP BSC

Contents

[What is Normalisation, and why is it important 2](#_Toc155862674)

[Identify the Entities 2](#_Toc155862675)

[Dental Practices 2](#_Toc155862676)

[Staff Members 2](#_Toc155862677)

[Patients 2](#_Toc155862678)

[Appointments 3](#_Toc155862679)

[Making things Atomic 3](#_Toc155862680)

[Entity One - Dental Practices 3](#_Toc155862681)

[Entity Two - Staff Members 4](#_Toc155862682)

[Entity Three – Patients 5](#_Toc155862683)

[Entity Four - Appointments 5](#_Toc155862684)

[First Normal Form 6](#_Toc155862685)

[The Rules of 1NF 6](#_Toc155862686)

[Working Through 1NF 6](#_Toc155862687)

[Violations of 1NF 7](#_Toc155862688)

[Second Normal Form 8](#_Toc155862689)

[The Rules of 2NF 8](#_Toc155862690)

[Working Through 2NF 8](#_Toc155862691)

[Third Normal Form 10](#_Toc155862692)

[The Rules of 3NF 10](#_Toc155862693)

[Working Through 3NF 10](#_Toc155862694)

[Further Normal Form 11](#_Toc155862695)

[Entity Relationship Diagram (ERD) 12](#_Toc155862696)

[Data Dictionary 13](#_Toc155862697)

[The Purpose of a Data Dictionary 14](#_Toc155862698)

[References 15](#_Toc155862699)

# What is Normalisation, and why is it important

Normalisation is the process you follow to reduce data redundancy in your database tables, this means to remove duplicates and improve data integrity. The process of Normalisation uses steps that a database developer can follow, these steps ask set questions along the way and the developer is encouraged to ensure each step is completed before stepping to the next.

In the 1970’s Edger F Codd published a paper where he first described a relational model, he identified the first three levels of Normal Form. Working with Raymond Boyce he later published the Boyce-Codd Normal Form (BCNF) which is offend considered to be slightly stronger than the original definition.

When designing a database, it’s important to consider that if repeating values are stored this increased;

* **The need to more storage**

More tables, more data, more storage. The more data we can start to see a decrease in database efficiency. Cost is also a factor to be considered, and whilst data storage is relatively cheap, extra computer power for our Database can often see significant impacts.

* **The risk of inconsistences**

With reoccurring data, we increase the chance of data inconsistencies, if the software must update the same data in more than one place you are introducing a risk that can be reduced.

* **The complexity of code**

Once the database is created and in production, reoccurring data means that the developer has to consider multiple table updates making code much more complex.

(Choudhary, 2023)(Carter, 2003)(Wikipedia , 2023)

# Identify the Entities

The first part of Normalisation is the task of identifying the Entities associated with the customer requirement, the importance of identifying you entities should not be underestimated as this will begin to allow you to think about your normalisation. Below are the main entities associated with the customer requirements.

### Dental Practices

These are the physical locations, they have addresses and contact details and will be visited MANY times by staff and patients for MANY appointments.

*Llandudno, Rhos-On-Sea, Conwy and Colwyn Bay, with their addresses, and phone numbers.*

### Staff Members

These are the people that will work across ONE or MANY of the Dental Practices, these guys will also undertake ONE or MANY tasks relating to appointments.

*Surgery manager, Dentists, Hygienists, receptionists and cleaners, job descriptions, salary ranges etc. are filed as well as names, addresses, phone numbers, start dates.*

### Patients

These guys are the people visiting the physical locations, they may have ONE or MANY appointments, and could see ONE or MANY staff members.

*Parent/guardian, names, addresses, phone numbers and medical details.*

### Appointments

The final entity or joining entity is the appointment, this is the entity that brings them all together.

Visits are recorded. These include date and time of appointments, the attending Dentist/hygienist, any diagnoses, advised treatment and any drugs prescribed appointment costs £30; each drug in the prescription costs £6.

# Making things Atomic

Atomic means to make the data as small as possible, or rather break it into as many little pieces of information that you can. However, you do need to consider how you intend to use the data after all if you get too small, you’ll never be able to get it back together.

Consider a person’s name, Mr David William Jones, to make this atomic you may consider breaking this in Title, FirstName, MidName and LastName. By doing this you can use the data in a much more efficient way within your database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **FirstName** | **MidName** | **LastName** |
| Mr | David | William | Jones |

Now consider an address, 123 Main Rd, Wrexham, North Wales, LL11 0ZY, although you could break this down into many parts you need to consider the logical approach. Eg 123 Main Rd could be considered as atomic enough, you certainly wouldn’t need to separate Main and Rd, but you may consider separating 123. Here’s what I would consider as Atomic. (Wikipedia, 2023) (Beighley, 2023)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AddressNo.** | **Address** | **TownCity** | **Area/Region** | **PostCode** |
| 123 | Main Rd | Wrexham | North Wales | LL11 0ZY |

### Entity One - Dental Practices

*Llandudno, Rhos-On-Sea, Conwy and Colwyn Bay, with their addresses, and phone numbers.*

*It could be argued that on this occasion the StreetNo and AddressOne could be combined as the chance of this Dentist having more than one practice on the same street is very low. (Composite Key)*

*If this was for an Estate Agent, one could argue this would be much more important as the Agent could have many different houses on a single street.*

|  |
| --- |
| PracticeName |
| StreetNo |
| AddressLineOne |
| AddressLineTwo |
| TownCity |
| County |
| Postcode |
| MainPhoneNumber |

### Entity Two - Staff Members

*Surgery manager, Dentists, Hygienists, receptionists and cleaners, job descriptions, salary ranges etc. are filed as well as names, addresses, phone numbers, start dates.*

*Assumptions have been made here that there is no requirement to track any changes within the entity. For example should the Practice want to track when a staff member moves house, has a change in role or salary this would not be possible with this schema.*

*For the field for JobDescription there is an assumption that this will be a string value, however, this could easily be a URL for a Cloud Storage container to reference a PDF document etc.*

*I have chosen to add a StaffId and Logical Delete to the Staff Members table, this will allow a staff member to leave the business and return at a later date. Assumption is that each time a Staff Member returned they would receive a new StaffId thus allowing for tracked changes.*

*The Logical Delete would allow the Staff Member to be removed from the System but still retain the Historic Appointment Records*

|  |
| --- |
| StaffId |
| FirstName |
| LastName |
| JobTitle |
| JobDescription |
| StreetNo |
| AddressOne |
| AddressTwo |
| TownCity |
| County |
| Postcode |
| PhoneNumber |
| MobileNumber |
| EmploymentStartDate |
| CurrentSalary |
| IsDeleted |

### Entity Three – Patients

*I have chosen to give each patient their own Id, given that there is a high chance of two or more patients having the same name the names cannot be considered as a Primary Key.*

*Assumptions have been made on the fields Medical Records, Allergies, ParentGuardianName, these will be a string value allowing for free text.*

*This assumes there is no relationship between a patient under the age of 18 and another patient as their parent. Should a relationship be required a foreign key of the PatientNumber could be included.*

*I have added a Boolean Key IsOver18 to allow for easy data quality validation on the DateOfBirth, and the potential to use a GETDATE() function. This will allow the ability to see all Patients as they are reaching the age of 18.*

|  |
| --- |
| PatientNumber |
| PatientTitle |
| PatientFirstName |
| PatientLastName |
| StreetNo |
| AddressOne |
| AddressTwo |
| TownCity |
| County |
| Postcode |
| PhoneNumber |
| MobileNumber |
| DateOfBirth |
| IsOver18 |
| Medical Records |
| Allergies |
| ParentGuardianName |
| PatientSurgery |

### Entity Four - Appointments

*Assumptions have been made on the attributes of an appointment as follows;*

* *Diagnoses, will be a string.*
* *Treatment, will be a string.*
* *Any drugs prescribed will be a string and the calculation fo the drugs will not be done within the database.*

|  |
| --- |
| AppointmentId |
| AppointmentDate |
| AppointmentTime |
| AppointmentCharge |
| PatientNumber |
| StaffId |
| PracticeName |
| Diagnoses |
| Treatment |
| DrugsPrescribed |
| PrescriptionCharge |

# First Normal Form

First Normal Form or 1NF aims to start reducing your data into *themes*, you may have to complete multiple rounds known as 2NF, 3NF and so on. We are aiming to have each database table dealing with only one *theme*, this reduces the chance of duplicated data. Some of the benefits of 1NF are a reduction in the size of the database, and the databases ability to run quicker, these are all by-products of NF as you remove the chance of duplicate data (Taylor, 2019).

### The Rules of 1NF

* A screenshot of a medical form

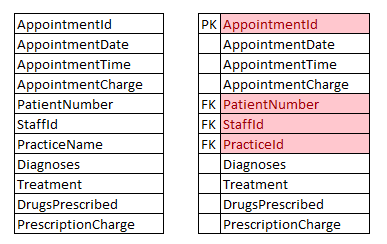
  Description automatically generatedThere are no repeating groups
* All data values are atomic
* Each field has a unique name
* It has a primary key

### Working Through 1NF

Whilst processing the data into First Normal Form I have identified that some fields did not have a unique name, to address this I have include the entity description eg. *AddressLineOne* appeared in multiple tables, these have been charged to *EntityAddressLineOne* and so on.

A screenshot of a table

Description automatically generatedI have also included a Primary Key for each table, this has been achieved with the introduction of an Id key, the intention is to make this a numeric value that is self-generating.



Whilst nominalising the Dental Practice Entity it was decided that whilst the PracticeName could be considered as unique this is not defined in the data.

There are two approaches to over come this as seen below;

1. You could indeed use two of the Simply Keys to create a Composite KeyA screenshot of a computer

   Description automatically generated
2. Or, you could introduce a new Primary Key in the form of an PracticeId

A screenshot of a computer

Description automatically generated

### Violations of 1NF

* If you use row order to convey information
* Mixing data types within the same column
* Not having a primary key (PK)
* Storing repeating groups on a single row

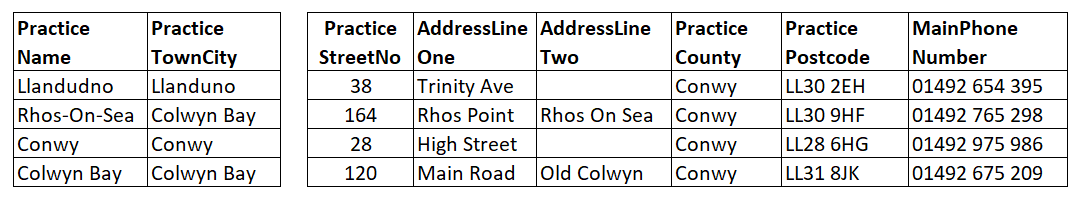
# Second Normal Form

To process Second Normal Form or 2NF, we need to consider the idea of Functional Dependencies. A functional dependency is the relationship between or amongst two of more attributes, put simply this means that one attribute is functional dependant on another. E.g. If you know the value of the first attribute you are able to determine the other attribute.

### The Rules of 2NF

* It is the first normal form (1NF)
* **All non-key attributes are dependent on ALL PARTS of the primary key**
* Each field has a unique name
* It has a primary key

Let’s take a minute to consider what *All non-key attributes are dependent on ALL PARTS of the primary key* means. Well let’s look at the Practice Entity in a slightly different way, consider that the below table is an example of the dataset within the table. All of the Non Key Attribute are dependent on both the Practice Name and Practice Town/City, for example if you know Llandudno, Llandudno you can determine the address of the Practice.



These are our Primary/Composite Key

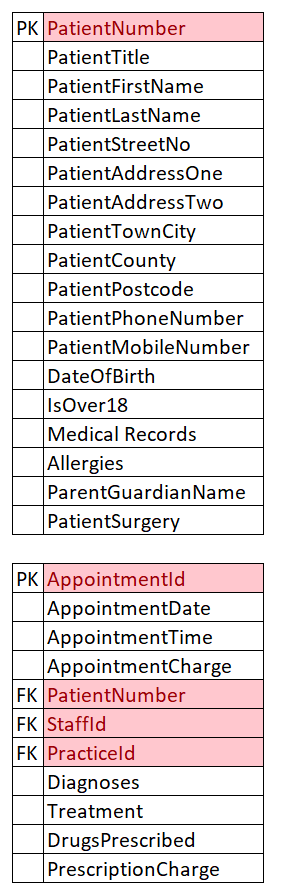
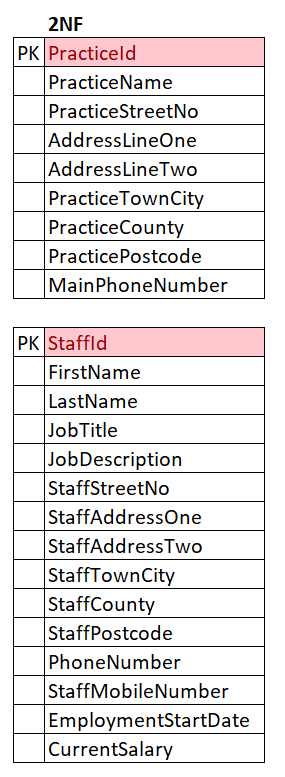
These are our NON-KEY attributes

### Working Through 2NF

To work through Second Normal Form or 2NF we will be reviewing that ***All non-key attributes are dependent on ALL PARTS of the primary key.*** However, it is worth noting here that I have chosen to use a table identifier is my schema, this means there are no composite keys, making my schema design 2NF by default.

#### Entity One – Dental Practices

As explained above, I have made the decision to introduce a PracticeId as the Primary Key for the Dental Practices entity. The benefit of using this approach will allow the ability to scale, consider should the Dental Practice decide to move premise but within the same Town/City, a new Practice ID could be created, along with the introduction of a logical delete. This approach would allow for historic records of appointments to remain with the old Practice.

A white box with black text

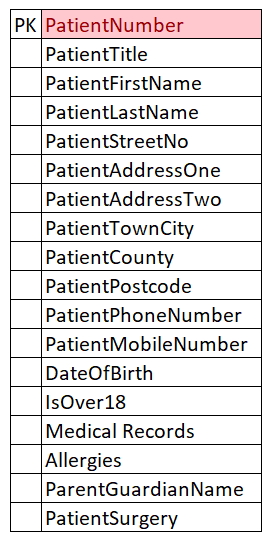
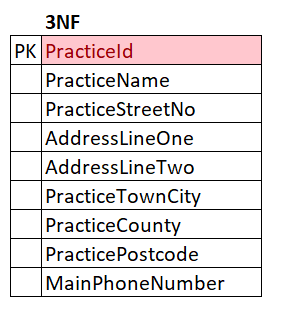
Description automatically generated

# Third Normal Form

Moving to Third Normal Form or 3NF, we now consider Transitive Dependencies. A Transitive Dependency is where one attribute depends on another attribute and another attribute and so on, this is important to consider as should a deletion occur in the first table what would be the impact in the subsequent tables.

### The Rules of 3NF

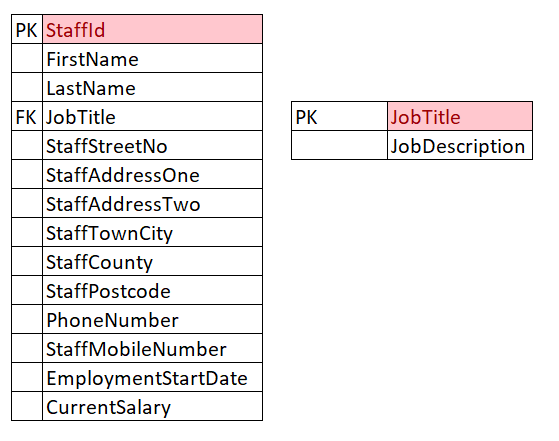
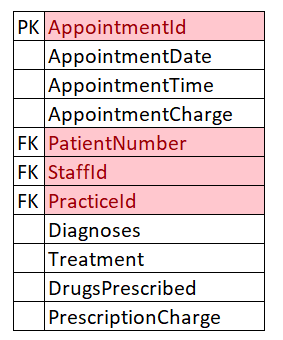
* It is the second normal Form (2NF)
* **All non-key attributes are not dependant on any other non-key attribute**
* Each field has a unique name
* It has a primary key

Lets take another minute to consider what *All non-key attributes are not dependant on any other non-key* attribute actually means. For this we can look at the Staff Table, consider the domains Job Title and Job Description, consider that Job Description has a transitive dependency on Job Title, meaning that a persons Job Title could be update without their Job Description being updated, causing inconsistencies.

### Working Through 3NF

The Entities of Practices and Patients are already in 3NF.

Now taking a look at the Staff Entity we can see that there is a transaitve dependacny between the Job Title and Job Description. To over come this I have moved the Job Description out into a new table



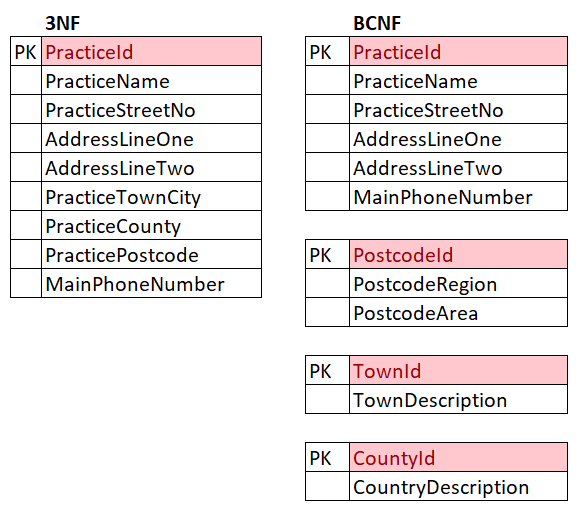
The final entity to consider is the Appointments table, this is considered a Bridging table and whilst an Appointment has attributes of it’s own it is highly dependent on Forgiven Keys that have been defined through 1 & 2 NF.

# Further Normal Form

Although the assignment does not ask for this, I could not ignore what I consider a large opportunity to reduce the risk of repetition, and considering my experience in de normalising data for OLAP and analytics I have included the below arguments which moves towards 4NF.

Reviewing the domains it could be argued that Practice Town/City, PracticeCounty and PracticePostcode are all data points that could be reuse by other Entities. An example of this would be that Postcodes are being used in three Entities, Practices, Staff Members and Patients

#### Entity One – Practices

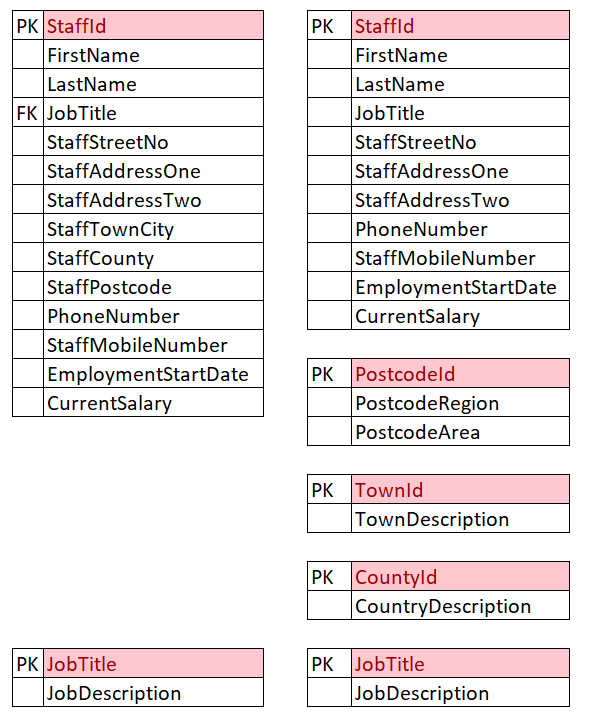


*The decision to split the Practice Entity into four is based around the ability to reuse the Postcode, Town and County domains in other entities, this is to reduce the possibility of duplicate data.*

*Postcodes have then been split into Postcode Region and Postcode area, in the UK we have circa. 1.7M postcodes, the use of Postcode regions allows for a larger area to be considered making the data slightly more atomic.*

*An assumption is being made that the Postcode Entity would not need to hold all 1.7M records, however this approach will allow for the scope of growth should the business decide to open in Rhyl.*

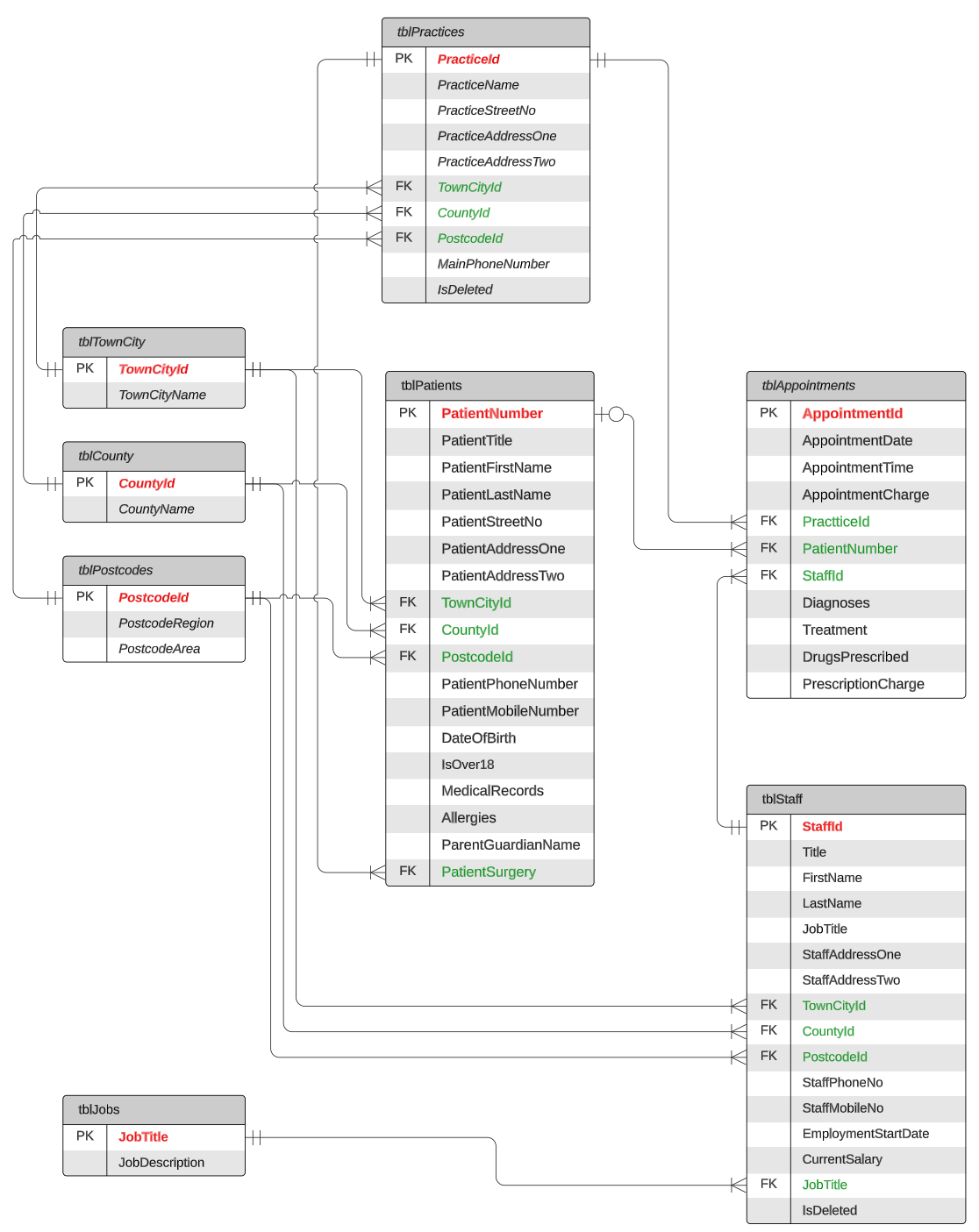
#### Entity Two – Staff

Working through NF on the entity of Staff Member I have identified the use of StaffTownCity, StaffCounty and StaffPostcode, this is the same decision as used for the Practice Entity.

Further consideration has been made for JobTitle and JobDescription, there is an assumption that multiple staff members could have the same Job Title.

There is then a further assumption that a JobTitle will have a unique Job Description, this mean that the JobTitle can be considered as the Primary Key.

# Entity Relationship Diagram (ERD)



(Wikipedia, 2023)

# Data Dictionary

| **P/F** | **Table** | **Field Name** | **Description** | **Nullable** | **Data Type** | **Field Size** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PK | tblPractices | PracticeId | Primary Key | No | Auto Number (Int) | 2 |  |
|  | tblPractices | PracticeName | Name of the Dental Practice | No | Short Text (CHAR) | 20 |  |
|  | tblPractices | PracticeStreetNo | Street Number of the Practice | No | Number (Int) | 5 |  |
|  | tblPractices | AddressLineOne | First line Address | No | Short Text (CHAR) | 20 |  |
|  | tblPractices | AddressLineTwo | Second Line Address |  | Short Text (CHAR) | 20 |  |
| FK | tblPractices | TownId | Foreign Key | No | Number (Int) | 10 | PK from tblTownCity (Lookup) |
| FK | tblPractices | CountyId | Foreign Key | No | Number (Int) | 10 | PK from tblCounty (Lookup) |
| FK | tblPractices | PostcodeId | Foreign Key | No | Number (Int) | 100 | PK from tblPostcodes (Lookup) |
|  | tblPractices | MainPhoneNumber | Phone Number |  | Short Text (CHAR) | 12 | Validation must start 0 |
| PK | tblPostcodes | PostcodeId | Primary Key | No | Auto Number (Int) | 100 |  |
|  | tblPostcodes | PostcodeRegion |  | No | Short Text (CHAR) | 4 |  |
|  | tblPostcodes | PostcodeArea |  | No | Short Text (CHAR) | 3 |  |
| PK | tblTownCity | TownId | Primary Key | No | Auto Number (Int) | 10 |  |
|  | tblTownCity | TownDescription | Name of Town or City | No | Short Text (CHAR) | 50 |  |
| PK | tblCounty | CountyId | Primary Key | No | Auto Number (Int) | 10 |  |
|  | tblCounty | CountyDescription | Name of County | No | Short Text (CHAR) | 50 |  |
| PK | tblStaff | StaffId | Primary Key | No | Auto Number (Int) | 1000 |  |
|  | tblStaff | FirstName |  | No | Short Text (CHAR) | 50 |  |
|  | tblStaff | LastName |  | No | Short Text (CHAR) | 50 |  |
|  | tblStaff | JobTitle |  | No | Short Text (CHAR) | 50 |  |
|  | tblStaff | StaffStreetNo | Street Number of the Practice | No | Number (Int) | 5 |  |
|  | tblStaff | StaffAddressOne | First line Address | No | Short Text (CHAR) | 20 |  |
|  | tblStaff | StaffAddressTwo | Second Line Address |  | Short Text (CHAR) | 20 |  |
| FK | tblStaff | TownId | Foreign Key | No | Number (Int) | 10 | PK from tblTownCity (Lookup) |
| FK | tblStaff | CountyId | Foreign Key | No | Number (Int) | 10 | PK from tblCounty (Lookup) |
| FK | tblStaff | PostcodeId | Foreign Key | No | Number (Int) | 100 | PK from tblPostcodes (Lookup) |
|  | tblStaff | PhoneNumber | Phone Number | No | Short Text (CHAR) | 12 | Validation must start 0 |
|  | tblStaff | StaffMobileNumber | Phone Number |  | Short Text (CHAR) | 12 | Validation must start 0 |
|  | tblStaff | EmploymentStartDate |  | No | Date/Time (DATE) | 8 | Date Only 2023/11/23 |
|  | tblStaff | CurrentSalary |  | No |  |  |  |
| PK | tblJobs | JobTitle | Primary Key | No | Short Text (CHAR) | 50 |  |
|  | tblJobs | JobDescription | Max 500 char for Staff Job Title Description | No | Long Text (TEXT) | 500 |  |
| PK | tblPatients | PatientNumber | Primary Key | No | Auto Number (Int) | 10,000 | Allows for 10,000 patient records |
|  | tblPatients | PatientTitle |  |  | Short Text (CHAR) | 4 | Validation Mr, Mrs, Miss, DR, Other |
|  | tblPatients | PatientFirstName |  | No | Short Text (CHAR) | 50 |  |
|  | tblPatients | PatientLastName |  | No | Short Text (CHAR) | 50 |  |
|  | tblPatients | PatientStreetNo | Street Number of the Practice | No | Number (Int) | 5 |  |
|  | tblPatients | PatientAddressOne | First line Address | No | Short Text (CHAR) | 20 |  |
|  | tblPatients | PatientAddressTwo | Second Line Address |  | Short Text (CHAR) | 20 |  |
| FK | tblPatients | TownId | Foreign Key | No | Number (Int) | 10 | PK from tblTownCity (Lookup) |
| FK | tblPatients | CountyId | Foreign Key | No | Number (Int) | 10 | PK from tblCounty (Lookup) |
| FK | tblPatients | PostcodeId | Foreign Key | No | Number (Int) | 100 | PK from tblPostcodes (Lookup) |
|  | tblPatients | PatientPhoneNumber | Phone Number | No | Short Text (CHAR) | 12 | Validation must start 0 |
|  | tblPatients | PatientMobileNumber | Phone Number |  | Short Text (CHAR) | 12 | Validation must start 0 |
|  | tblPatients | DateOfBirth |  | No | Date/Time (DATE) | 8 | Date Only 2023/11/23 |
|  | tblPatients | IsOver18 |  | No | Yes/No (Bit) | 3 | Boolen 0/1 Yes or No |
|  | tblPatients | MedicalRecords |  |  | Long Text (TEXT) | 500 | Assumes these are historic records, current records will belong to Appointment |
|  | tblPatients | Allergies |  |  | Long Text (TEXT) | 500 |  |
| FK | tblPatients | ParentGuardianName |  |  | Short Text (CHAR) | 50 |  |
| FK | tblPatients | PatientSurgery |  | No | Number (Int) | 2 | PK from tblPractices (Lookup) |
| PK | tblAppointments | AppointmentId | Primary Key | No | Auto Number (Int) | 100,000,000 |  |
|  | tblAppointments | AppointmentDate |  | No | Date/Time (DATE) | 8 | Date Only 2023/11/23 |
|  | tblAppointments | AppointmentTime |  | No | Date/Time (DATE) | 8 | Time Only 10:30:00 |
|  | tblAppointments | AppointmentCharge |  | No | Currency (FLOAT) | 5 | Max 2 Decimal Places 100.50 |
| FK | tblAppointments | PatientNumber | Foreign Key | No | Number (Int) | 10,000 | PK from tblPatients (Lookup) |
| FK | tblAppointments | StaffId |  | No | Number (Int) | 2 | PK from tblStaff (Lookup) |
| FK | tblAppointments | PracticeId |  | No | Number (Int) | 2 | PK from tblPractices (Lookup) |
|  | tblAppointments | Diagnoses |  |  | Long Text (TEXT) | 500 |  |
|  | tblAppointments | Treatment |  |  | Long Text (TEXT) | 500 |  |
|  | tblAppointments | DrugsPrescribed |  |  | Long Text (TEXT) | 500 |  |
|  | tblAppointments | PrescriptionCharge |  |  | Currency (FLOAT) | 5 | Max 2 Decimal Places 100.50 |

### The Purpose of a Data Dictionary

In the context of this assignment a Data Dictionary is being used to support the understanding of the database design, structure, and relationship. It also helps to define a common vocabulary across developer and data communities.

In a wider context a Data Dictionary or Data Catalogue will become a fundamental part of an organisations management of Master Data. By introducing the concept of Master Data Management (MDM) you can extend the above concept of this assignment to include Access Management Policies, Data Relationships across the whole organisation, Data Quality checks with the ability to score the quality of your data across multiple systems.

# References

Beighley, L., 2023. Atomic Data. In: T. P. Programmers, ed. *Head First SQL.* s.l.:O'Reilly, p. 168 to 171.

Carter, J., 2003. *Datebase design and Programming.* 2nd ed. s.l.:McGraw-Hill Eductation.

Choudhary, V., 2023. *What is Data Normalization and Why Is It Important?.* [Online]   
Available at: https://www.geeksforgeeks.org/what-is-data-normalization-and-why-is-it-important/  
[Accessed 9 11 2023].

Microsoft, 2023. *Control data entry formats with input masks.* [Online]   
Available at: https://support.microsoft.com/en-au/office/control-data-entry-formats-with-input-masks-e125997a-7791-49e5-8672-4a47832de8da  
[Accessed 9 11 2023].

Microsoft, 2023. *Restrict data input by using validation rules.* [Online]   
Available at: https://support.microsoft.com/en-au/office/restrict-data-input-by-using-validation-rules-b91c6b15-bcd3-42c1-90bf-e3a0272e988d  
[Accessed 9 11 2023].

Taylor, A. G., 2019. Using SQL to Build Databases. In: *SQL for Dummies.* 9th ed. s.l.:John Wiley & Sons, p. 134 to 136.

Wikipedia , 2023. *Boyce–Codd normal form.* [Online]   
Available at: https://en.wikipedia.org/wiki/Boyce%E2%80%93Codd\_normal\_form  
[Accessed 26 11 202].

Wikipedia, 2023. *Atomicity (database systems).* [Online]   
Available at: https://en.wikipedia.org/wiki/Atomicity\_(database\_systems)  
[Accessed 11 11 2023].

Wikipedia, 2023. *Data Dictionary.* [Online]   
Available at: https://en.wikipedia.org/wiki/Data\_dictionary  
[Accessed 9 11 2023].